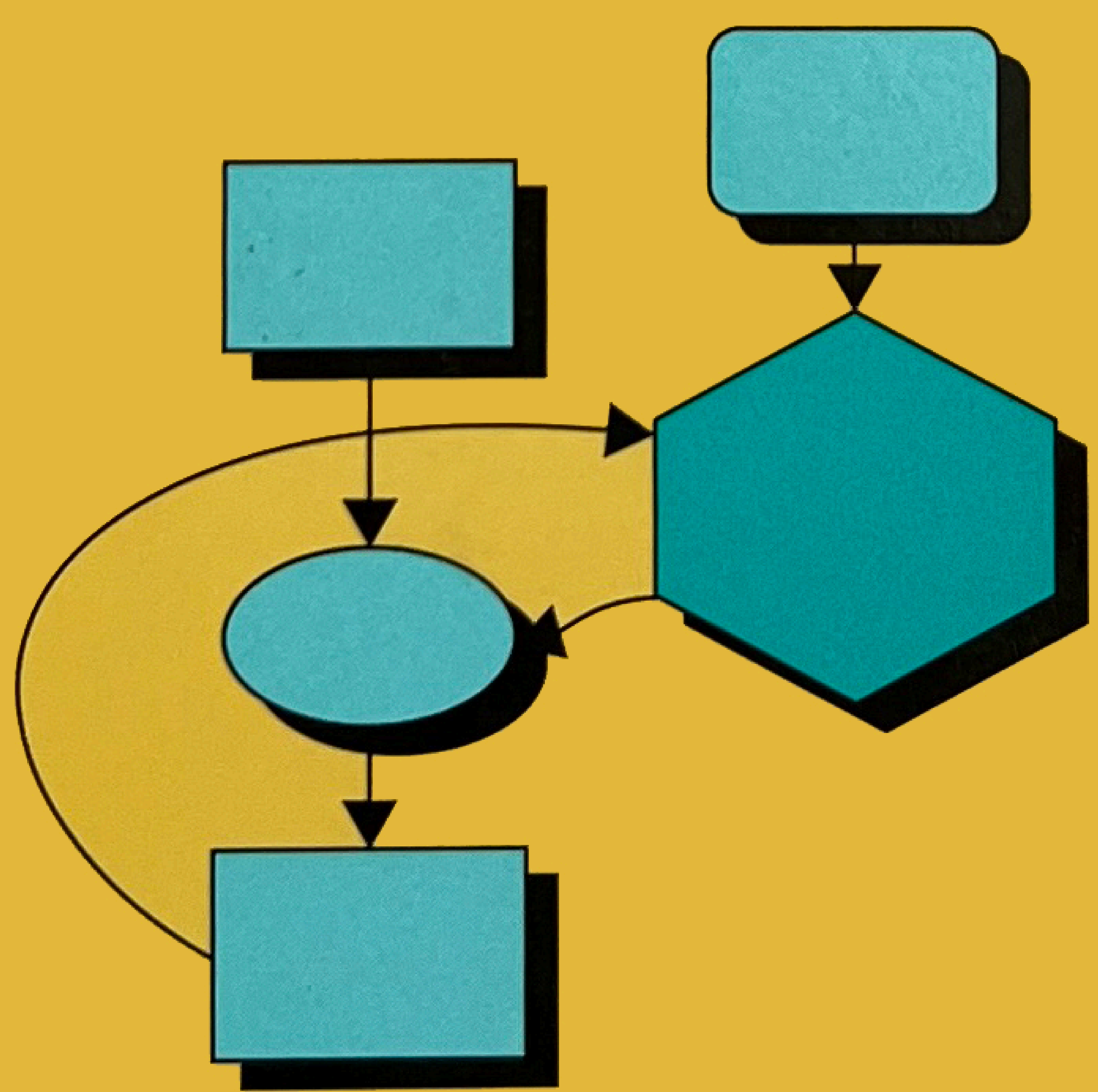
Trinity

Model-based support for multi-actor problem solving

applied to environmental problems

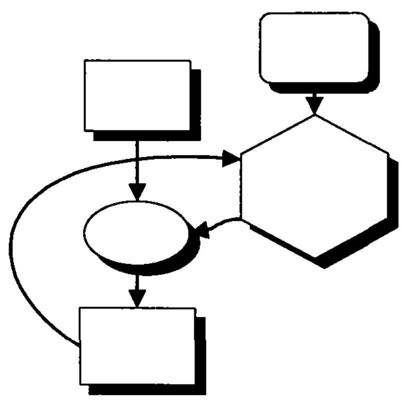


Henk B. Diepenmaat

Trinity

Model-based support for multi-actor problem solving

applied to environmental problems



Henk B. Diepenmaat

TRINITY

MODEL-BASED SUPPORT FOR MULTI-ACTOR PROBLEM SOLVING

APPLIED TO ENVIRONMENTAL PROBLEMS

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof. dr J.J.M. Franse

ten overstaan van een door het college van dekanen ingestelde
commissie in het openbaar te verdedigen in de Aula der Universiteit

op woensdag 3 september 1997 te 12.00 uur

door Hendrikus Bernardus Diepenmaat
geboren te Brunssum

Promotoren:
prof. dr J. Cramer
prof. dr B. Wielinga

Co-promotor:
prof. dr W. Harder

ISBN 90-9010837-8

Het in dit proefschrift beschreven onderzoek is mede mogelijk gemaakt door de Nederlandse organisatie voor toegepast natuurwetenschappelijk onderzoek (TNO) en het Ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieubeheer (VROM).

aan mijn vader

en

in dierbare herinnering aan mijn moeder

CONTENTS

PART I: INTRODUCTION

1. SCOPE, OBJECTIVES AND APPROACH 1

1.1 SCOPE 1
1.2 RESEARCH OBJECTIVES 3
1.3 RESEARCH APPROACH 4
1.4 OUTLINE OF THE DISSERTATION 9

PART II: PHILOSOPHICAL BACKGROUND

2. PROBLEM SOLVING, A PHILOSOPHICAL INQUIRY 15

2.1 INTRODUCTION 15
2.2 PROBLEM SOLVING 16
2.2.1 A generic model of intentional activities 17
2.2.2 Another interpretation of problem solving 19
2.3 THE MODEL OF INTENTIONAL ACTIVITIES: CRITIQUES AND ANSWERS 21
2.4 PHILOSOPHY 33
2.4.1 Dualism, relativism and perspectives 33
2.4.2 Simple answers to difficult questions 40
2.5 SYSTEMIC REFLECTION 41
2.6 MODEL-BASED SUPPORT FOR PROBLEM SOLVING 43
2.7 DISCUSSION AND CONCLUSIONS 45

PART III: THEORY

3. KNOWLEDGE DISTRIBUTIONS AND KNOWLEDGE PROCESSES (CONSEQUENCES FOR PROBLEM SOLVING) 49

3.1 INTRODUCTION 49
3.2 KNOWLEDGE DISTRIBUTION SPACE (KDS) 50
3.3 THEORETICAL FEATURES OF KDS 56
3.4 A TYPOLOGY OF ENVIRONMENTAL SITUATIONS OF CONCERN 64
3.5 METHODOLOGICAL CONSEQUENCES FOR PROBLEM SOLVING 70
3.6 DISCUSSION AND CONCLUSIONS 75

4. A GENERIC THEORY OF QUALITATIVE MODELLING 79
4.1 INTRODUCTION 79
4.2 MODELS, REFERENTS AND MODEL RELATIONS 81
4.3 MODELLING 85
4.3.1 Modelling processes and modelling steps 86
4.3.2 Parallel steps 90
4.3.3 Multi-referent steps 92
4.3.4 Multi-representation steps 94
4.4 SEQUENCING MODELLING STEPS 96
4.4.1 Non-primitive modelling processes 96
4.4.2 Typical sequences: modelling strategies 100
4.5 DISCUSSION AND CONCLUSIONS 112

PART IV: METHODS

5. *TRINITY*: MODELLING METHODS TO SUPPORT MULTI-ACTOR PROBLEM SOLVING 119
5.1 INTRODUCTION 119
5.2 PHILOSOPHICAL AND THEORETICAL BASIS OF *TRINITY* 120
5.3 THE *TRINITY* APPROACH: A BIRD'S EYE VIEW 123
5.3.1 Models in *Trinity* 123
5.3.2 The problem-solving process 125
5.4 THE *TRINITY* MODELLING LANGUAGE 130
5.4.1 Specify qualities of interest (Step 1) 130
5.4.2 Select symbols to refer to qualities (Step 2) 133
5.4.3 Design ways to represent complex model relations (Step 3) 134
5.4.4 Design modelling steps to change model relations (Step 4) 148
5.5 *TRINITY* MODELLING STRATEGIES 163
5.5.1 Parallel strategies 163
5.5.2 Multi-referent strategies 172
5.5.3 Multi-representation strategies 177
5.5.4 Compound strategies 178
5.6 USING *TRINITY*: PRACTICAL GUIDELINES 179
5.6.1 A global outline of the *Trinity* process 180
5.6.2 Reflection on the practical guidelines 186
5.7 MODES IN USING *TRINITY* 188
5.7.1 Isolated use, hidden use and participative use 188
5.7.2 Trouble-shooting, back-casting and trick-exploiting modes 189
5.7.3 Developing several perspectives 190
5.8 DISCUSSION AND CONCLUSIONS 191

PART V: EXPERIMENTS

6. INTRODUCTION TO THE EXPERIMENTS 197

6.1 INTRODUCTION 197
6.2 THE PROBLEM CONTEXTS 197
6.3 DIFFERENCES BETWEEN THE PROBLEM CONTEXTS 199

7. INDOOR ENVIRONMENTAL PROBLEMS 203

7.1 INTRODUCTION 203
7.2 A GENERIC "AS IS" MODEL OF PROBLEMATIC INDOOR ENVIRONMENTS 205
7.3 SPECIALISING THE GENERIC "AS IS" MODEL TO SPECIFIC SOURCES AND AGENTS 208
7.4 GENERIC SCRIPTS TO TAKE AWAY (SUB-CLINICAL) HEALTH EFFECTS 213
7.5 USING THE MODELS IN INDOOR ENVIRONMENTAL PROBLEM SOLVING: A SKETCH OF A MODEL-SUPPORTED INFRASTRUCTURE 218
7.6 DISCUSSION AND CONCLUSIONS 222

8. VOLATILE ORGANIC COMPOUNDS 2000 227

8.1 INTRODUCTION 227
8.2 A *TRINITY* INTERPRETATION OF THE PROBLEM CONTEXT 228
8.3 THE KNOWLEDGE ACQUISITION PROCESS 231
8.4 FROM PROBLEM AREAS TOWARDS A *TRINITY* MODEL 235
8.5 FROM *TRINITY* MODEL TOWARDS RECOMMENDATIONS 237
8.6 DISCUSSION AND CONCLUSIONS 242

9. THE STRATEGIC CONFERENCE "BUILDING AND DEMOLITION WASTE" 245

9.1 INTRODUCTION 245
9.2 A GLOBAL "AS IS" MODEL OF THE BUILDING AND DEMOLITION SECTOR 248
9.3 "TO BE" MODELS OF THE BUILDING AND DEMOLITION SECTOR 250
9.4 THINKING ABOUT THE TRANSITION: FEASIBLE SCRIPTS 255
9.5 DISCUSSION AND CONCLUSIONS 259

10. DISCUSSION AND CONCLUSIONS OF THE EXPERIMENTS 263

10.1 INTRODUCTION 263
10.2 USE AND ADDED VALUE OF *TRINITY* 263
10.2.1 Use of *Trinity* methods: rules of thumb 263
10.2.2 Use in practice 268
10.2.3 Added value 269
10.3 CONCLUSIONS 271

PART VI: GENERAL DISCUSSION AND CONCLUSIONS

11. GENERAL DISCUSSION 275

11.1 INTRODUCTION 275
11.2 THE BASIS OF THE *TRINITY* METHODOLOGY 275
11.3 KEY FEATURES OF *TRINITY* 277
11.4 REFLECTIONS ON THE *TRINITY* METHODOLOGY 283
11.4.1 Exposing the backbone 283
11.4.2 Positioning *Trinity*: mainstream paradigms for dealing with complexity 288
11.5 ADDED VALUE OF *TRINITY*: A METHODOLOGICAL POINT OF VIEW 294

12. CONCLUSIONS, RECOMMENDATIONS AND FUTURE RESEARCH 297

12.1 GENERAL CONCLUSIONS 297
12.2 RECOMMENDATIONS 299
12.3 FUTURE RESEARCH 300

APPENDIX A: A COMBINATORIAL SCHEME OF MODELLING PROCESSES 303

APPENDIX B: *TRINITY* MODELLING CONVENTIONS:
PRAGMATIC CORRECTNESS 305

APPENDIX C: GENERIC SYSTEMS SPACE 313

APPENDIX D: EXPLANATION OF THE NAME "*TRINITY*" 315

REFERENCES AND LITERATURE 317

SUMMARY 323

SAMENVATTING 329

CURRICULUM VITAE 335

DANKWOORD 337

LIST OF FIGURES 341

LIST OF TABLES 345

INDEX 347

PART I: INTRODUCTION

CHAPTER 1

SCOPE, OBJECTIVES AND APPROACH

1.1 SCOPE

Our contemporary society is complex. The activities of different actors become more and more inter-linked and intertwined, and depend on each other to a degree that has never been observed before. These actors form complex networks (for example, production and consumption networks). One outcome of this development is the growing importance of communication and information services, sciences and technologies.
Perhaps by virtue of the level of specialisation and co-operation that is implied by these networks, modern life for many of us is at a level of prosperity that is unprecedented. Although sharing this standard of life proves to be difficult, and keeping in mind that prosperity is different from well-being, most of us will agree that many of the features of contemporary society are an improvement compared to former times.
The down side is, however, that many of the problems that contemporary society (including both social and business life) is facing, are very complex and difficult to solve. Problem-solving processes are severely influenced, and often even dominated, by the fact that many different parties are involved. Each of these parties is acting in line with a distinct, albeit not always clear, set of possibilities, points of view and interests. The processes are *multi-actor* in nature.

In order to improve multi-actor problem situations, many different parties have to make complex decisions affecting their own position. Simultaneously, these decisions should contribute to the improvement of the overall situation. The problem context as a whole, however, is typically highly complex and only partially understood. Bearing in mind that the success of individual actions highly depends upon the actions of other actors participating in the problem context, it is not surprising that multi-actor problems time and again prove to be extremely difficult to solve.

A manifest and urgent category of multi-actor problems is constituted by "environmental problems". In environmental problems, consumers, governments, scientists of various disciplines, intermediate organisations, non-governmental organisations and many other actors may play a role in the problem context. Indeed, the research described in this thesis originated in an attempt to specifically support environmental problem solving. However, during this work the insight emerged that "environmental" is not a fruitful restriction because the conceptual framework presented is mainly built upon notions that are not specific for environmental problems. This observation points to the way in which environmental problems and environmental problem-solving processes are being interpreted in our research: as a special case of a more general type of multi-actor intentional activities.

As a result of the developments outlined above, several researchers and practitioners have been thinking, talking and writing about solving complex problems and supporting complex decision-making processes. Examples of contributions in this area are[1] the Soft Systems Methodology [Checkland (1981), (1989a), (1989b), Checkland and Scholes (1991)], supporting team learning by means of Systems Dynamics [Vennix (1996)], Strategic Options Development and Analysis [Eden (1989)], the Action Workflow Approach, and the SADT or IDEF approach. These examples show a large diversity in philosophical stances, disciplinary backgrounds, methods and means (see, for example, [van Lierop (1997), Hofman (1994)]). This makes it difficult to address such a topic while meeting scientific conventions including covering a substantial part of the relevant scientific literature, tracing all the concepts and ideas that are being adopted and adapted back to their original source, and developing some genuinely new thoughts. It may even be argued that the notion of complex problems involving many actors, in all its varieties, is far too vague and diffuse to be the central research topic of a scientific dissertation.

Notwithstanding this, the subject of this dissertation is a methodology that is intended to *support multi-actor processes in general*. The name of this methodology is *Trinity*[2]. The reason for addressing this topic, in spite of the difficulties mentioned earlier, is pragmatic: it is important more than ever to be able to deal with multi-actor problems. They are being experienced and are knocking at our doors in an ever increasing rate and intensity. In line with the observation that in multi-actor situations the problem context is in general only partially understood (see above), the methodology will be directed at supporting the very process of obtaining such an understanding.

As mentioned before, "multi-actor processes" constitute quite a heterogeneous collection. Examples that fall within this category are policy framing, implementation and evaluation; managing knowledge-intensive multi-actor processes (knowledge management); analysing and changing organisations (business process redesign, workflow management), et cetera.
The obvious benefit of a *generic* approach for multi-actor processes is that, at least in principle, it can be used to provide support in many different situations (i.e. a high reusability). In addition, especially in complex situations this might minimise the necessity to change methodology during the overall process. A potential danger in pursuing

---

[1] An overview of several methods is presented in [Rosenhead (1989)].

[2] The name *Trinity* is explained in Appendix D.

genericity is, however, that the well-known problem of balancing between breadth of scope and level of support applies. Genericity may well introduce triviality, especially in the case of a rather heterogeneous research subject. Nonetheless, we considered the benefits mentioned above important arguments to initiate a research focused on supporting multi-actor processes *in general.*

1.2 RESEARCH OBJECTIVES

The overall goal of our work was to develop a methodology designed to support multi-actor processes in general. An important problem in these processes is that in many cases the multi-actor problem context as a whole is only partially understood. For this reason, we will focus on means to support the development of a coherent understanding of multi-actor problem contexts; a central role is reserved for modelling methods.

In line with this, the central research question is:

> Is it possible to design *modelling methods* that specifically support problem-solving processes in multi-actor situations?

From this central research question, two specific sub-questions can be derived.

Preferably, modelling methods should be built upon a solid foundation. Therefore, the first sub-question is:

> Is it possible to develop a *philosophical and theoretical basis* that positions the central concepts of this dissertation ("problem solving", "multi-actor", "modelling", and "model-based support") and thereby provides a foundation for the envisaged modelling methods?

On the other hand, the true value of methods is to be established in practice, by means of experiments. Therefore, the second sub-question is:

> What can be said about the *use* and *added value* of the envisaged methods, when applied in real-world multi-actor problem solving?

The relation between the research questions is presented in figure 1.

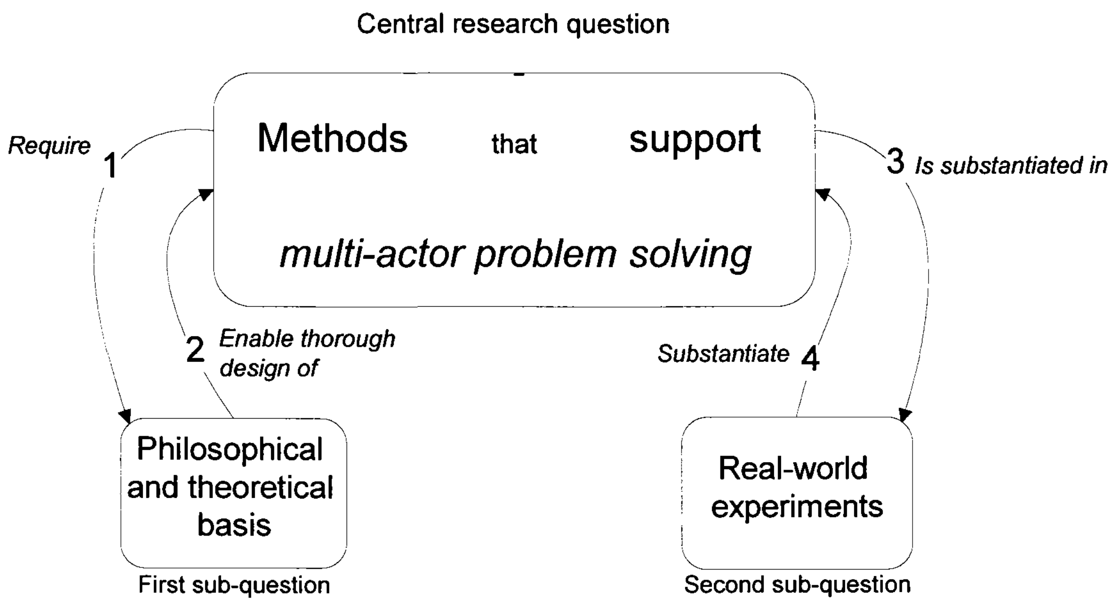


Figure 1: The relations between the research questions.

Answering these questions results in the *Trinity* methodology: a conceptual vocabulary that fleshes out the notion of "model-based support for multi-actor problem solving".

1.3 RESEARCH APPROACH

So far we have used the word "methodology" several times already, albeit in a rather intuitive manner. In the context of this dissertation, however, this word has a specific meaning. The research approach underlying this dissertation is derived from a conceptual framework called the *methodological pyramid* [Wielinga et al. (1989)].

The methodological pyramid is a triangle consisting of five layers: *world view*, *theory*, *methods*, *tools* and *use* (from bottom to top). Lower layers are viewed as building blocks that support the layers on top of it. In addition, *using* the methodology (i.e. activities in the *use* layer) results in feedback to the other four layers.

We have followed a research approach that adopts the five layers of the methodological pyramid. The reason for this is that these five layers clarify concisely the essence of methodological research. In contrast with, for example, experiments or case studies (the *use* layer), tools-directed research (the *tools* layer), methods-directed research (the *methods* layer), theoretical research (the *theory* layer) or philosophical research (the *world view* layer), a characteristic feature of methodological research is that it explicitly addresses *all* *five layers*[3]. This is the reason why we call the research, presented in this dissertation, *methodological research*, and *Trinity* a methodology: research activities took place at *all five layers*, and (consequently) *Trinity* encompasses all five layers.

We slightly re-interpreted the meaning of the five layers of the original description [Wielinga et al. (1989)], resulting in the following descriptions:

In the *philosophy* layer (originally: *world view* layer), emphasis is on the most general principles and assumptions underlying the methodology. These fundamental principles and assumptions, although in many cases not stated explicitly, determine rather fundamental issues and features of the methodology as a whole (in our case, for example, the interpretation of problems and problem-solving processes).

In the *theory* layer, theories (conceptual structures, see, for example, [Kangassalo (1990)]) are developed and described that elaborate on the *philosophy* layer, and simultaneously provide a conceptual framework that justifies and positions the other three layers (methods, tools, and use).

The *methods* layer encompasses generic "prescriptions" of what can be done in order to successfully finish an application of the methodology in specific situations. The methods layer makes operational the theory layer in that it provides a *conceptual toolbox* for the user of the methodology.

The *tools* layer supports operational use of the methods: it provides the *operational toolbox* of the methodology. Tools are directed at making the application of the methods (the *conceptual* toolbox, shown above) more *efficient*. An example is a computer programme that enables one to use the methods.

Finally, the *use* layer encompasses the use (application) of the methodology in specific situations, in attempts to achieve a goal of the type that the methodology is intended to support.

In addition to these re-interpretations, rather than assuming a *hierarchical* supportive relation between the five layers (as was the case in the original pyramid), we consider the supportive relation between the five layers to be *symbiotic*: all five parts should be present (and every layer therefore supports all the other ones). For example, the *use* layer supports the other four layers, as does the *philosophy* layer. This implies that the *philosophy* layer supports the *use* layer, as well as the other way around.
During the research described in this thesis, it was indeed the case that *use* of the methodology resulted in feedback towards the other four layers (this is the dominant

[3] This implies that developing methods is quite different from developing a methodology: methodological research encompasses activities at *all five layers*, and therefore *encompasses* rather than *equals* developing methods.

feedback mechanism in the original description of the pyramid). However, the feedback mechanisms were more complicated. A major step forward in any layer typically resulted in consistency adaptations, or at least re-considerations, in each of the other layers. In addition, feedforward mechanisms could be recognised: a major advance in one of the layers typically induced advances in other layers as well (although this may well be caused by the fact that *Trinity* was *in statu nascendi* during this research because in the case of a mature methodology feedback is likely to be the more dominant mechanism).
In combination, these adaptations result in the picture presented in figure 2.

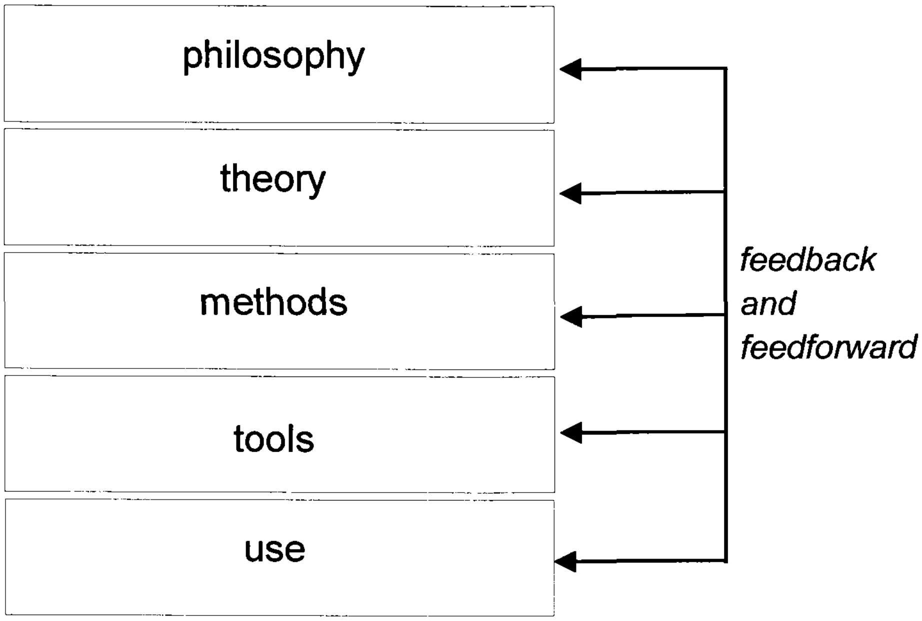


Figure 2: The five symbiotic elements of a methodology (derived from the methodological pyramid [Wielinga et al. (1989)]).

From a historical point of view, the research took place in the following way. At the start, we recognised an urgent need to improve (make both more effective and efficient) environmental problem-solving processes. As a first step, we designed a premature version of the *Trinity* modelling language (which in terms of figure 2 must be positioned in the *methods* layer), and we implemented a simple graphical aid in a flowcharting programme (which is an element of the *tools* layer). Simultaneously, we executed some finger experiments [see Vermeulen and Diepenmaat (1993)] that fuelled our conviction that we were on the right track.

Although intuitively appealing, this rudimentary version of *Trinity* resulted in a number of serious questions with respect to the basis underlying these modelling methods. In other words, the methods were in desperate need of a philosophical and theoretical basis (see also the first sub-question of section 1.2). The bits and pieces of this basis that were already present were elaborated upon: both layers were filled in.

This in turn induced a complete redesign of the *Trinity* modelling language. After we had completed this redesign, the tool was adapted. Together, the results of these research activities encompassed the top four layers of figure 2. In this way, the machinery to more rigidly test the methodology was developed.

This testing was done by means of conducting three quite different experiments in multi-actor situations, all three in an environmental problem context (environmental problems are typical and manifest examples of multi-actor problems).

This dissertation describes the results of these research activities[4]. Figure 3 presents an overview of these research activities. The vertical arrows refer to a time axis on a rather large scale. The horizontal arrows emphasise the developments on a smaller time scale. The bi-directional horizontal arrows emphasise that intensive interaction existed between the various research activities.

[4] Only four layers are described. The tools layer is omitted. The reason for this omission is that, as tool, we adapted a flowcharting programme in such a way that it became dedicated to *Trinity* symbols. We hardly consider this to be a research activity. In the future, however, further tool research is anticipated, which will be directed at offering support for using the *Trinity* methodology in different ways.

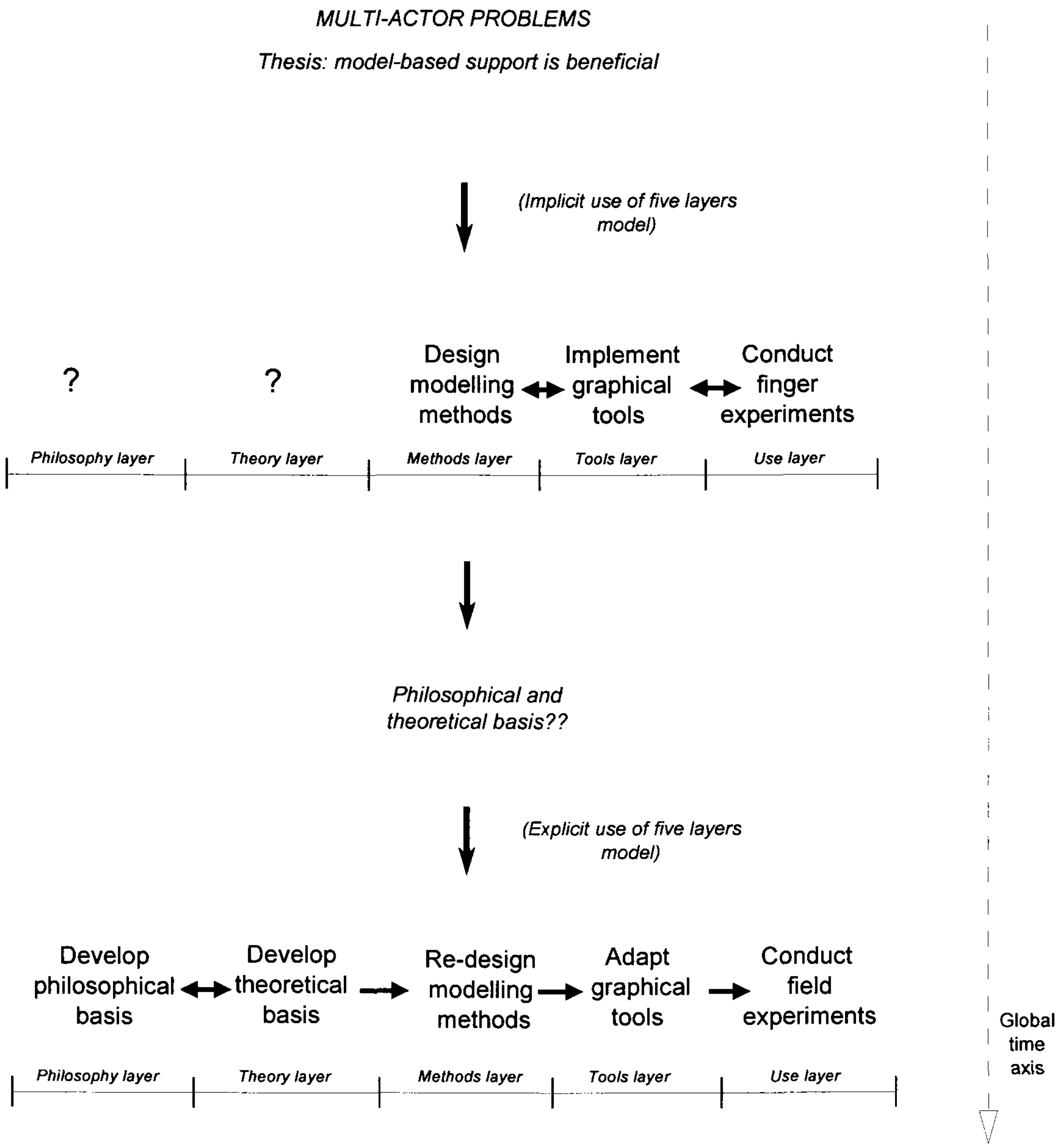


Figure 3: An overview of the research approach.

1.4 OUTLINE OF THE DISSERTATION

The dissertation consists of six parts. Each of these parts will be introduced below.

Part I: Introduction provides the scope, objectives and general approach of the work described in this thesis.

Part II: Philosophical background encompasses the *philosophy* layer of the *Trinity* methodology. This part consists of one chapter:

Chapter 2: Problem solving (a philosophical inquiry)

In this chapter the central concept of this dissertation, namely problem solving, is discussed in a philosophical context and is given a specific meaning.
The resulting definitions of problems and problem solving are founded on a knowledge theoretical line of reasoning. Problems are regarded as entities, experienced by specific persons or groups, and (therefore) exist in the realm of human concern. In line with this, problem solving encompasses the process of obtaining a potential for human action, of which the actual execution is believed to result in a reduction of this concern. Although subsequently *exploiting* this action potential (i.e. taking action) may take place (among others) in a physical environment, problem solving is considered to be a preparatory stage with respect to taking action, that takes place predominantly in a mental domain. This implies that the definitions used in this thesis differ, for example, from definitions customary in the environmental sciences, where environmental problems are positioned in the physical environment, and problem solving encompasses actually improving this physical environment. In order to avoid epistemological confusion, all the claims to be made in this thesis must be considered within the context of the knowledge theoretical definitions to be presented and substantiated in this chapter.

Part III: Theory encompasses the *theory* layer of the *Trinity* methodology. More specifically, this part consists of two chapters:

Chapter 3: Knowledge distributions and knowledge processes: consequences for problem solving

In this chapter, on the basis of a rather abstract theory of knowledge distributions, the notions of multi-actor situations and multi-actor processes are elaborated upon in a systemic manner. In addition, a typology of problems on the basis of knowledge distribution criteria is presented. The *Trinity* methodology is specifically directed at offering support in one of these problem types.

Chapter 4: A generic theory of qualitative modelling processes

This chapter presents a generic theory of qualitative modelling processes. In the Methods part of this dissertation (Chapter 5), this theory will be used to design the *Trinity* modelling language.

Part IV: Methods presents the *Trinity* modelling methods. The results of the *philosophical background* and *theory* part are used as a basis to design these methods. This part consists of one chapter:

Chapter 5: *Trinity*, modelling methods to support multi-actor problem solving

This chapter presents the *Trinity* modelling methods. The core element of these methods is a qualitative modelling language, designed in full compliance with the theory of qualitative modelling processes (Chapter 4), that is directed at supporting multi-actor (Chapter 3) problem-solving processes (Chapter 2).

Part V: Experiments describes the results of three (semi-)field tests in which *Trinity* was applied. Empirically "proving" a methodology directed at supporting real-world multi-actor problem solving processes is a difficult task. A situation in which both an experimental setting and an independent control setting are present is extremely difficult to realise in multi-actor problem solving. And even then, this would result in only *one* application of the methodology and *one blanco* (the process without applying the methodology). Multi-actor processes are different from, for example, (many) natural scientific phenomena in this respect. Notwithstanding these problems, we have conducted several experiments that, although perhaps unable to completely confirm or refute the methodology in a very strict scientific meaning, will allow us to substantiate or decline the claim that *Trinity* offers model-based support for multi-actor problem solving.

The three experiments took place in environmental problem contexts. In addition, an introduction to and a discussion of the experiments are provided. The Experiments part therefore consists of five chapters:

Chapter 6: Introduction to the experiments
In this chapter, the rationale for the selection of the three experiments as described in the following three chapters, as well as their differences, are discussed.

Chapter 7: Indoor environmental problems
In this experiment, a rather diagnostic use of *Trinity* is presented. The problems of concern are characterised by their confronting nature and a rather short time scale for remedial action.

Chapter 8: *VOC2000*
In this experiment, emphasis is on the first steps in an attempt to improve the VOC2000 programme: a national environmental policy process based on an agreement between the Dutch government and corporate sectors to considerably reduce the emission of volatile organic compounds (VOCs). The problems of concern are characterised by a time scale of one year and up.

Chapter 9: The strategic conference *Building and demolition waste*
In this experiment, a strategic conference is reviewed from a *Trinity* point of view. Emphasis is on a future situation. The problem of concern is characterised by a time scale of several decades.

Chapter 10: Discussion and conclusions of the experiments
In this chapter, the experiments are discussed in combination, and general experimental conclusions are drawn.

Part VI: General discussion and conclusions presents a general discussion of the *Trinity* methodology and the conclusions of this research as a whole. It consists of two chapters:

Chapter 11: General discussion
In this chapter, several aspects of the *Trinity* methodology are discussed in a comprehensive fashion. First, a concise review of the fundaments of the *Trinity* methodology is presented. Second, several key features of the methodology are discussed. Third, the "backbone" is exposed: a theory that ties together the different parts of this dissertation into one underlying theory. Fourth, *Trinity* is positioned in the field of some mainstream paradigms in dealing with complexity. Finally, the added value of using *Trinity* is discussed from a methodological point of view.

Chapter 12: Conclusions, recommendations and future research
In this chapter, the central research questions (section 1.2) are answered. In addition, general recommendations and a research agenda for the future are presented.

PART II: PHILOSOPHICAL BACKGROUND

CHAPTER 2

PROBLEM SOLVING, A PHILOSOPHICAL INQUIRY

2.1 INTRODUCTION

The *Trinity* methodology is intended to support multi-actor problem-solving processes. But what is problem solving? The answer to this question is not an easy one. A first requirement, therefore, is that the concepts "problem" and "problem solving" are given a clear meaning within the boundaries of this dissertation. For this reason, this chapter is devoted to an in-depth investigation of the concept of problem solving. This, in addition, enables us to flesh out the notion of model-based support for problem solving.
During our attempts to answer this question, we touched upon several philosophical disputes. In addition, we devoted a good deal of attention to philosophical backgrounds. For this reason, we have subtitled this chapter "a philosophical inquiry" and this part of the dissertation "Philosophical background". We did not attempt, however, to trace all historical antecedents, to assign proper credit for originality, or to present a complete overview of the many philosophers that have addressed the central issues of this chapter far more profoundly. We are not philosophers. We merely acknowledge the fact that every intellectual endeavour has philosophical roots. This chapter is our attempt to account for them.

First, different interpretations of problem solving are discussed. These interpretations are shown to be described by one *generic model of intentional activities*. This generic model allows us to clearly define the notion of problem solving that we will adhere to in this dissertation.
Subsequently, several critiques towards the model of intentional activities, that is at the basis of both the *Trinity* methodology and our interpretation of problem solving, will be answered.
After that, the philosophical stance underlying the model of intentional activities as well as our interpretation of problem solving is elaborated upon.
Finally, on the basis of this philosophical stance, the notion of model-based support for problem solving is developed.

2.2 PROBLEM SOLVING

Different interpretations of problem solving exist. In this section, three of them will be discussed: a behavioural interpretation and two cognitive interpretations.

Problem solving is acting

A broadly accepted common-sense interpretation of problem solving is that it amounts to actually changing a real-world situation of discontent into a better situation. In short: *problem solving is acting.*

Consider, for example, the situation in which a house is on fire. In the behavioural interpretation, problem solving amounts to extinguishing the fire.

This behavioural interpretation neglects the fact that intentional acting is preceded by thinking. This is an omission, as features of this thinking process are important characteristics of problem-solving processes.

Problem solving is decision making

Another interpretation of problem solving, one that is particularly manifest in management science, is that it amounts to decision making. This interpretation is, for example, at the basis of the "bounded rationality" concept of Simon. Simon states that one, if not the most important, aspect of problem solving is rational decision making. This typically involves a series of three steps: 1. list alternative strategies (a list that bounds the choices, hence *bounded*); 2. determine consequences of implementing each of the strategies; and 3. comparatively evaluate consequences (hence *rationality*). In this interpretation, problem solving is a rational process of selecting between alternatives [Simon (1957) p. 198, Simon (1976) p. 67].

In contrast with the first interpretation, this is a cognitive interpretation. Indeed, some cognitive scientists even describe *knowledge* in terms of rational decision making. For example, Newell defines knowledge as the potential to select actions that lead towards a goal. In his landmark article "the knowledge level", Newell elaborates upon this, resulting in the "principle of rationality": "If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action" [Newell (1982) p. 102].

The stance “problem solving is decision making” is rather popular, and has resulted in a number of different approaches that support decision making (see, for example, [Hendriks et al. (1992)], [Shaw and Fox (1993)], [Singh, Bennavail and Chen (1992)], [Badiru, Simin Pulat and Kang (1993)] and many more).

Problem solving is generating alternatives for taking action

Also the interpretation "problem solving is decision making" can be criticised. In many cases attention is focused on too limited a set of alternatives in too early a stage of the problem-solving process. Although the choice is "rational", it is far too limited. This phenomenon, also known as “premature closure”, is a serious draw-back in complex problem-solving processes. Some authors mention that people tend to "stick" to only one solution or a few solutions [see, for example, Beach (1990) Chapter 1]. These

considerations make clear that it is well advised to stress the importance of *generating* alternatives, (i.e. to explore the solution space) in addition to selecting between them (i.e. to focus on specific parts of this solution space).
Winograd and Flores [1986] argue that, although the "bounded rationality" approach does not assume that a decision maker can evaluate *all* alternatives, it takes for granted that these alternatives exist. According to them, "The hard part is understanding how the alternatives relevant to a given context come into being" [Winograd and Flores (1986) p. 146].
These critiques point out another cognitive interpretation (that in a sense is the inverse of the decision making interpretation): Problem solving is generating alternatives for taking action.

In our research, we will use an interpretation of problem solving that is different from, or rather a mixture of the three discussed above. It is based upon the more general notion of an *intentional activity*. According to our point of view, the *action*, the *decision making* and the *generate alternatives* interpretation emphasise different aspects of the concept of an *intentional activity*. In the next section, a generic model of intentional activities will be presented.

2.2.1 A generic model of intentional activities

Many different attempts have been made to describe intentional activities as a sequence of simpler steps, and this in a large diversity of domains.
Examples from policy science and management science are: the policy cycle; Checkland's experience-action cycle [Checkland (1981)]; the group communication cycle; Schön's spiral model of appreciation, action and reappreciation [Schön (1983)]; the participative modelling cycle in policy analysis [Geurts and Vennix (1989)]; and the learning cycle of Kolb [Kolb (1984)]. Many of these models are called *cycles*, as they are assumed to reappear in a never ending story. In this respect, the notion of spiralling, as used by Schön, is perhaps more to the point, as at the end of each cycle a *new* situation is at stake, rather than the original one. As Heraclitus said: "We cannot step into the same river twice".
Also in engineering sciences, for example in informatics, linear waterfall approaches (for example, *a sequence* of analyse, design, code, maintain) are being replaced more and more by iterative approaches in which *several* cycles (for example, analyse, design, prototype, evaluate) finally result in a satisfactory product (for example a computer programme); see, for example, [Tolido (1995)] and [Boehm (1988)].

Inspired by these many attempts to develop stage-based, spiral models[5], we want to present yet another model. This model does not add detail to existing models. Rather, it is simpler in that it neglects domain-specific features altogether. We call it a *generic model of intentional activities* because in principle it models *any* intentional (purposeful)

[5] Our model is a "recursive stage-based spiral". This will be explained further on.

activity, from doing the shopping to attaining a sustainable society. The model consists of four stages, and is presented below (figure 1):

stage 1: acknowledge situation of concern
stage 2: construct perspective

stage 2a: analyse situation "as is"
stage 2b: synthesise script (a plan) for taking action
stage 2c: predict situation "to be"

stage 3: implement script
stage 4: evaluate situation "to be"

Figure 1: A generic model of intentional activities.

The basic structure of the model is the pattern **think**, **act**, **check**. In this respect, it resembles many of the examples of stage-based models mentioned above. We consider this to be a strong point: a large deviation from this well-founded and intuitively appealing pattern should raise suspicion, rather than applause.

At a more detailed level, the central concept of our model is the *perspective*. According to our theory, a perspective is a body of knowledge that guides and motivates intentional action (see also Chapter 3). It can be thought of as a conglomerate structure of three tightly connected mental models: a *descriptive* model of the "as is" situation; a *prescriptive* model (a script) for taking action (intervention) in this "as is" situation; and a *predictive* model of the (better) "to be" situation, expected to result from actually implementing (executing) the script. These three parts are the outcome of stages 2a-c, respectively. Stage 2, therefore, is the *perspective construction stage*.

The threefold structure of a perspective is an important feature of the model: in order to be able to act intentionally (stage 3), an actor must possess *all three parts*. An *actor* is defined as the agent, engaged in an intentional activity. To be even more specific: an actor is an agent who acts intentionally. The difference between the descriptive ("as is") part and the predictive ("to be") part ***motivates*** taking action (as this difference is considered to be an improvement according to the actor's value system: it explains the "why" of the action). The prescriptive (script) part ***guides*** this action (it tells the "how"). Conversely, an intentional *action* (stage 3 of an intentional *activity*) can be defined as an action that is motivated and guided by a perspective[6]. The notion of a perspective, and the referents it models, are presented in figure 2.

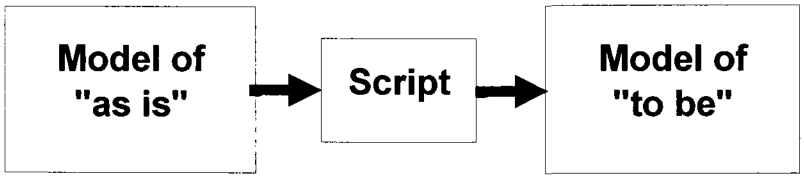


Figure 2: A perspective is a mental model consisting of three tightly coupled parts.

2.2.2 Another interpretation of problem solving

The relation between the generic model of intentional activities and the three interpretations of problem solving (as *acting*, as *decision making* and as the *generation of alternatives*) is as follows. In the first interpretation (problem solving is acting), the problem-solving process matches with stage 3: execute script. In the second interpretation (problem solving is decision making), the problem-solving process is part of stage 2. More specifically: if during stage 2 several alternative perspectives are at stake, a decision has to be made. In the third interpretation (problem solving is generating alternatives) the problem-solving process also is part of stage 2. In this case, however, focus is on the *construction of perspectives*, rather than on *deciding among them*. This shows that all three interpretations are captured by the generic model of intentional activities.

The notion of problem that we will use in our research is not unlike the one of Newell and Simon [Human Problem Solving (1972) p. 72-73]: "A person is confronted with a problem when he wants something and does not know immediately what series of actions he can perform to get it...".
We call a *problem* a situation in which an actor is *willing* to act intentionally in order to improve his or her situation. A rudimentary perspective has emerged. This means that this actor has passed stage 1. However, this actor does not possess the knowledge to *do* so. A

[6] Note that, in a sense, the definitions of *perspective* and *intentional action* are circular: a perspective is a body of knowledge that enables intentional action; and an intentional action is an action motivated and guided by a perspective. Likewise, the definitions of *actor* and *intentional action* are circular: an actor is an agent engaged in an intentional action, and an action is the activity of an intentional actor. This resembles, for example, the definitions of paradigm and (social) group in the work of Kuhn: a group is defined by a shared paradigm, and a paradigm is shared by a group. Also the notion of "research programme", as used by Lakatos, exhibits this circularity. This circularity, however, is not a problem from a pragmatic point of view, provided that *either* the action *or* the perspective (either the actor or the action) is at stake. As for non-pragmatic solutions, we do not know of a way to avoid this kind of circularity.

lack of knowledge, *the absence of an appropriate and sufficiently complete and/or clear perspective*, prevents this actor from acting intentionally. The actor is not yet ready to enter stage 3.
In line with this, we call *problem solving* the (at least partially) cognitive process of filling in this lack of knowledge:

problem solving is perspective construction.

This corresponds with stage 2 of the generic model, and (therefore) encompasses the two cognitive interpretations of problem solving (*generating* and *deciding among* action potentials) mentioned above. A difference[7] with Newell and Simon's definition is that the strongly situated, context-dependent and even emergent nature of the process of *generating* alternatives (as emphasised by Winograd and Flores) is considered to be an important factor influencing the perspective construction process. Indeed, we position the problem-solving process in between the acknowledgement stage (which from the point of view of the actor involved is emergent, situated, context-dependent) and the implementation stage. Further on in this chapter we will elaborate upon the definition of problem solving as perspective construction from a philosophical point of view.
Perspectives resemble the notion of schemas (and related notions like frames and scripts); a notion that was and is rather influential in cognitive science and artificial intelligence (see, for example, [Minsky (1975)], [Schank and Abelson (1977)], [Holland, Holyoak, Nisbett, and Thagard (1987) p.12]). This notion emphasises the availability of mental "clusters" that describe typical situations (rather than individual situations) and that can be accessed and used to generate plausible inferences and problem solutions.
A perspective is also a mental body of knowledge. At the start of a problem-solving process, the perspective may be incomplete and/or too abstract to motivate and guide the taking of actions. At the end of the problem-solving process (which can be thought of as a time series of perspectives), this perspective has developed into a solution. The result of a successful problem-solving process is a perspective that enables intentional action: both the "as is" and the "to be" parts are sufficiently elaborated to recognise some improvement, and the script is sufficiently elaborated to guide actions that implement (or at least start, cause) the transition.

According to this view, problem solving encompasses *both* the process of generating alternatives (the third interpretation of problem solving) *and* constantly deciding amongst them (the second interpretation of problem solving). It excludes the very actions, implementing the script (the first, behavioural interpretation of problem solving). These

[7] Newell and Simon do not explicitly exclude the generative nature of problem solving, but do not mention or emphasise it either (as Winograd and Flores very explicitly do in their critique on the decision-making paradigm). Therefore, we consider it correct to conclude that Newell and Simon's interpretation falls within the decision-making paradigm.

actions *are*, however, a crucial part of the intentional activity, of which the problem-solving process is a part.
Although at any point during the problem-solving process several alternative perspectives may be considered, only one perspective is allowed to result. The reason for this is that for an actor at a specific place and a specific time it is impossible to act according to more than one perspective: the respective script implementation stages (stage 3) of alternative perspectives are an "exclusive or". It is possible, though, that a perspective consists of several perspectives at a lower (systemic) level, either sequentially or concurrently ordered in time. This will be explained further on.

We think that excluding the behavioural interpretation from our interpretation of problem solving is appropriate, as a problem is said to be solved (the problem solving stops) at the very moment that a perspective is *available*: a solution is found. It is not required that the solution be *implemented*: it is sufficient that the problem owner considers this route to be *feasible*.
Consider, for example, the situation that you want to cross a river. The situation is problematic, as you want to cross, but you do not possess knowledge of a way to do so. You cannot *act* intentionally; a perspective is lacking. You start to look around and try to think of a way to cross. You stumble across a rowing boat. The moment that you recognise the boat as a means to cross the river, a perspective is available. At *that* very moment your problem is solved, and *not* at the moment that you have actually crossed the river.

2.3 THE MODEL OF INTENTIONAL ACTIVITIES: CRITIQUES AND ANSWERS

In this chapter, we will raise some serious critiques that articulate the presumed simplicity and rigidity of the model, and we will answer to them. This will further specify, clarify and substantiate our model.

Critique 1: The generic model does not incorporate notions like knowledge, thinking and learning.

Indeed, the model does not incorporate these notions, which are fundamental to problem solving. Nonetheless, the model *implies* a simple theory of knowledge, thinking and learning on the basis of perspectives. We consider this theory to be sufficiently elaborate to meet our goal: obtaining a theoretical framework for model-based support for multi-actor problem solving. This theory is presented below.

We assume that, at any specific moment in time, an actor possesses a (large) number of perspectives. For example, I know at this moment that using a corkscrew (the script) changes a closed bottle of wine ("as is") into an open bottle ("to be"). In terms of our theory, possessing knowledge amounts to possessing perspectives.

Some of the perspectives I possess I can use directly. This means: the script part can be executed; the perspectives are *situated* in time and space; the referent (for example, the bottle of wine) can be interacted with directly. A coupling exists between perspective and environment, which turns the perspective into an *action potential*. Some of the available perspectives cannot be used directly; they are not situated in time and space, but rather stand-by. The environment they refer to does not exist at that time, at that place. For example, when sitting behind my word processor, the corkscrew perspective is of no direct use, as normally I do not have bottles of wine around when I am working. The "type words" perspective, however, I can use directly.

This distinction enables to describe the notions of "this", "now" and "here" in terms of perspectives: "this" is the set of perspectives that is being applied to the environment; "now" is the set of perspectives that can be used without any delay in time; and "here" is the set of perspectives that can be used without changing place. Intentional activities, as well as the notions "this", "now" and "here", are tightly coupled to actors. This is related to the notion of time as worked out by Heidegger. As Heidegger put it: “Alles Geschehende rollt aus endloser Zukunft in die unwiederbringliche Vergangenheit” [Heidegger (1992) p. 18]. The very point where this transition manifests itself he calls a “Jetztpunkt” (now-point). An intentional activity, however, is situated in both a domain that enables intentional change (for example, a physical domain, a knowledge domain, a communication domain or a combination) as well as a time domain, the first one covering the “this” and the “here” on top of the “now” of the latter one. The domain of discourse and the time domain, however, are difficult to understand in separation (as is manifested by our model of intentional activities). In terms of the model of intentional activities, the "this", "now" and "here" of actors are positioned in-between stages 2 and 3 (see also figure 3a).

In addition to *possessing* perspectives, an actor may *process* perspectives. For example, when being confronted with a bottle of milk closed with a cork, I may adapt the "bottle-wine-and-corkscrew" perspective. In terms of our theory, knowledge consists of perspectives, and thinking amounts to *processing* perspectives. This processing takes perspectives as input, and results in perspectives as output. Although we may *know* the perspective that guides and motivates this processing, the very processing itself is out of the scope of our model. Perspectives are the *input as well as the output* of mental activities.

Experiental learning (see, for example, Kolb’s [1984] model of experiental learning, and other learning models he mentions) takes place in stage 4 of our model. The situation resulting from acting is being compared with the "to be" part of the perspective. In any case, this results in new knowledge (something has been learned; a new perspective). If prediction and result *agree*, the problem owner is satisfied and the intentional activity is finished. The perspective proved[8] useful, is kept stand-by, and may be used again in the future. If prediction and result *disagree*, the original intentional activity is also finished, but new intentional activities may be called for. Two things have been learned. First, the original perspective was inappropriate and needs some processing (and should perhaps even be discarded). Second, a new perspective, consisting of the original "as is" model, a model of the actions that were executed (likely the original script part, or some deviation); and the *new* mental model of "to be", results. In terms of our theory, experiential learning is changing the perspectives (i.e. the knowledge) that one possesses on the basis of comparing results and expectations.

Other learning mechanisms (for example, inductive learning and generalisation) can be understood in terms of perspectives as well. For example, *generalisation* is the process of constructing a more general perspective, of which specific perspectives are instances. This may be beneficial if the intentions of an actor do not require the level of detail provided by the instances. In addition, the more general perspective may guide recognition of new instances. *Induction* is the process of constructing a generic perspective, that is thought to cover *all* the members of this genus, on the basis of only a limited set of perspectives that are instances of this genus. This resembles generalisation, but in addition extends the scope of the generic perspective.

Inference mechanisms can also be understood in terms of perspectives. For example, a *deduction* can take place if an actor is being confronted with a situation, that can be modelled by the "as is" part of a perspective. The actor may deduce that executing actions that are modelled by the script will result in a situation modelled by the "to be" part of this perspective. Note that this is different from a logical deduction. For example, the *fact **a*** and the *rule **a implicates b*** enable one to logically deduce the *fact **b***. In our case, however, the deduction is made with respect to the *referent* of the perspective, and not with respect to the perspective itself. A model relation (that is possibly wrong) is at stake.

In summary: Within the boundaries of our theory, possessing knowledge equals possessing perspectives; problem solving equals processing perspectives; and experiential learning is thinking at the interface of results and expectations.

Critique 2: There is only one generic model of intentional activities, but many different types of intentional activities can be distinguished.

In order to be generic, our model will have to be able to model many different intentional activities. Indeed, several types of intentional activities are not only described, but can actually be *derived* from our model. Some examples are worked out below.

---

[8] A Popperian might argue that nothing was proved: the only thing learned is that the perspective worked on this occasion, which is history at the very moment of evaluation.

The many reasons that may cause an actor to act intentionally can all be summarised as *a wish for improvement*, according to the value system of this actor. Wigg [1994] gives a slightly more specific explanation [see also Minsky, The Society of Mind, p. 37]:

"We avoid pain and seek pleasure"

Consequently, two different types of intentional behaviour can be distinguished: "avoiding pain" behaviour and "seeking pleasure" behaviour.

A second step in typing intentional activities can be made by specifying where the pain or the pleasure manifests itself: in the "as is" situation or in the "to be" situation. In other words: is the intentional action required to *achieve* pleasure in the "to be" situation (proactive), to *maintain* the pleasure in the "as is" situation (preservative), to *prevent* pain in the "to be" situation (preventive), or to *reduce* the pain in the "as is" situation (curative). These types are presented graphically in figures 3a and 3b. These figures also make clear that "as is" and "to be" must be understood in relative terms, with respect to the action (intervention), rather than in absolute terms, with respect to now (present) and future. For example, although an increased greenhouse effect will (is likely to) manifest itself in the future, it is part of the "as is" situation, because *without intervention* this effect is expected to take place. The intervention "branches off" the normal flux of events; taking action and doing nothing are an "exclusive or". (Another way of looking at this is to interpret intentionally doing nothing as the script implementation stage of a perspective, of which "as is" models "now", the script models doing nothing and "to be" models the future as it will develop autonomously. It is possible indeed to intentionally do nothing.)

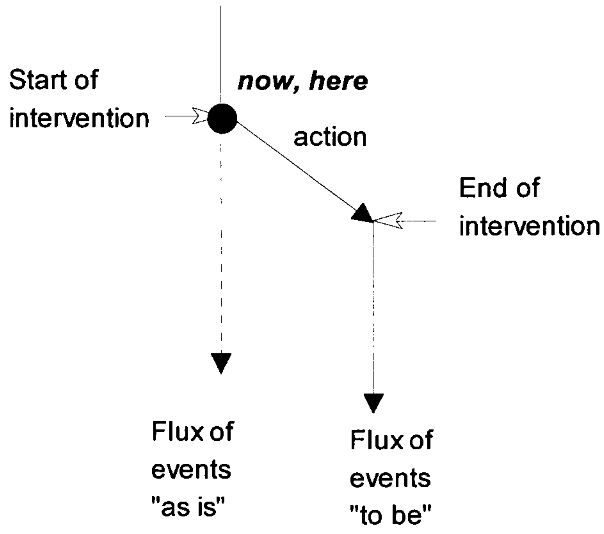


Figure 3a: "As is" and "to be" are relative to taking action.

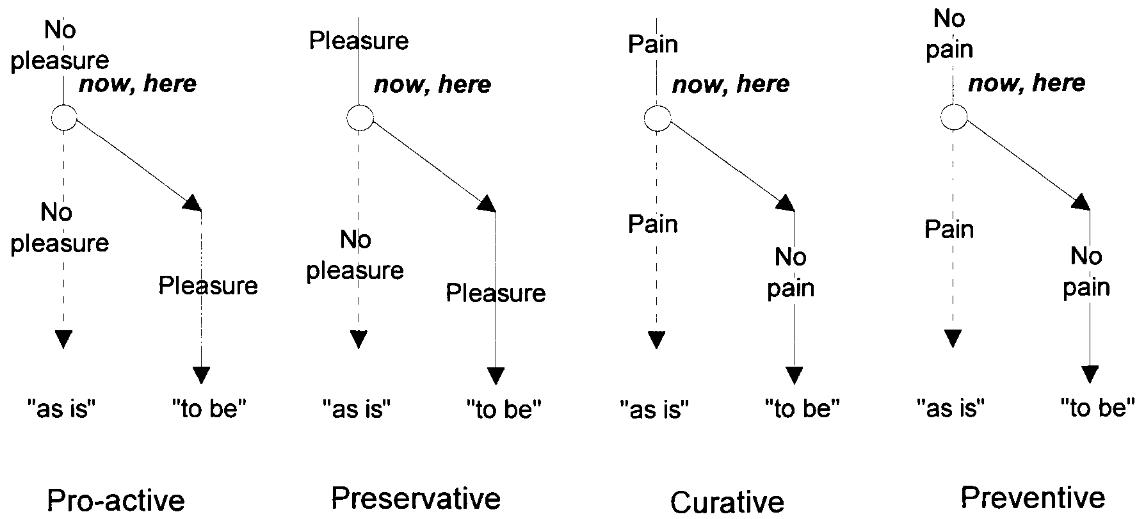


Figure 3b: A typology of intentional activities.

Another way of typing intentional activities is to characterise the perspective construction process (i.e. stage 2). A perspective construction process can take place fast, and require little or no cognitive strain at all; or it can take place slowly, and require a lot of cognitive effort.

In the first case, the intentional activity is a *routine* activity: the perspective is readily available (lying on the mental shelf, so to speak). In the second case, the intentional activity is *problematic*: the problem-solving process is difficult. Rather than its "avoiding pain" or "seeking pleasure" origin, it is an experienced lack of knowledge (a partially missing, incomplete perspective) that distinguishes a routine activity from a problematic activity. A disturbance of the correspondence between the actual environment and the intentions of an actor takes this actor out of his "routine" mode and puts him in a reflective and conscious mode.

In answering this critique we presented several examples of the way in which our generic model of intentional activities can be used to distinguish more specific subclasses of intentional activities. This substantiates our claim that the model is highly generic.

Critique 3: In many cases stages 2a, 2b and 2c (analyse "as is", synthesise script and predict "to be") do not take place in this sequence.

Our answer to this critique is: we agree. Our model of intentional activities, however, does not require that this sequence be obeyed. This will be elaborated upon below.

We do think that (in a very strict sense) it is *impossible* to give a meaningful characterisation of a situation "as is" without having a clue about what should be done (the script), and we do not think that it is possible to have a clue about what should be done without having a clue about the situation "to be", resulting from doing it.
In addition, in many cases dreaming about a pleasant future induces scripts, and these scripts in turn influence analysis. Some strategies in forecasting are even based on a reverse order. For example, a *back-casting* strategy in strategic explorations is based on describing potential futures first, and only after that investigating potential routes to achieve this [Quakernaat (1995)]. Also a remark of Mintzberg: "Analysis can never result in synthesis" [Mintzberg (1990) p. 29] implies a critique on this sequence (although we do not share this view; see below).

In our interpretation, models of "as is" and scripts and models of "to be" are coupled, and cannot exist without each other[9]. Stages 2a-c should not be interpreted as a sequence, but rather as one "perspective construction" (problem solving) stage, consisting of three threads. In stage 1, a rudimentary perspective emerges: an incomplete body of knowledge

[9] An ultimate consequence of this line of reasoning would be that, although a perspective can be thought of as consisting of three parts, these parts *in isolation* are not very ***use***ful. It is impossible to distinguish meaningful parts of a perspective that are not (part of or assemblies of) perspectives themselves. Perspectives consist of perspectives, at several systemic levels: the notion of perspectives is self-contained. Safranski, p. 126, discusses the way in which Heidegger explains the manifestation of a cathedra. In very much the same way the manifestation (emergence) of any object of concern can be explained as an awareness of an (assembly of) perspective(s).

about the situation "as is", a script for taking action, and a better situation "to be" expected to result from executing the script. On the basis of this incomplete perspective, the actor decides that it may be required to intervene in the flux of events that forms its environment. In other words: in order to be intentional (to have passed the acknowledgement stage) at least a very rudimentary version of *all three parts* of a perspective should exist.
During stage 2, the perspective is completed step by step. A change in any one of the three threads may have consequences for the other threads. This means, for example, that in some cases analysis may result in script synthesis, but in some other cases the reverse order is possible as well. The problem-solving stage (stage 2) *results* in a perspective of which the three parts are linearly ordered in time. The perspective construction *process*, however, can be a complex mixture of all three processes.

In order to shed further light on this, we will use an analogy. Perspective construction (hence problem solving) is like filling a set of three buckets of approximately the same size with water. The system of three buckets is standing on a shaky table. (The contents of) each of the buckets represents a part of the perspective to be constructed (the "as is" part, the script and the "to be" part, respectively). While filling the buckets, it is important to maintain an equilibrium to a certain degree: it is allowed to let one of the buckets take the lead, but be aware of the shaky table; water may be spilled. In some cases, water is short, and it may be required to reduce the water level of one of the buckets to re-establish equilibrium. The buckets may even become empty again: the perspective disappears altogether, and stage 3 will never be entered. Filling the buckets is a non-monotonous process.

At the end of problem solving, just before entering stage 3, the buckets should be filled, preferably but not necessarily as much as possible. When they are not yet filled, the resulting perspective is simply uncertain. Consider, for example, a scientific experiment. The situation "as is" as well as the script are described as precisely as possible. Preferably, laboratory conditions are used. The "to be" situation is only partially known though: this is what makes the experiment meaningful.
When the stakes are not too high or resources (to obtain additional "water") are lacking, uncertainty may be taken for granted, in one, two, or all three of the parts of the perspective, and stage 3 may be commenced. Thorough evaluation is advised.

However the problem-solving process may proceed: in order to be able to act intentionally, a perspective must be available. This perspective is not necessarily completely clear at the moment of acting. But it should be sufficiently clear to motivate and guide the taking of action. This does not imply that we think that all acting is conscious, intentional acting. On the contrary; we think that a large part is routine acting. The actor is hardly aware of the perspective at all (like in opening a door). It merely implies that *a claim of intentionality must be accompanied by the presence of a perspective.* It is typical for problem-solving activities though that perspective construction is a conscious and reflective process. A disturbance in the routine flux of events, the absence of an appropriate perspective, a mis-correspondence between an actor's intention and an actor's environment puts the actor in this reflective mode (for a Heideggerian account of this phenomenon, see [Safranski (1994) p. 133-134]).

Critique 4: The model assumes an axiomatic mind-body dualism, as it separates thinking in a mental domain (stage 2) from acting in a physical domain.

An axiomatic mind-body dualism is one of the main features of what is generally described as a rationalistic philosophical stance. This stance, with firm roots in the work of Galileo and Descartes, and originating even further back in the ideas of Plato and Aristotle, is based on a kind of mind-body dualism that accepts the existence of two separate domains of phenomena: an objective world of physical reality, and a subjective mental world of an individual's thoughts and feelings. For a short discussion of rationalism and alternative philosophical stances, see, for example, [Winograd and Flores (1986) p. 21] and [Sowa (1984), p. 356]. The assumptions on which the rationalistic tradition rests are summarised by [Winograd and Flores (1986) p. 30-31]:

1. We are inhabitants of a 'real world' made up of objects bearing properties. Our actions take place in that world.
2. There are 'objective facts' about that world that do not depend on the interpretation (or even the presence) of any person.
3. Perception is a process by which facts about the world are (sometimes inaccurately) registered in our thoughts and feelings.
4. Thoughts and intentions about actions can somehow cause physical (hence real-world) motion of our bodies.

The very notion of environmental situations of concern seems to imply a common sense mind-body dualism.

On the one hand, the expression "*our* environment" strongly suggests that we all share one single environment: our natural (physical) environment. Because of a number of causes, this natural environment is exposed to a lot of pressure. Our natural environment is highly complex and encompasses both human processes (production and consumption processes) and natural processes (for example, dispersion and deposition processes). Many believe that it *is*, however, *one* environment, and in principle the state of this environment can be known "objectively". Although we quarrel much about the exact identity of this shared environment, we do not dispute its existence. This belief is reflected, for example, by the fact that we monitor and measure the state of our environment from a rather "natural scientific" point of view.

On the other hand, the social context of environmental problems is widely recognised as well. A social context implies that several (and in general many) points of view must be considered in environmental problem solving. Manufacturers, consumers, governments and scientists, as individuals or as social groups, all formulate, adhere to, and act according to quite different perspectives. Multiplicity of points of view may result in the situation that different actors interpret "the same" situation in quite different ways. Typically, in an environmental situation of concern, different points of view are in conflict or at least incommensurable.

Most people will agree that environmental situations of concern and environmental problem-solving processes encompass both a physical dimension and a human dimension (see, for example, [de Groot (1992)], [Riviere (1991)], [the EU Research Programme "The Human Dimension of Environmental Change"]). According to this point of view, environmental problems result (typically as a side effect) from people, *intentionally* changing their *natural* environment. A methodology that aims at supporting environmental problem-solving processes should pay attention to both dimensions.

The duality explained above, however, introduces a number of difficult questions. For example, in what way do the notions of one "objectively" knowable natural environment and multiplicity of points of view relate to each other? And how should both dimensions be separated? These questions in more general terms are known as the "mind-body" or "mind-matter" problem, a problem that is in the middle of a fundamental philosophical dispute. Searle [1984] described the problem as follows:

> *"We think of ourselves as conscious, free, mindful, rational agents in a world that science tells us consists entirely of mindless, meaningless, physical particles. Now, how can we square these two concepts?"*

Our pragmatic solution to this problem is that we do not consider it to be a problem at all[10]. According to our point of view, it makes sense to distinguish a physical and an intentional dimension, as they are deeply rooted in common sense. It does not make sense, however, to postulate that only one specific interpretation of such a dual system is "true". We rather adhere to a remark that underlies the pragmatic philosophical school [Bain (1871)]:

> *"Belief is that upon which a man is prepared to act"*

and extend this to actors consisting of several individuals (including women). Our philosophical stance is *dualistic* in that we will assume that both a mental dimension and a physical dimension coexist. For example, we will talk about (objective) emissions as well as about (subjective) intentions of the owner of the factory causing these emissions. The two dimensions, however, exist *in the mind of an intentional actor*. And actors may

[10] Indeed, the mind-body "problem" does not prevent us from acting intentionally, so it is not a problem in terms of our definition.

disagree about what is relevant, what is objective and what is subjective. Next to *dualistic*, our stance is *relativistic* in that we recognise that *each* observer may formulate his/her own, private interpretation of such a dual system. The goal of problem solving is not to establish truth, but to construct a perspective of which the actors involved think (believe) that it models both their environment and their intention. This perspective will be the basis for concerted action.

The generic model of intentional activities at first sight seems to separate a *cognitive* perspective construction stage and a *behavioural* script implementation stage, which implies the dualism discussed above: thinking takes place in a mental domain and acting in a physical domain. This suggests that thinking stops at a certain point (notably at the end of stage 2), and physical action takes over. Some serious comments can be made to such a separation. For example, Mintzberg [(1990) p. 19] states that separating thinking and acting too rigidly seriously hampers learning "on the road", which not only is found to be disastrous in management science but has also been recognised as a serious problem in quite different fields of human endeavour. In addition, we would like to add that such a separation excludes the possibility to think about thoughts (which in our experience is a rather important thought process).

But our generic model of intentional activities does not separate a cognitive and a behavioural stage., The model rather separates perspective construction and acting according to the script part of this perspective. It distinguishes building a perspective ***about*** (modelling) an environment and interacting ***with*** this environment motivated and guided by this perspective.
Perspective construction results in a mental model, but is not necessarily a 100 % mental activity. It may include thinking, communicating and experimenting in complex combinations. Implementing a script changes an environment, but this environment is not necessarily physical. The environment of an actor can, for example, be mental, physical, or a complex combination (slightly running ahead: as is the case in a D-type context, see Chapter 3). For example, using a strategy in solving a riddle or debugging a perspective, are script-implementing actions, although they take place in a cognitive domain. Distinction of perspectives and the referents to which they refer implies a reflective stance, an object-meta relation. It is possible to think about physical things, it is possible to think about thoughts, but it is not possible to "physic" over thoughts or over things. This notion will be elaborated upon in the section about systemic reflection, and will reappear in the *Trinity* modelling language (Chapter 5 of this dissertation).

Critique 5: a stage-based approach is too rigid to account for intentional activities in a generic way.

Stage-based generic models of intentional activities are rather popular, as the large amount of examples cited earlier shows. However, at the same time they are rather rigid, which seriously limits their use in complex real-world activities. How do the claim of genericity and the rigid stage-based nature of the model agree?

The answer to this question is that the generic model can be used at different systemic levels. The concept "system" as we use it is more specific than its common sense interpretation (i.e. something that can be separated from its environment). A crucial aspect is the notion of *emerging properties*. An emerging property is a property of a larger system ("larger" implying some extensive quantity that can be attributed to both the system and the parts) that is not present in the parts of this system. To be even more precise: this emerging property is a function of the structure and interrelations of the parts of this system. In the words of Gosling [1962]:

"those properties that is, which are possessed by the system but not by its parts"

For example, a bicycle can be used as a means of conveyance, but its parts cannot.

We would like to add that the notion of emerging properties has an inverse counterpart, that is equally relevant but seldom if at all distinguished. To paraphrase Gosling: "those properties that is, which are possessed by the parts but not by the system". We propose the term *vanishing properties* for properties that are manifest at lower systemic levels, but disappear at higher systemic levels. An example is a tube of a tire of a lorry: in separation it can be used to play with in a swimming pool. As an integral part of a lorry this becomes problematic: this particular property vanishes.

Any change in systemic level (either upward or downward) results in both emerging and vanishing properties. For example, when changing systemic level downward, from the lorry to the parts (including the tube), the property that things can be transported vanishes, and the properties of a toy for the swimming pool emerge.

The importance of distinguishing both emerging *and* vanishing properties when changing systemic level is twofold.
First, it makes clear that changes in systemic level are not necessarily related with changes in complexity. Some new properties will emerge, but some will also vanish. Changes in systemic level are distinguished on the basis of a rather "objective" interpretation of referents, whereas the notion of complexity has to do with an intentional interpretation of referents. Whether one is interested in sub-atomic particles, understanding the universe, psychological phenomena or making a living with selling flowers: every systemic level (every discourse of intentional activities) has its own degree of complexity, that is not

influenced by the size, weight or other (physical or "objective") properties of the parts of concern.
Second, many concepts are relevant only at a specific systemic level. For example, a charm is relevant for someone investigating sub-atomic particles; a black hole is relevant for someone investigating the universe, the absence of a value system in a youth is interesting for a psychologist or a judge, and a flower is relevant for someone selling flowers. And a quantum-mechanical interpretation of selling flowers does not make sense from a flower selling point of view.
But the notions of intentional activities, problem solving, perspectives and actions are relevant at any systemic level. They are abstract concepts that do not vanish or emerge when changing systemic level. They are intimately related with being intentional, i.e. with being human. This is a deeper meaning of our claim that the model of intentional activities is generic: it is independent of systemic level, and meaningful at all systemic levels.

The generic model of intentional activities is systemic: it manifests itself at several systemic levels, and when changing systemic level it may be used again.
For example, when looking at a set of intentional activities, in many cases they can be interpreted as one overall intentional activity at a higher systemic level. Likewise, when looking at a complex intentional activity, this may be unravelled into a sequence or a set of concurrent smaller intentional activities. For example, the perspective of repairing a car can be interpreted as one overall perspective (repair car) or as a sequence of smaller perspectives (repair flat tire; repair engine; et cetera). Different mechanics may repair different parts concurrently: their perspectives are part of the overall "repair car" perspective. The notion of Knowledge Distribution Space (KDS, to be presented in Chapter 3) enables one to visualise the systemic, self-contained structure of a body of knowledge (perspectives). The same holds for intentional activities (in which perspectives play a crucial role): they are systemic and self-contained.
In some cases the problem is that a complex set of intentional activities *cannot* be interpreted as one overall intentional activity at a higher systemic level. Consider, for example, the complex whole of human intentional activities (production and consumption processes). Many intentional activities can be attributed to specific groups at specific systemic levels. For example, some actors intentionally produce cars, and others intentionally drive them around. However, at the level of the human race as a whole, the generic model is difficult to apply. A shared intention can perhaps be recognised (survival?), but concerted action is absent. A typical notion that emerges at this systemic level is "a sustainable society". A sustainable society should support the intentions of actors at several systemic levels: individuals, groups, nations as well as the human race as a whole. At this moment at higher systemic levels, shared concerted intentions are missing. This suggests that a systemic applicability of the generic model of intentional activities is an important guiding principle in attempting to attain sustainable societies.

2.4 PHILOSOPHY

What elements should be parts of a perspective? What systemic level is the "correct" one? And when is a perspective "ready"? The difficulty in answering these questions is in sharp contrast with the ease with which we solve small problems and act intentionally every day. Therefore, we suggest that it should be possible to give an interpretation of perspective construction that provides simple answers to these questions.
We developed a philosophical stance that provides such simple answers. This stance is at the basis of our generic model of intentional activities, our definition of problem solving, and the way we intend to support problem-solving processes by means of modelling processes. It is presented below.

2.4.1 Dualism, relativism and perspectives

Earlier in this chapter we emphasised that many people, at least for practical purposes, adhere to a mind/matter dualism of some sort when being engaged in environmental situations of concern. We explained that we will operate according to a flexible "mixture" of dualism and relativism: we recognise the practical value of distinguishing a mental domain and an environmental domain[11], (but) we do not adhere to the (rationalistic) idea that only one distinction is correct. In this section, we will present the underpinning of this stance.

Environments, perspectives and intentions

The three central concepts in our philosophical stance are *perspectives*, *environments* and *intentions*.

Perspectives were introduced earlier in this chapter: they are mental models, possessed by actors, consisting of three tightly interrelated parts: an "as is" part, a "script" part and a "to be" part. A more flexible description would be that they consist of a *before* part, a script part and an *after* part, as this does not introduce a situation in time. For example, when being confronted with an opened water tap, I may interpret this as an *after* part, and use the "open water tap" perspective to abduce that this tap was opened by somebody (modelled by the script) and before that it was closed (modelled by the *before* part, see also the example of parallel dynamic extensions of the *Trinity* modelling language in Chapter 5).

Environments are the media that an actor is aware of, and is able to interact with (in order to improve)[12]. They form the domain in which a change takes place.

---

[11] For example, interacting with a physical or a knowledge domain requires quite a different set of methods and instruments, a phenomenon well reflected by the differences between β and γ sciences.

[12] A simple dichotomy would be: environments can be physical (you can change your environment) or mental (you can change your mind). Within the *Trinity* modelling language (see the Methods part of this dissertation) three domains are distinguished: a knowledge domain, a

Intentions are wishes to improve environments. They are the driving force for intentional activities. They form the domain in which progress (improvement) is established.

A perspective can model either an actor's environment or an actor's intention. In this way, perspectives enable actors to be aware of their environments and intentions.

Perspectives model *environments* by means of a trivalent model relation (figure 4).
At the moment just between stages 2 and 3, the "as is" (the *before*) part of a perspective *descriptively* models the actor's current environment; the script part *prescriptively* models some actions that this actor may carry out; and the "to be" (the *after*) part of the perspective *predictively* models the "to be" situation in the environment, expected to result from actually executing the script. In terms of our philosophy, environments can only be understood in terms of these tripartite entities[13].

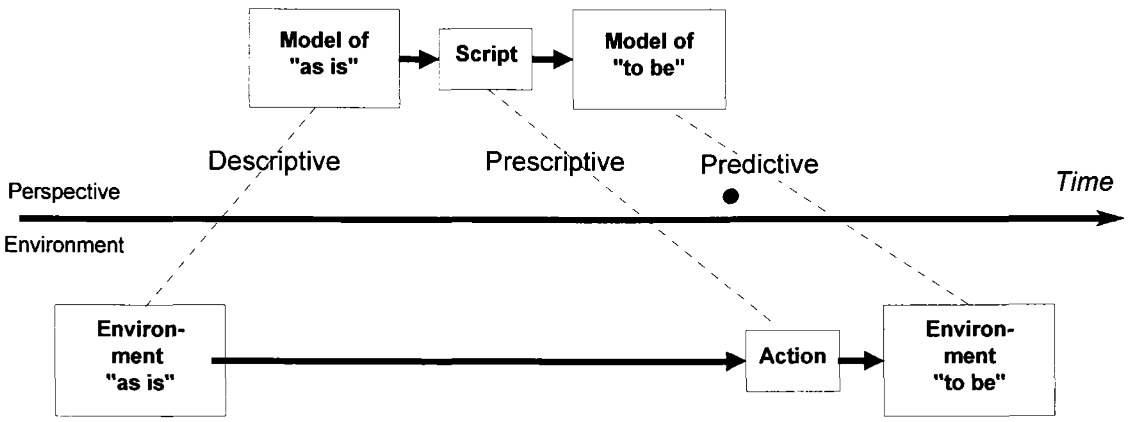


Figure 4: A perspective is linked to an *environment* by means of a trivalent model relation.

Perspectives also model *intentions* by means of a trivalent model relation (figure 5).

physical domain and a communication domain. Within each of these three domains both states and processes can be distinguished. The ***Trinity* principle** states that, in order to be able to intervene in a multi-actor environment of concern, it is sufficient to describe this environment in terms of a perspective using states and processes from these three domains.
[13] In rather Heideggerian terms: an object that is distinguished is distinguished as a space of human potential and concern. In terms of this dissertation, this potential and concern is known in terms of action potentials: pragmatically correct perspectives.

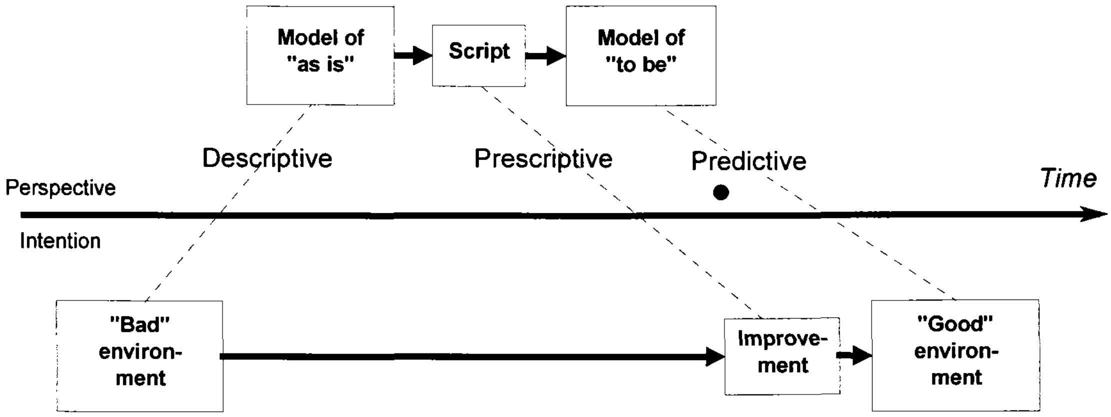


Figure 5: A perspective is linked to an *intention* by means of a trivalent model relation.

At the moment between stages 2 and 3, the "as is" part of a perspective models a situation that is less desirable. The script part models an action that realises (or initiates) an improvement. The "to be" part models the more desirable situation resulting from this action. In terms of our philosophy, intentions can only be understood in terms of these tripartite entities.

In short: perspectives model both intentions and environments, and as such enable an actor to know (to be aware of) these environments and intentions. This is interesting, as a specific perspective may form a "bridge" between the "objective"[14] notion of an existing environment and the "subjective"[15] notion of an intention to improve. In order to do so, a perspective must model *both* this environment *and* this intention. This implies a bi-directional model relation (figure 6). Perhaps a more intuitive way of understanding this is that at that very moment perspective, intention and environment become *one* (they unite). The mis-correspondence between environment and intention vanishes, the problem disappears.

Intentional acting is only possible if such a bi-directional model relation exists. At the moment just before intentional acting, the environment and the intention correspond: the perspective constitutes a bridge. The "as is" environment is less desirable; the actions improve; and the "to be" environment is better. A tripartite intention and a tripartite environment are coupled by means of a perspective. Perspectives that enable intentional action model both an environment and an intention.

[14] "Objective" in the sense that the existence of this environment is not questioned.

[15] "Subjective" in the sense that the intention only makes sense with respect to the environment.

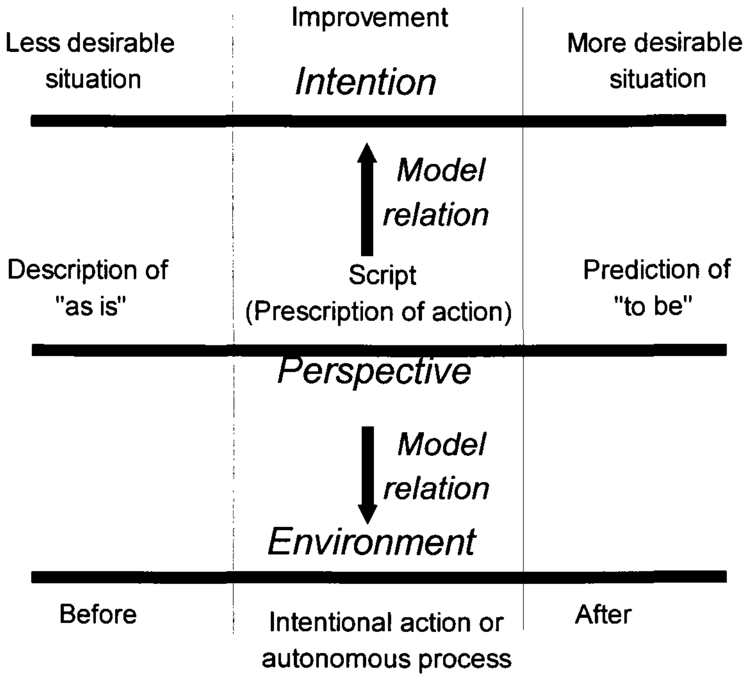


Figure 6: A perspective bi-directionally modelling an intention and an environment (for an autonomous process the script prescribes: *do nothing*).

An actor possesses the necessary capabilities to realise such a bi-directional coupling: he may interact with environments, pursue intentions, and possess and process perspectives[16]. At the moment that an actor is confident about such a bi-directional model relation, intentional acting can start. Only when such a bi-directional model relation is pragmatically correct, the intentional action will be successful.

In short, in order to be able to act intentionally:

1. an actor must *possess* knowledge (the actor possesses a perspective);
2. the actor must be *willing* to use this perspective (the perspective models an intention); and
3. the environment must *allow for* using this perspective (the perspective models an environment).

---

[16] We do not wish to blur the point to be made here, but, in principle, an intention can assume the role of an environment, and an environment can assume the role of an intention. For example, it is possible to intentionally improve an intention (which assumes the role of a mental environment), and to use the output of this process as a model of an intention to change another environment. See also systemic reflection.

Some examples will further explain the possible relations between perspectives, intentions and environments.

Possessing an uncoupled perspective

At the moment that you are reading this, you may not be in the neighbourhood of a water tap, and you may not have the intention to open a water tap. But you are likely to know the "open a water tap" perspective, that tells you that opening a closed water tap is likely to result in water pouring out.

A perspective modelling an environment

Consider the situation that you see a water tap in a greenhouse. You may model this situation by means of the "open the water tap" perspective[17]. The "as is" part of your model of the environment refers to the tap; the script part consists of the prescriptive model "open the tap"; the "to be" part pictures a tap with water pouring out. In effect you "deduced" that opening the tap will result in water pouring out of it, by means of the perspective already known to you. But you may not want to open the tap. The perspective models the environment, but the intention is missing. Nothing will happen.

A perspective modelling an intention

Consider that you see flowers in another greenhouse, that desperately need water. You may develop the intention to fetch some water: the "open water tap" perspective models this intention. But the tap may be absent. There is no environment that can be modelled with this perspective. The perspective models an intention, but the environment is missing. Nothing will happen.

A bi-directionally coupled perspective enables intentional acting

Consider the situation that you a) want to use a water tap and b) that you see a water tap. The perspective bi-directionally models an intention and an environment. This enables intentional acting: you enter stage 3, and open the water tap. Water is pouring out.

A bi-directionally coupled perspective, incorrectly coupled to the environment

Consider the same situation as before (the bi-directionally coupled perspective). You open the tap. Nothing happens. Apparently the model relation between perspective and environment was not pragmatically correct.

A bi-directionally coupled perspective, incorrectly coupled to an intention

Consider the same situation as before (the bi-directionally coupled perspective). You open the tap. Water is pouring out. However; you do not feel any better.

---

[17] As a matter of fact, becoming *aware* of the tap may be interpreted as the equivalent of a number of perspectives, in which a water tap plays a role, coming to the mind. The water tap "worlds" in the terms of Heidegger: it assembles a world in time and space (although Heidegger does not use the notion of a perspective). The set of perspectives becoming coupled, coming to the mind, make up what the water tap means to you. This in principle is relativistic, as each individual has a private set of perspectives. See also [Safranski (1994) p. 126-127].

Earlier in this chapter, we defined problem solving as perspective construction. At this moment, we are able to refine this definition:

> *Problem solving is the attempt of an actor (a problem owner) to re-establish correspondence between its environment and its intentions. This attempt manifests itself as a process of developing an incompletely coupled perspective into a perspective that models both the actor's environment and the actor's intention. The very moment that such a perspective is obtained, the correspondence between environment and intentions is re-established: the problem is solved; the actor acts intentionally.*

This new definition of problem solving is more sophisticated than the earlier definition (i.e. problem solving is perspective construction) in that it explicitly allows for the phenomenon that, during a problem-solving process, the intention and/or the environment of concern may change as well. An environment is "a space of potential for human concern and action" [Winograd and Flores (1986) p. 37][18]. And an intention is the driving force for intentional change. Environment and intention can be known by means of perspectives.

Taking sides?

The philosophy explained above is dualistic: it distinguishes both environments and intentions. This raises some questions about the precise relationships between perspectives, intentions and environments. Is it required to postulate a primacy of one above the other? And, as environments and intentions can only be known by means of perspectives, do they really (objectively) exist?

When we see a bottle of wine, we know (assume) that we can open and drink it, that we can smash someone's head with it, that it was bottled, that drinking a lot of it causes a head-ache, et cetera. This means that our knowledge of our environment is limited by the perspectives that we possess (at that moment). On the other hand, the perspectives that we couple are determined by the environment: not all the perspectives that we know apply, but rather the environment induces (or perhaps even is) a subset. When we see an orange, we cannot throw an apple.
In short: what we can know about an environment is limited by the perspectives that we possess, but the perspectives that we couple are influenced by what we call our actual environment.

---

[18] This interpretation of an environment resembles the stance of Winograd and Flores, derived from the work of Heidegger [Winograd and Flores (1986) p. 37]: "what really *is*, is not defined by an objective omniscient observer, nor is it defined by an individual, but rather by a space of potential for human concern and action. See also next page.

Obviously, our intentions are also limited by the perspectives that we possess. You must know that you can eat an orange in order to intend to eat one. On the other hand: not all the perspectives that I possess are candidate intentions. For example, I know that driving my Deux-Chevaux into the river Rhine will cause this superb car to disappear under water. A perspective indeed, but hardly a candidate to model an intention.
In short: what we can intend is limited by the perspectives that we possess, but the perspectives that we indeed use to model intentions are influenced by general values concerning "good", "bad" and "improvement".

We are in dire straits. Both environments and intentions can only be known by means of perspectives. However; from the discussion above it follows that at the same time environments and intentions induce the perspectives that are used to model them. How can environments and intentions induce perspectives, and how can perspectives enable one to know environments and intentions at the same time?

This situation, however, is not inconsistent, but paradoxical. People believe that they share certain perspectives. As a result, it is not only most inconvenient, but virtually impossible not to assume a shared referent as well. This is what the common-sense notion of physical reality amounts to: a shared set of perspectives and a shared action potential induces a strong belief that a shared environment, independent of the actors involved, exists as well. Nonetheless, physical reality is an emerging and vanishing property, depending on the actors (to be more precise: the perspectives) involved, rather than an axiom or a "natural" phenomenon. This interpretation of physical reality resembles the stance of Winograd and Flores, derived from the work of Heidegger [Winograd and Flores (1986) p. 37]:

> *"What really is, is not defined by an objective omniscient observer, nor is it defined by an individual, but rather by a space of potential for human concern and action."*

This is also what the common sense notions of good, bad and improvement amount to: a shared set of perspectives induces a strong belief that sensible intentions (a value system), independent of the actors involved, exist as well. In principle, however, these notions are emerging and vanishing properties, that depend on the actors (or, more to be more precise, the perspectives) involved. History provides us with many examples of this. Perspectives that are shared at different systemic levels account, for example, for the notions of cultural, disciplinary and personal value systems (see also the typology of problems to be presented in Chapter 3 of this dissertation).
Environments and intentions do not exist because of some "external" cause. They are heavily influenced by the perspectives that an actor possesses (which accounts for "relativistic" in "dualistic but relativistic"). Environments and intentions are known by

means of perspectives. But the perspectives that one possesses determine the environments and intentions that one can distinguish as well[19].

In summary; both environments and intentions are meaningful notions. In physics it may be useful to emphasise that one is modelling an independent objective reality. And in the humanities it may be useful to emphasise that one is interested in understanding the subjective points of view of different individuals or groups. In problem solving, however, it is important to construct a perspective that, according to the actors involved, models both their intention and their environment.

Perspectives that are shared and used intensively are believed to be *true*, which implies that they do not need further justification. This comes very close to the pragmatic definition of true knowledge, as a compliment to a set of beliefs that do not need further justification (see, for example, [Rorty (1990) p. 24]). Shared perspectives are true in the sense that everybody who shares them thinks that their *environment* can exist and their *intention* makes sense. It is our knowledge about shared perspectives[20], that is the basis for the notions of a "real" world and "universal" intentions. And knowledge about a shared environment and shared intentions enable communication and co-operation. They offer a very useful platform to start multi-actor problem-solving processes. In this sense, bi-directionally coupled perspectives that are *not* shared constitute the difficult part of problem solving.

2.4.2 Simple answers to difficult questions

What elements should be parts of a perspective? What systemic level is the "correct" one? And, when engaged in a problem-solving process: when is a perspective "ready"? The stance elaborated upon above provides remarkably simple answers to these questions, that were posed at the beginning of this section.

For problem-solving purposes, perspectives must be constructed of elements that model parts of *both* the intention *and* the environment of an actor. (Note that this implies that both the intention and the environment of concern may change during the problem-solving process.)

The "correct" systemic level of a perspective is that level that enables the problem solver to realise such a bi-directional model relation: the perspective should be sufficiently detailed to motivate and guide intentional action.

---

[19] In other words: an environment is referred to by a subset of the perspectives that one possesses. An intention prunes this subset further. A value system is referred to by a subset of the perspectives that one possesses. An environment prunes this subset further.

[20] For a rather intriguing theory about actors at sub-individual levels see Minsky's book "The society of mind". The notion of sub-individual actors, used in this book, enables one to extend the notion of shared perspectives, intentions and environments even to sub-individual actor levels.

Problem solving stops and intentional action is possible at the very moment that the actor judges a perspective to model *both* his environment *and* his intention, and this to such a degree that the difference between "as is" and "to be" motivates taking action, and that the script guides taking action. At this point, intention and environment are believed to correspond.

2.5 SYSTEMIC REFLECTION

People possess the ability to reflect. This enables them to be aware of what they do. For example, actors may reflect about an environment. But they may also reflect about an intention to change this environment. And so on.
Reflection can be modelled by means of our generic model of intentional activities. This is so because the systemic nature of this model allows to chain intentional activities. Two different ways of chaining intentional activities can be distinguished: 1) chaining by means of a shared environment and 2) reflective chaining.

The first way of chaining intentional activities is by means of a shared environment. For example, actor A intentionally manufactures a bicycle (guided and motivated by a "construct bicycle" perspective); actor B intentionally buys this bicycle in order to visit his mother-in-law (guided and motivated by a "visit mother-in-law" perspective). The intentional activities are chained by means of the bicycle, which forms a shared element in the perspectives of both actors with respect to their environment.

In the second way of chaining intentional activities, a shift in reflective level is at stake. Two types of shifts can be distinguished: a shift upward and a shift downward.
In a shift upward, the *perspective* of intentional activity 1 becomes the (mental) *"as is" environment* of intentional activity 2. The environment of intentional activity 2 consists of perspectives and the processes that may change them.
In a shift downward, the (mental) *"to be" environment* of intentional activity 2 becomes the *perspective* of intentional activity 3.
We call this kind of chaining intentional activities, in which perspectives assume the role of mental environments (a meta-object shift) and mental environments assume the role of perspectives (an object-meta shift) *systemic reflection.* It resembles the notion of reflective architectures, as was worked out by Reinders in the field of knowledge based systems [Reinders (1991)].

For example, recall the greenhouse-with-water-tap example. Imagine carbon-dioxide coming out of the tap instead of water (carbon-dioxide is used to boost the production of crops). Apparently, the coupling between perspective and environment was not pragmatically correct. This is an interesting situation. You may recall that the tap looked slightly different from a normal water tap. You may specialise (refine) the "open a water tap" perspective (and in addition generate a new "open a carbon-dioxide tap" perspective).

You may forget the original "open a water tap" perspective (as it was wrong). Note that you are engaged in an intentional activity in an environment that consists of *perspectives*, rather than the original greenhouse. You shifted upward. Whatever you will do in this reflective mode, it will not physically change the original greenhouse environment (although you may interpret it differently).
The output of this reflective intentional activity 2 (which is a perspective) may become the *intention* of a subsequent intentional activity 3 (i.e. a shift downward). For example, the new "open a water tap" intention may be *used* intentionally: a water tap is looked for, and it is opened, resulting in water pouring out.

Upper and lower levels of systemic reflection

The concept of systemic reflection in principle makes it possible to systemically shift upwards or downwards in a never ending sequence. This would result in a "loop" of upward or downward shifts. A process that is hardly desirable. What are the guiding principles that prevent this from happening? We will distinguish two guiding principles: one that prevents a loop upward, and one that prevents a loop downward.

The guiding principle that controls a loop upward is that a *reason* must exist to reflect. This reason in general terms is a manifestation of a disturbance in the correspondence between environment and intention: intentional acting is not possible. For example, I want to cross a river, but I am not able to do so (this starts a line of thoughts). Or the evaluation stage of the former intentional activity showed a discrepancy between expectations and results (this continues a line of thoughts). These events may cause a new reflective level (a shift upward).
Now imagine that also at the reflective level a discrepancy exists between expectations and result. Another shift upward will be the result. And another. And another. At a certain moment, the situation will become very complex. The actor is likely to start to doubt his/her approach. The effort put in realising the intention is not worth the trouble. Perhaps it is not even possible. Perhaps it would be wiser to just abandon it, and do something else. In practice, this phenomenon will limit the number of shifts upward[21]. (But it also limits the potential to intentionally realise complex paradigm shifts, like establishing a sustainable development.)
The guiding principle that prevents a loop downward is that at a certain moment, perspectives are modelling an environment that does not reflect to another referent anymore. The referent is considered to be "the physical world" rather than an intention. For example, the "open water tap" perspective may model a water tap in a greenhouse. It

[21] A well known rule of thumb, known as "the seven why's", says that seven is about the maximum number of levels of explanation. This might suggest that seven reflective shifts is a limit as well. Experiences in *Trinity* modelling so far suggest that far fewer levels are common practice: it appears to be difficult to operate on different reflective levels. For explicit modelling purposes, in many cases one reflective level suffices, and two (which would, for example, take into account strategic processes on top of primary processes) is the maximum. See for example the "throw away battery" example of Chapter 5.

does not make sense to imagine this tap reflecting about its environment, let alone to intentionally change its environment. A water tap not liking to be in a greenhouse, and therefore changing position, is a rather nonsensical idea for most of us. This is another way of looking at "physical reality": this is an environment to which we do not attribute the potential to formulate and pursue intentions.

2.6 MODEL-BASED SUPPORT FOR PROBLEM SOLVING

The goal of this chapter was to provide a foundation for the notion "Model-based support for problem solving". All the building elements we need to do so have been elaborated upon in depth in the above sections of this chapter:

1. the generic model of intentional activities;
2. the definition of problem solving as perspective construction;
3. the philosophical interpretation of problem solving as a perspective construction process that stops once the perspective models *both* the environment *and* the intention of the problem solver.

Especially the third point reveals a direct relationship between modelling and problem solving. The result of a problem-solving process is a perspective that models both an environment and an intention. Therefore, perspective construction must be a modelling process itself.

However, in this interpretation all the mental activities that constitute a problem-solving process would be modelling steps (problem solving would be modelling). Although this interpretation has a certain appeal (it provides us with a metaphor to theorise about and experiment with thinking processes), it does not help us in actually providing model-based support *for* problem solving. Therefore, we will use a different relation between modelling and problem solving than the *equality* relation (i.e. *perspective construction* ***is*** *a modelling process*). We rather propose that the relation between modelling and problem solving is a *supportive* relation (i.e. *the modelling process* ***supports*** *perspective construction*). Model-based support for problem solving should enable us to model the development of perspectives, and should result in a model of a perspective. The model has both a heuristic and a descriptive function. The result of a problem-solving process still is a perspective that models both an environment and an intention. The modelling process, however, enables us rather to reflect on the perspective than result in one. A double model relation is at stake (see also [Diepenmaat (1993b)]. The relation between models of perspectives, perspectives, intentions and environments is presented in figure 7.

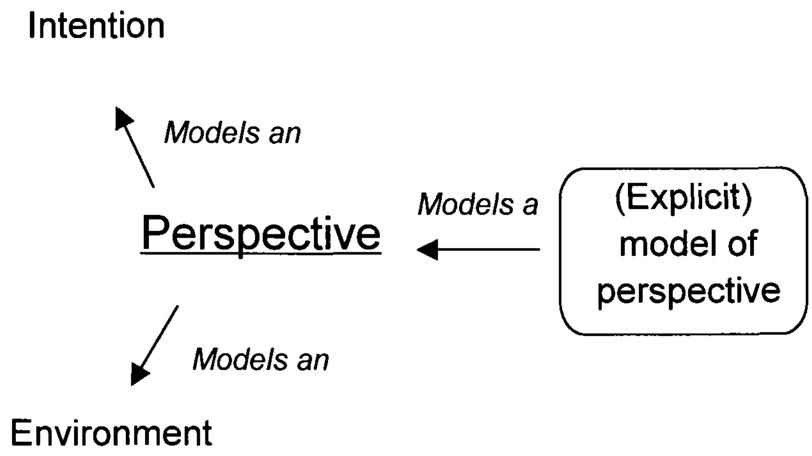


Figure 7: A perspective may model an intention and/or an environment, and may be modelled in turn by an (explicit) model.

General functions of models

Why bother? What is the benefit of introducing models *of* perspectives and model-based support *for* perspective construction?
The answer to this essential (and, as far as this dissertation is concerned, existential) question is that using models and modelling introduces a number of benefits when dealing with a referent. Below, these general benefits of models and modelling will be presented in terms of functions (see also [Diepenmaat (1993a)]).

- A **reflective** function (models support thinking *about* a referent, as opposed to being engaged *in* or interacting (experimenting) *with* a referent).
- A **heuristic** function (the model suggests additional qualities of the referent).
- An **explorative** function (for example, exploring the behaviour of its referent, "what if I would do this" tests).
- An **abstraction** function (reduction of complexity by means of emphasising certain features and neglecting others).
- A **formalisation** function (for example, using a clear modelling language instead of natural language).
- A **generalisation** function (reusability of models).
- An **explanation** function (opening the "input -> output" black box).
- An **efficiency** function (making experiments manageable, for example by means of a scale model or a simulation programme).
- A **communication** function (explicitly represented models facilitate communication, which facilitates knowledge integration).

- A **knowledge management** function (unknown knowledge cannot be managed (taught, thought, exchanged, integrated, changed and forgotten). Explicit models provide a basis for knowledge management. See, for example, [Laske (1990)], [Wigg (1993)]).

In model-based support for multi-actor problem solving the referent to be modelled is a multi-actor perspective. The result will be a model of several actors, acting and communicating in a shared environment, according to complex points of view. The problem-solving process is likely to be complex as well: several actors, i.e. informers and consultants, are likely to contribute to the problem-solving process. The model typically describes a process of change, in which several actors will change their perspectives and hence their actions in the shared environment. Typically, several alternative perspectives will be constructed and examined. The general functions presented above fit close to this description: it may be expected that providing model-based support is beneficial for multi-actor problem-solving processes.

2.7 DISCUSSION AND CONCLUSIONS

In this chapter we presented a philosophical foundation for the notion of "model-based support for problem solving". We developed a theory of problem solving within the framework of intentional activities. Central concepts in this theory are intentional activities, perspectives, environments and intentions. In terms of our theory, a problem emerges as a discrepancy, a mis-correspondence between an actor's intentions and an actor's environment. A lack of knowledge (a missing perspective) prevents this actor from acting intentionally. In line with this, problem solving is defined as follows:

> *Problem solving is the attempt of an actor (a problem owner, possibly a group of individuals) to re-establish correspondence between its environment and its intentions. This attempt manifests itself as a process of developing an incomplete perspective into a perspective that models both the actor's environment* ***and*** *the actor's intention. The very moment that such a perspective is obtained, the correspondence between environment and intentions is re-established: the problem is solved; the actor can act intentionally.*

Our philosophy is developed with the idea to provide model-based support in mind: it should therefore not come as a surprise that the notion of modelling can be integrated easily. The process of perspective construction is supported by a modelling process.

Perspectives, although tightly connected with some action potential, are a rather static notion of knowledge. When modelling them, the modelling process merely shows a "time series" of intermediate results. The transition processes (including the knowledge that drove them) are not shown (although they in turn can be modelled as intentional actions). We do not feel that this is a serious restriction, especially in the light of our rather pragmatic overall goal: providing model-based support for multi-actor problem-solving processes. Perhaps the analogy with a movie will make clear why. In the same way that a film is a sequence of frames, with each frame referring to a snapshot in time, the problem-solving process can be described as a "trace" of intermediate perspectives, with each perspective referring to a snapshot in time. The sequence slowly (although not necessarily monotonously) converges towards a bi-directionally coupled perspective. Although a film as a medium does not include notions that explicitly refer to the change from one frame to another (it is merely a sequence of frames, and does not include transitions), presenting frames as a time series results in an emerging notion of change. Dynamics emerge, rather than are present (explicitly represented). The same we hold true for developing perspectives: a sequence of intermediate perspectives can be understood as a process. In much the same way that we understand a sequence of frames as a movie, the modeller (or another interpreter) understands a sequence of perspectives as a modelling process. From a strictly pragmatic point of view, only the resulting perspective, the last one in the time series, the one just before acting according to the script part of this perspective, matters. For example, when I am opening a bottle of wine, just before that I am not reconstructing the way in which I originally obtained the corresponding perspective.

It is the spectator of a movie that attributes meaning to the sequence of frames: this meaning, strictly spoken, is not in the very frames themselves. The same holds for a sequence of perspectives, describing a problem-solving process: the "spectator" must understand the progress of (or, more neutral, the meaning behind) two or more models of perspectives in sequence. This requires a conceptual vocabulary that enables one to understand and think about perspectives *in sequence*, on top of a means to represent perspectives *in isolation*.

As a consequence, in addition to the elements elaborated upon in this chapter, model-based support for multi-actor problem solving requires **a)** a clear notion of multi-actor situations and multi-actor processes, and **b)** a modelling language that both enables us to represent and adapt perspectives concerning multi-actor situations. Developing exactly these notions will be our research agenda for the two theoretical chapters to follow.

PART III: THEORY

CHAPTER 3

KNOWLEDGE DISTRIBUTIONS AND KNOWLEDGE PROCESSES (CONSEQUENCES FOR PROBLEM SOLVING)

3.1 INTRODUCTION

In environmental problem solving, as a special case of multi-actor problem solving, typically a large number of actors (participants) are involved, each of them acting according to a different point of view. As a result of this, large amounts of highly diverse knowledge are involved, knowledge that somehow must be brought into coherence. This is an important, and often even dominating, factor in environmental problem-solving processes.
This observation leads to a number of questions with respect to the relation between knowledge distribution (the way in which knowledge of relevance is distributed over persons) and problem-solving processes. Answering these questions is of both methodological and practical relevance, not in the least because this will provide a deeper insight into the notions of multi-actor situations and multi-actor problem-solving processes. In this chapter, two of these questions will be addressed:

> *"Is it possible to classify situations of concern based on the way in which knowledge of relevance is distributed over individuals?"*

> *"If so, does this distinction provide (methodological) guidelines for the subsequent problem-solving process?"*

In section 3.2 we will develop a conceptual framework for investigating the relation between knowledge distribution and problem solving. Three attributes will be presented that allow for characterising a body of knowledge in terms of its distribution over actors. The combination of these three attributes results in the concept of *Knowledge Distribution Space*: a means to investigate the relation between knowledge distribution and problem-

solving processes in general, and to specifically answer the two central questions of this chapter.
In section 3.3 the first research question is addressed. A qualitative interpretation of Knowledge Distribution Space results in a typology of environmental problems, based on knowledge distribution criteria.
In section 3.4 the second research question is addressed. The problem types of section 3.3 are investigated from a methodological point of view. We will show that problem types can be related to methodological problem-solving principles and types of problem solvers.

3.2 KNOWLEDGE DISTRIBUTION SPACE (KDS)

In this section, a conceptual framework is presented that enables the investigation of the relation between knowledge distributions and problem-solving processes. This framework in principle is not restricted to *environmental* situations of concern. However, environmental problems are typical and manifest examples of a problem type, in which large amounts of widely different knowledge must be integrated. They are *examples par excellence* for multi-actor problem solving. For this reason, the framework is of specific interest to environmental problem solving.
The framework is rather abstract and theoretical. At first sight this may seem to be a strange way to investigate a phenomenon like knowledge distribution. However, the framework provides a basis for addressing many questions related to knowledge distributions and knowledge processes. We will show that it permits us to give very practical answers to the two central research questions of this chapter.

First, the notion of a *perspective* will be re-introduced: a static notion of knowledge that enables an actor to act intentionally. Second, three attributes, based on perspectives, are presented. These attributes enable one to characterise a body of knowledge from a knowledge distribution point of view. Representation of these three attributes in a three-dimensional space results in the concept of *Knowledge Distribution Space*: a means to visualise knowledge distributions and the changes within knowledge distributions (i.e. knowledge processes). Third, a few intriguing features of KDS are described in more detail. These features form the "backbone" of the research presented in this dissertation.

Perspectives revisited

Knowledge is a rather open notion, to which many different interpretations can be given (see, for example, [Aamodt (1990)]). As a result, it is difficult to make operational the notion of a knowledge distribution (which is the central concept of this chapter). Therefore, as a more specific alternative for the concept knowledge, we will use the notion of a *perspective*.

Perspectives have already been introduced in Chapter 2. A *perspective* is a body of knowledge that an actor can apply to his/her environment and that guides and motivates

the actions of this actor toward his/her environment. The meanings of the concepts of "perspectives" and "intentional actions" are circular: an intentional action is an action guided and motivated by a perspective; a perspective is the body of knowledge that makes an action intentional. In Chapter 2 of this dissertation, both concepts were interpreted as parts of the umbrella concept "intentional activity".
Perspectives are distinguished on the basis of *pragmatic* considerations: it is a notion of knowledge that is closely connected to a human agent who intends to perform an action which is expected to result in an improvement. The notion of perspectives enables one to delimit specific bodies of knowledge, that serve specific intentions. This, in turn, enables one to define knowledge distributions in a clear way.

For example, during a bicycle tour I may get a flat tire. I may fix this tire. The body of knowledge that enables me to do so is a perspective, as I may apply it to this specific situation. This in turn enables me to actually fix the tire and continue my tour. The knowledge that performing the action will result in a fixed tire *motivates* my actions (the why), and the knowledge about the procedure to do so *guides* my action (the how).

The process of fixing the tire can be thought of as a conglomerate of smaller actions, for example, "get repairing tools", "get bucket filled with water", et cetera. Some of them may take place in a sequential order, and some of them may take place concurrently. Each of these actions is intentional, and, therefore, is guided and motivated by a perspective in turn. These smaller actions also can be thought of as conglomerates. For example, "get bucket filled with water" can be decomposed into "find bucket", "walk towards water-tap", "put bucket beneath tap", "open tap", et cetera. Each of these recursive actions is guided and motivated by a perspective in turn. This sheds light on the recursive structure of a perspective (and intentional activities for that matter): complex perspectives may be thought of as consisting of smaller perspectives, until a level is reached at which the actions associated with this perspective are considered to be *atomic*. The notion of perspectives is *self-contained.*

The level at which perspectives are considered to be atomic depends on pragmatic considerations, which may differ considerably. For example, a chief executive officer may have knowledge about the same referents as the specialists he/she is managing, but his/her knowledge is typically more global[22].

Knowledge distribution attributes

Within environmental problem situations it is common that the body of knowledge of concern is distributed over several persons. For example, several societal actors take part

[22] The level at which perspectives are considered to be atomic is a pragmatic choice. It can perhaps be compared with a Fourier analysis, in which a signal in a time domain is converted into a signal in a frequency domain. At a certain point, adding still higher frequencies does not contribute to improving the signal anymore (is of no further pragmatic use), and therefore the frequency range is cut off.

in the problem context; several specialists contribute to a solution. Obviously, knowledge distribution affects both the potential and the way to operate in such a situation. In general, however, the precise effect of knowledge distribution is rather obscure, due to the lack of clear definitions of knowledge distributions.
For this reason, this section will present a more precise characterisation of knowledge distributions. Three attributes will be presented that enable one to characterise the distribution of a body of relevant knowledge. The three attributes are based upon a common denominator: perspectives. They are distinguished because distinction of the three of them is both necessary *and* sufficient to be able to characterise knowledge distributions as well as knowledge processes. They are:

- the complexity of a point of view[23] (*complexity*)
- the number of individuals adhering to a point of view (*adherence*)
- the number of different points of view (*diversity*)

The *complexity* (**c**) of a point of view specifies the number of different perspectives that are distinguished in this point of view (see also the flat tire example above). For example, an expert's point of view typically is high **c**, as many different partial perspectives can be distinguished. The knowledge of a novice, on the other hand, is likely to be low **c**.

The *adherence* (**a**) to a point of view characterises the number of individuals sharing this (the same[24]) point of view. In combination, these individuals constitute a homogeneous group. For example, a disciplinary or cultural point of view is high **a** (as many individuals share it), expertise is low **a**.

In many cases, more than one point of view must be distinguished. The reason for distinguishing several combined points of view (as *one* body of knowledge of relevance) typically is a feeling of discontent: the combined actions of the actors adhering to these points of view result in a pattern of behaviour that is not desirable or can be improved. A typical example is an environmental problem context. The situation is multi-actor. This phenomenon is taken into account in the third attribute, diversity.

The *diversity* (**d**) of a body of knowledge specifies how many different[25] points of view are distinguished.

---

[23] Point of view is a synonym for perspective. In this section, we use the term point of view when referring to a perspective that has attributed a complexity, an adherence and a number of other points of view with which it forms a body of knowledge as a whole.
[24] Two perspectives are "the same" if it is not required to distinguish them from a pragmatic point of view (see also diversity).
[25] Two perspectives are "different" if they cannot be treated as the same for pragmatic reasons (see also adherence).

Each of the **d** different points of view, in principle, has its own complexity and adherence. However, we can express the distribution characteristics of any body of knowledge by means of only three attributes. Therefore, in case **d** > 1:

> the *complexity* of the body of knowledge as a whole is the *mean* complexity *per individual*; and
>
> the ***adherence*** to the body of knowledge as a whole is the *mean* adherence *per point of view.*

In case **d** = 1, the body of knowledge is a single point of view, and the "**d** > 1" definitions of **c** and **a** for bodies of knowledge reduce to the "**d** = 1" definitions of **c** and **a** for points of view (the mean of one single number equals this very number).

In mathematical terms, the three attributes are described by the following equations:

$$\mathbf{c}_{\text{body of knowledge}} = \Sigma\, \mathbf{c.a.d} \;/\; \Sigma\, \mathbf{a.d} \qquad \textbf{(1a)}$$

$$\mathbf{a}_{\text{body of knowledge}} = \Sigma\, \mathbf{a.d} \;/\; \Sigma\, \mathbf{d} \qquad \textbf{(1b)}$$

$$\mathbf{d}_{\text{body of knowledge}} = \Sigma\, \mathbf{d} \qquad \textbf{(1c)}$$

When confronted with a societal problem situation, the body of knowledge of relevance can now be characterised by a triplet **c**, **a** and **d**. Such a triplet specifies *distribution characteristics* of the body of knowledge of concern. For example, when **d** is large, many different points of view must be considered in combination. When **c** is large, the points of view are complex. When **a** is large, the points of view are shared by many individuals. In addition, combinatorial effects may manifest themselves. For example, if *both* **c** *and* **d** are large, many complex points of view must be considered. Each of these characteristics introduces specific requirements for attempts to improve this societal problem situation (this is the key that makes it possible to answer the research questions of this chapter).

An example

A numerical example to explain equations 1a-c is presented below.

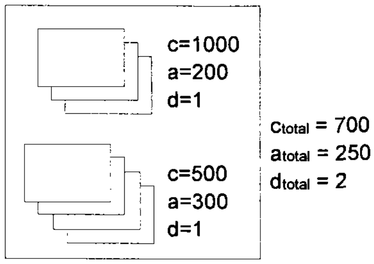


Figure 1: Two points of view, one body of knowledge.

The total number of perspectives in the two points of view of figure 1 equals

Σ **c.a.d** = 1000.200.1 + 500.300.1 = 350000

$\mathbf{c}_{total}$ = mean complexity per individual =

Σ **c.a.d** / Σ **a.d** = (1000.200.1 + 500.300.1) / (200.1 + 300.1) = 700

$\mathbf{a}_{total}$ = mean adherence per point of view =

Σ **a.d** / Σ **d** = (200.1 + 300.1) / (1 + 1) = 250

$\mathbf{d}_{total}$ = number of different perspectives =

Σ **d** = 1 + 1 = 2

Note that Σ **c.a.d** equals $\mathbf{c}_{total}.\mathbf{a}_{total}.\mathbf{d}_{total}$.

Knowledge Distribution Space

By means of assigning the three knowledge distribution attributes to the axes of a three-dimensional space, it becomes possible to represent bodies of knowledge as points in this space (figure 2). A point in this space specifies distribution characteristics of this body of knowledge. Consequently, we will call the space *Knowledge Distribution Space* (KDS). A point in KDS, which we will refer to as a KD (for Knowledge Distribution), refers to a societal situation of concern from a knowledge distribution point of view. Atomic perspectives are perspectives of which no further decomposition is known. An atomic perspective is represented in KDS as **c** = 1, **a** = 1, **d** = 1.

The three attributes **c**, **a** and **d** are numerical attributes. Above (in the box) we presented a numerical example of equations 1a-c in order to explain these equations. This, however, suggests a level of precision that is not intended: for practical purposes only qualitative interpretations are used. It is possible to estimate and reach agreement about the approximate position of a societal situation of concern in KDS. For example, it is possible to qualify the complexity of a point of view as “high", "medium" or "low"; to qualify the number of points of view as large or small; and to qualify the adherence (the size of groups) as small, medium or large. "Measuring" (counting) this, however, will be cumbersome, and different persons are likely to disagree about the exact numerical value. When different persons start arguing about approximate positions, this implies that they interpret the societal situation of concern quite differently.

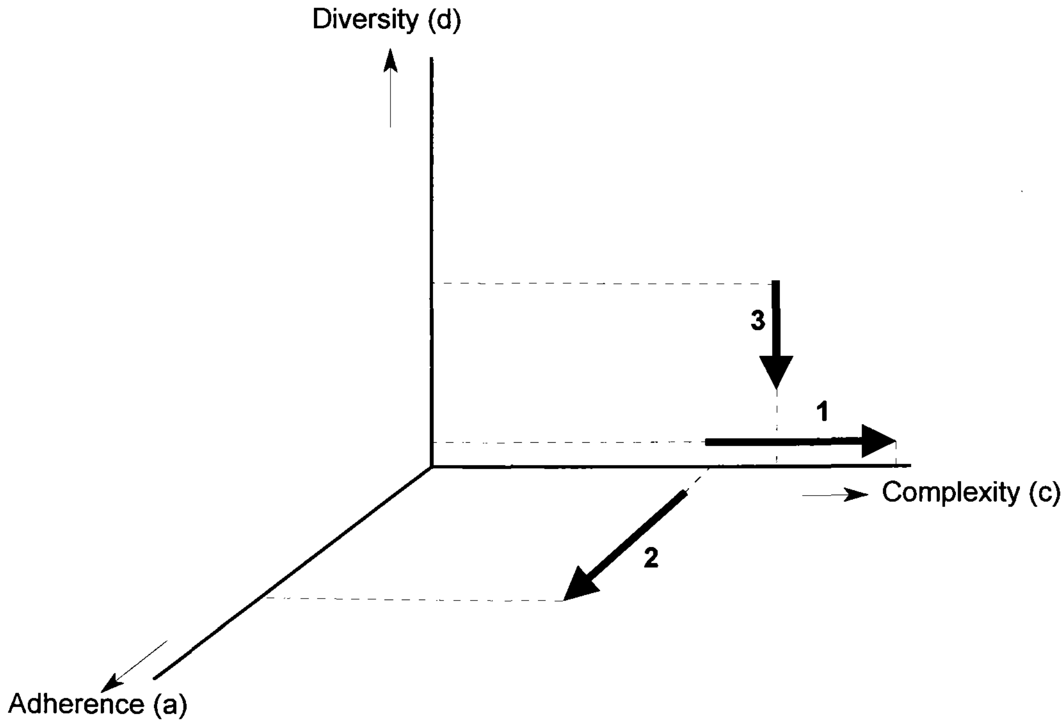


Figure 2: Knowledge Distribution Space.

3.3 THEORETICAL FEATURES OF KDS

A very practical answer to the question "why use the KDS framework?" is that a qualitative interpretation of KDS enables us to answer the two central research questions of this chapter. In subsequent sections we will show this by deriving a typology of societal problem situations from KDS, and relating problem types to methodological problem-solving issues.

The reason, however, that this is possible at all is rather fundamental. The KDS representation has many remarkable features that are implied by the definitions of the three attributes (they are not distinguished by chance). These features were used intensively as a background for developing the theory of qualitative modelling processes, to be presented in Chapter 4, and the *Trinity* modelling language, to be presented in Chapter 5 of this dissertation. For this reason, at this place we want to present these intriguing features. KDS, however, is a rather abstract framework, and at this point it may be difficult to understand the relevance of these features for multi-actor problem solving, or the impact of these features on this research. In Chapter 11, General discussion, we will re-visit the relations between KDS, the theory of qualitative modelling and the *Trinity* modelling language. In this section, seven features of KDS will be highlighted.

Feature 1: bodies of knowledge (parts of society) are represented by points in KDS

A KD represents a distributed *knowledge state*. As such, a snapshot of a part of society can be represented by means of a point in KDS.

It is a structural characterisation rather than a semantic one: the contents (meaning, action potential) of this knowledge is abstracted. The knowledge distribution characteristics are highlighted. This implies that several semantically different bodies of knowledge may map at the same point in KDS (or, slightly running ahead, at the same iso-plane[26]), as their knowledge distribution characteristics may be similar.

---

[26] In features 6 and 7 it will be explained that re-interpreting the knowledge distribution characteristics of a body of knowledge corresponds with moving **on** an iso-plane, and changing a body of knowledge corresponds with jumping towards **another** iso-plane. This implies (although this may not be straightforward at this moment) that it is impossible to change the referent of a body of knowledge by means of a trajectory that follows this iso-plane. A route to accomplish this would have to go via other iso-planes.

Feature 2: knowledge processes (societal change) are represented as trajectories in KDS

A knowledge distribution refers to a distributed knowledge state. *Knowledge processes* are processes that change $KD_{initial}$ into KD_{final}. Knowledge processes can be visualised as trajectories in KDS. As a change in knowledge distribution implies a new action potential (perspectives are part of intentional activities), trajectories in KDS, therefore, may be thought of as corresponding with social change[27].

Simple, unidirectional examples are: an expert acquires additional experience in solving a problem (**c** increases, **a** and **d** remain 1, transition 1 in figure 2); pupils are taught by a mentor (**a** increases, knowledge diffusion, transition 2 in figure 2); a theory is discarded as a potential explanation of some surprising phenomenon (**d** decreases, discarding of an alternative point of view, transition 3 in figure 2).

The three axes of KDS (**c**, **a** and **d**) emphasise different distribution characteristics: either the complexity changes or the adherence changes or the diversity changes. Trajectories in KDS are "blends" of these characteristics: they imply structurally different types of knowledge processes, which are likely to require a different methodological "regime" (i.e. a different set of supporting methods). We will use these very features to answer the central research questions of this chapter in sections 3.4 and 3.5.

Feature 3: the number of perspectives in a KD depends on the position in KDS

The number of atomic perspectives that a KD *consists of* (i.e. its internal *structure*) is related to its position in KDS. The relation can be expressed in a simple formula:

$$\mathbf{n_{structure} = c \cdot a \cdot d} \qquad \textbf{(2)}$$

This follows from the definitions of these three attributes (see formulas 1a-c): **c** is the mean number of atomic perspectives per individual, **a** is the mean number of individuals per point of view, and **d** is the number of different points of view. Therefore, **c.a.d** equals the number of atomic perspectives that the body of knowledge as a whole consists of (see also the example in the box containing figure 1).

[27] To be very precise: social change is at stake in regions of KDS where diversity >> 1, adherence >> 1 and complexity >> 1. Near the axes boundary phenomena manifest themselves, for example; individuals (in isolation) are not social systems.

Feature 4: the number of knowledge steps required to realise a KD (from ground level) depends on the position in KDS

A *knowledge step* is the process that results in the addition or deletion of one atomic perspective to or from a body of knowledge. The *minimal* amount of knowledge steps (i.e. the process) required to *realise* a specific KD starting from the origin (i.e. beginning with "zero knowledge") is also related to its position in KDS. We call this amount $n_{process}$. The formula is:

$$\mathbf{n_{process} = c \cdot a \cdot d} \qquad (3)$$

The above formula is created because each of the atomic perspectives that make up a body of knowledge requires one knowledge step. The "*minimal*" ensures that roundabout ways through KDS are excluded.

Note that equation 2 and equation 3 in combination relate the most efficient *process* of realising a KD to the *structure* of this KD ($n_{structure} = n_{process}$).

Feature 5: stepping through KDS corresponds with societal change; however, some steps are more difficult than others

Equi-distance steps (Cartesian distance) in KDS are quite different in terms of the number of knowledge steps required. For example, going from [**c**=10, **a**=1, **d**=1] to [**c**=10, **a**=2, **d**=1] (following the shortest route) requires far fewer knowledge steps than going from [**c**=100, **a**=1, **d**=1] to [**c**=100, **a**=2, **d**=1]. A real-world counterpart is a teacher, learning a novice to mow the lawn (**c** is small) or to calculate quantum physics (**c** is large).

The *minimal* number of knowledge steps required to realise a change of position in KDS (called n_{step}) depends on the position of start and endpoint in KDS. The relation between n_{step} and position in KDS is:

$$\mathbf{n_{step} = |c_{final} \cdot a_{final} \cdot d_{final} - c_{initial} \cdot a_{initial} \cdot d_{initial}|} \qquad (4)$$

For the previous example, the application of formula 4 would result in:

n_{step} from [**c**=10, **a**=1, **d**=1] to [**c**=10, **a**=2, **d**=1] = **|10.2.1 - 10.1.1| = 10**

n_{step} from [**c**=100, **a**=1, **d**=1] to [**c**=100, **a**=2, **d**=1] = **|100.2.1 - 100.1.1| = 100**

The number of steps corresponding with a curved and complicated trajectory through KDS would be the integral of the number of steps required to travel small segments of this trajectory. The reason for this is, that curved trajectories would underestimate the number of steps (equation 4 assumes the shortest route between start and endpoint). In theory, the steps should be as small as possible.

In order to obtain a more visual understanding of this phenomenon, we introduce the notion of an iso-plane: a plane that connects the points that represent knowledge distributions of the *same* overall structure (a constant number of atomic perspectives) but a different distribution over **c**, **a** and **d**. An iso-plane is described by the equation $\boldsymbol{c} \cdot \boldsymbol{a} \cdot \boldsymbol{d} = constant,\ constant \in \mathbf{N}$. An example of an iso-plane is presented in figure 3a.
A series of iso-planes, of which the constant increases in fixed amounts, gives an impression of the number of knowledge steps required to travel through KDS. Figure 3b presents a two-dimensional cross-section of such a sequence of iso-planes.

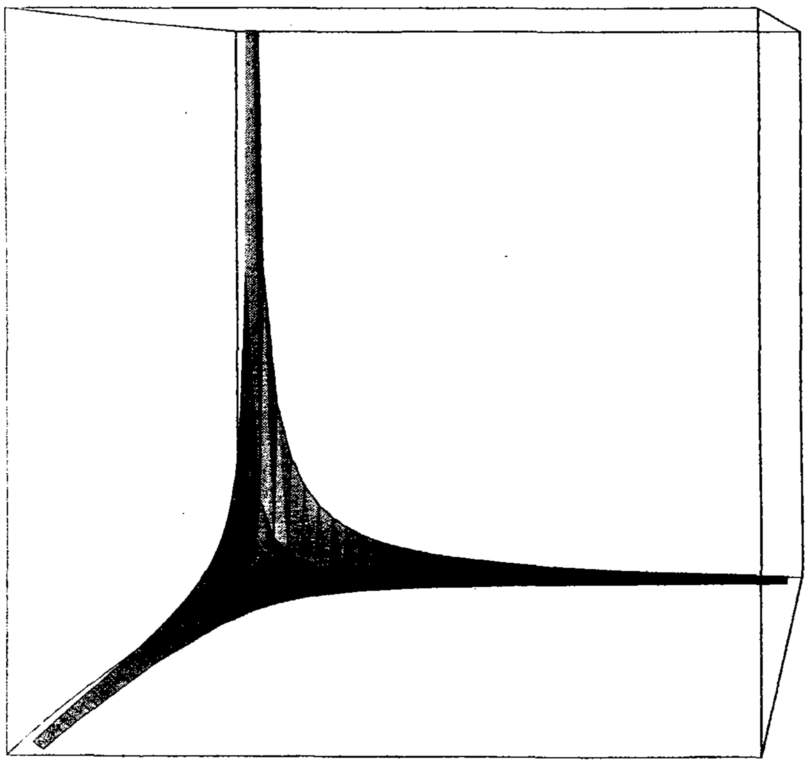


Figure 3a: A three-dimensional iso-plane.

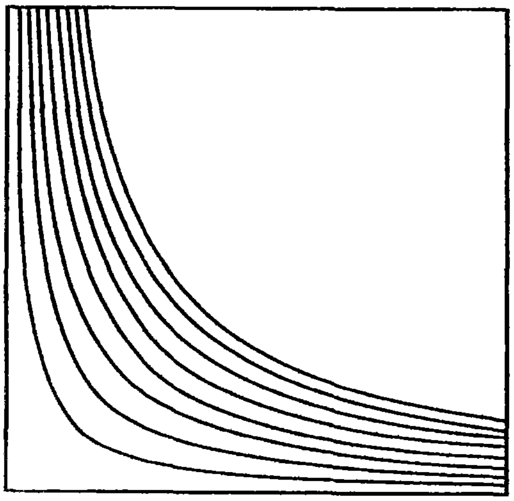


Figure 3b: A cross-section of a sequence of iso-planes.

A sequence of iso-planes, as presented in figure 3b, resembles altitude lines on a hiking map. The number of altitude lines on a map to be crossed per distance is an estimator for the difficulty (the physical effort) to be expected. In KDS, an iso-plane is defined by a fixed number of atomic perspectives in a body of knowledge. When assuming a mean effort per acquired atomic perspective, the number of iso-planes to cross per distance is an estimator for the difficulty (the cognitive effort) to be expected. A series of iso-planes can therefore be understood as **iso-effort planes**. The assumption is that a meaningful value exists for the mean effort per acquired atomic perspective. This assumption relates structure to process to effort.

Admittedly, we have no direct evidence for the structure-process-effort correspondence assumption presented above. Especially on a meso-scale (i.e. the distances are large in comparison with individual knowledge steps, and small in comparison with the overall process), however, the correspondence has a strong intuitive appeal. Some qualitative examples of effort estimates are presented below.

Knowledge diffusion (increasing a).
In the high **c** regions of KDS, it is more difficult to increase **a** than in the low **c** regions. This corresponds with the general observation that spreading complex knowledge (for example, expertise) is more difficult than spreading simple knowledge.

In the high **d** regions of KDS, it is more difficult to increase **a** than in the low **d** regions. This corresponds with the observation that training a good soccer team is more difficult than training a good defender.

Extending perspectives (increasing c).
In the high **a** regions of KDS, it is more difficult to increase **c** than in the low **a** regions. For example, introducing a new colour in the colour code of electrical wire (a disciplinary knowledge process) will cost far more effort than obtaining consensus about changing the password for a personal computer (a personal process).
In the high **d** regions of KDS, it is more difficult to increase **c** than in the low **d** regions. This corresponds with the observation that for a specialist it is easier to extend his discipline in isolation than as part of an interdisciplinary team (or conversely that interdisciplinary research requires far more effort than disciplinary research). Another example is the difference between a soccer team and a chess player: a change in the way in which a mid-fielder plays is likely to require changing the manner in which fellow team members in other positions play. The chess player can change his/her methods of playing without consulting his/her opponent.

Increasing the number of different perspectives (increasing d).
In the high **a** regions of KDS, it is more difficult to increase **d** than in the low **a** regions. This corresponds with the observation that introducing a new perspective requires far more effort in a large group than in a small group. For example, innovation (a rather explorative enterprise) takes place best in rather small teams.
In the high **c** regions of KDS, it is more difficult to increase **d** than in the low **c** regions. This corresponds with the observation that actors tend to "stick" to well-known options, or to consider only one perspective at a time and either "select" or reject this perspective, rather than to explore several complex options and compare them [Beach (1990)].

Problem-solving processes
Problem-solving processes are highly explorative and, in terms of KDS, typically require a large amount of "moving around" without knowing the exact direction. The difficulty of "moving around" (the amount of effort required to move around) depends on both position and direction in KDS, as this determines the number of knowledge steps required to move in KDS. Moving around is difficult if one of the three attributes is large and the direction of movement is perpendicular to the corresponding axis (see figure 3b). In problem situations in which more than one attribute is high-valued, combinatorial effects manifest themselves: moving around becomes more and more difficult. An example is a multi-expert problem-solving process (high **c**, high **d**, low **a**). The worst case is problem situations in which all three attributes are substantial. Examples are interdisciplinary research and ethnic disagreements. Problem solving may become extremely difficult, in agreement with the large number of knowledge steps required to change position in KDS.

Many (if not most) environmental problems are characterised by the involvement of several individuals, disciplines, social groups, adhering to and acting according to widely varying perspectives. In terms of the KDS framework, they tend to be of the combinatorial type: according to the structure-process-effort correspondence assumption, solving these problems therefore must be expected to be difficult. This expectation corresponds with experience: solving environmental problems (or, less ambitious, managing our environment) is an extremely complex, effort-consuming enterprise indeed.

Feature 6: alternative KDS interpretations of *one* body of knowledge are positioned on the same iso-plane

In essence, a body of knowledge consisting of k atomic perspectives can be interpreted as a KD in a fixed number of different ways. Changing interpretation is a cognitive change made by an interpreter: the object that is being interpreted (the referent, in this case a body of knowledge) does not change. Rather, the systemic level of the *model* changes.

A system of k atomic perspectives can, for example, be interpreted as one overall complex perspective possessed by one actor [**c**=k, **a**=1, **d**=1]; as one perspective of complexity 1 shared by k individuals [**c**=1, **a**=k, **d**=1]; or as k different perspectives [**c**=1, **a**=1, **d**=k]. These are extreme interpretations. Intermediate interpretations are also possible. The boundary conditions for possible interpretations are that **c.a.d** = k and that **c**, **a** and **d** $\in$ N.

An example is an interpretation of a university. A university can be thought of as a system consisting of individuals, disciplinary groups, faculties, et cetera. The university as an object, however, does not change. Rather, the interpretation, the systemic level of the model changes.

Another example is the "fix tire" example (shown previously). Fixing the tire can be understood as one overall intentional activity of an individual (**c** = k, **a** = 1, **d** = 1), or perhaps as a system of 20 intentional activities (**c** = k/20, **a** = 1, **d** = 20), or perhaps as a system of 100 intentional activities (**c** = k/100, **a** = 1, **d** = 100). Different interpretations change the number of perspectives, hence the number of intentional activities that are explicitly distinguished as different points of view in a body of knowledge (i.e. they are not "hidden" in the complexity or the adherence attribute, but distinguished as separate points of view in the diversity attribute). For example, in manufacturing automobiles, the introduction of assembly lines resulted from interpreting the assembly of a car (until then a body of knowledge distributed over only a limited number of team members) on lower systemic levels. After this change in interpretation, the body of knowledge of concern was distributed over (or, perhaps more to the point, divided among) far more persons (an actual change in the referent of this KD), which introduced a revolution in automobile manufacturing.

Visually (in terms of KDS), alternative interpretations of one and the same referent (body of knowledge) are points positioned on a plane **c.a.d** = k (an iso-plane, see figure 3a). *Changing interpretation of a body of knowledge corresponds with moving on an iso-plane.*

In Chapters 4 and 5 of this dissertation, moving on iso-planes will be the basis for transformation modelling steps and transformation modelling strategies (transformations change the systemic level of a description of a referent by means of either abstractions or specifications).

Feature 7: actually changing a body of knowledge itself can be understood as a jump towards a higher or lower iso-plane
A body of knowledge can be extended with another body of knowledge, resulting in a new body of knowledge. Likewise, a part of a body of knowledge can be restricted, resulting in a smaller body of knowledge. These are changes that modify the *referent* of a KD (i.e. a body of knowledge of concern), rather than the interpretation (see feature 6). Whereas changes in *interpretation* of a body of knowledge can be visualised as movements *on an iso-plane*, changes in the body of knowledge *itself* can be visualised as a migration *from one iso-plane to another*. A numerical example will illustrate this.

Consider three different bodies of knowledge.
The first body of knowledge can be characterised as KD1 at **c** = 30, **a** = 5 and **d** = 8, and (therefore) is located on iso-plane 30.5.8 = 1200.
The second body of knowledge can be characterised as KD2 at **c** = 70, **a** = 3 and **d** = 2, and is located on iso-plane 420.
The third body of knowledge can be characterised as KD3 at **c** = 41, **a** = 7 and **d** = 4, and is located on iso-plane 1148.

The *total* number of atomic perspectives in these bodies of knowledge is 30.5.8 + 70.3.2 + 41.7.4 = 1200 + 420 + 1148 = 2768. In other words: the total body of knowledge (which is the sum of these three bodies of knowledge) can be characterised as KD_{total} that is located on iso-plane 2768. Note that KD_{total} refers to *another* body of knowledge than the three parts in isolation (to be more specific: to the sum of these three).

The exact position of KD_{total} can be calculated by means of the equations for complexity, adherence and diversity (see equations 1):

$$\mathbf{c_{total} = \Sigma\, c.a.d\ /\ \Sigma\, a.d = 2768 / 74 = 37.405}$$
$$\mathbf{a_{total} = \Sigma\, a.d\ /\ \Sigma\, d = 74 / 14 = 5.285}$$
$$\mathbf{d_{total} = \Sigma\, d = 14}$$

The structure of this total body of knowledge must equal the sum of the structures of its parts. Indeed,

$$37.4054 \cdot 5.2857 \cdot 14 = 1200 + 420 + 1148 = 2768.$$

The total body of knowledge therefore can be characterised as KD_{total} at **c** = 37.4, **a** = 5.3 and **d** = 14, and is located on iso-plane 2768.
This interpretation informs us that the KD consists of 14 *different* points of view, each of them *shared* by 5 individuals (mean) and *consisting of* 37 atomic perspectives (mean).

Visually (in terms of KDS), *changing a body of knowledge corresponds with moving between iso-planes.* In Chapters 4 and 5 of this dissertation this will be the basis for building blocks modelling steps and building blocks modelling strategies (which change the very system of concern, rather than its interpretation, by means of either extensions or restrictions).

In summary, KDS is a remarkable space indeed:

- Bodies of knowledge (snapshots of parts of society) can be represented as points in KDS, and knowledge processes (changes in parts of society) can be represented as trajectories in KDS.
- Changes in actual bodies of knowledge can be given an effort interpretation.
- A specific body of knowledge can be interpreted at several levels of abstraction. Re-interpretations of the same body of knowledge in terms of **c**, **a** and **d** can be visualised as a movement in a specific iso-plane[28]. This preludes the notion of transformation modelling steps and strategies in Chapter 4, and their *Trinity* equivalents in Chapter 5.
- A body of knowledge (i.e. the referent of a KD) can be made larger or smaller, again resulting in a body of knowledge (the notion of knowledge is self-contained). Extensions and restrictions of bodies of knowledge can be visualised as movements between different iso-planes. This preludes the notion of the building block modelling steps and strategies in Chapter 4, and their *Trinity* equivalents in Chapter 5.

3.4 A TYPOLOGY OF ENVIRONMENTAL SITUATIONS OF CONCERN

In this section, we will use the KDS framework to address the first research question: *is it possible to classify problem situations based on the way in which knowledge of relevance is distributed?*
Several ways of classifying environmental problems exist at this moment. Examples of problem typologies are (see also [Glasbergen (1994)]):

1. typologies based on the *discipline* that studies the problem (environmental chemical problems, environmental sociological problems);

[28] This is the reason why moving **on** an iso-plane does not require effort in terms of an actual change in the referent (see feature 4): it is the *interpretation* that changes, not the *referent*.

2. typologies based on the physical *compartment* where the undesired effect manifests itself (air pollution problems, soil pollution problems);
3. typologies based on a specific *physical agent* that causes problems (the CO_2 problem, Cadmium problems, acidification);
4. typologies based on themes, relevant from a *governmental policy* point of view (in the Netherlands e.g. acidification, spill (Second National Environmental Policy Plan, The Netherlands, 1993-1994));
5. typologies based on the *actors* and *sectors* whose actions result in environmental problems (e.g. problems related to agricultural activities);
6. typologies based on the specific *region* (geographical site) where problems manifest themselves;
7. typologies based on the *scale* at which problems manifest themselves (e.g. local, regional, national, international, global).

A typology based on knowledge distribution criteria is quite different from the example typologies mentioned above: it highlights structural aspects of knowledge distributions and knowledge processes, rather than domain-related semantic aspects. This is of great importance for managing environmental problem-solving processes, as it highlights methodological differences, independent of the specific problem of concern. For practical reasons, we will use a qualitative interpretation of KDS as a basis for a problem typology.

A qualitative interpretation of Knowledge Distribution Space

Below, each of the three knowledge distribution attributes **c**, **a** and **d** will be given a qualitative interpretation. Subsequently, this qualitative interpretation will be superpositioned on KDS, resulting in a problem typology based on knowledge distribution criteria.

Complexity: Several authors mention differences in complexity[29] of points of view. A well known qualification is the range Beginner, Advanced Beginner, Competent Performer, Proficient Performer and Expert, presented by Dreyfus and Dreyfus [1986] and extended by Wigg [1993]. For our purposes, however, a bisection suffices: only low **c** points of view and high **c** points of view will be distinguished.

Diversity: A binary distinction is made between the situation in which one actor is involved, and the situation in which multiple actors are involved. The reason is that for **d** > 1, co-ordination (finding and maintaining coherence between points of view) is required. High **d** knowledge distributions introduce emerging properties: properties of the KD as a whole, that are not properties of the perspectives in isolation [Checkland (1990)]. The need for a co-ordination perspective is an example of this phenomenon.

Adherence: Finally, the adherence axis is divided into four regimes: *personal*, *disciplinary*, *cultural* and *common* points of view. These regimes are presented in figure 4a [Diepenmaat (1993b)], and will be explained below.

[29] The term "complexity" as an indicator for the number of perspectives is not actually used in these references, as it is introduced in this dissertation. The notion, however, can be recognised.

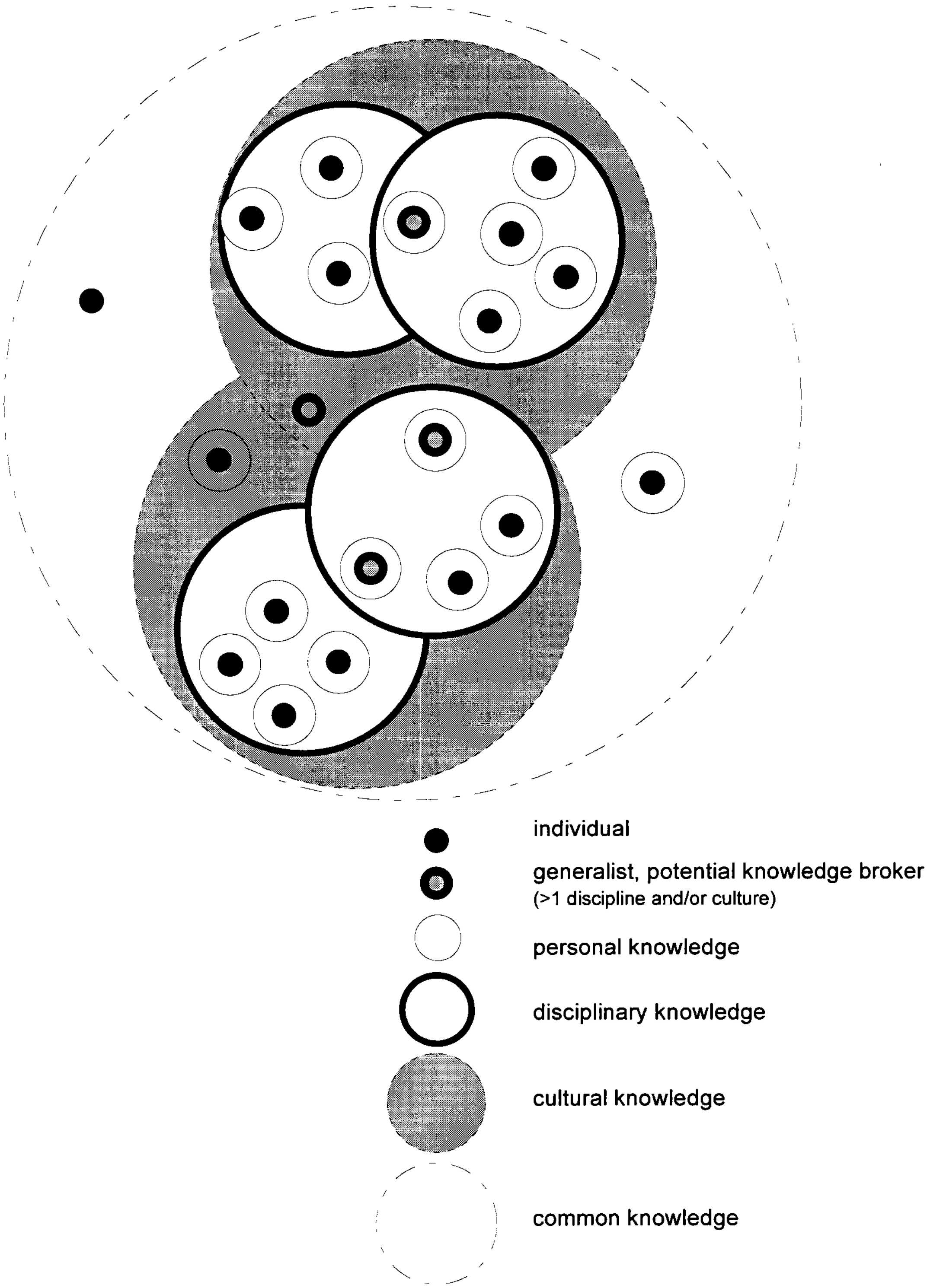


Figure 4a: Knowledge types according to the adherence attribute.

Personal knowledge (a personal perspective) is possessed by one person. An example is the perspective of an expert.
Disciplinary knowledge (a disciplinary perspective) is shared by the members of a group trying to achieve the same goal for the same purpose, using the same set of theories and methods [Mittelstrass (1993)]. An example is the knowledge required to investigate chemical atmospheric reactions.
Cultural knowledge (a cultural perspective) results from social processes. Social processes resemble disciplinary processes (the members of a discipline are a social group), but the *perspective* involved is less explicit. The route to acquire cultural knowledge is rather implicit as well: one has to function as a member of the group. High **a** examples are ethnic perspectives, low **a** examples are corporate cultures.
Common knowledge (a common perspective) is considered to be shared by all persons involved. It is not required to communicate this kind of knowledge: it rather serves as "common background" for communication.
Note that the notion of shared knowledge is relative with respect to the problem context of concern. For example, what is considered to be shared knowledge *within* a discipline may become highly disputable *between* disciplines. This phenomenon can be recognised in the differences between the lower four examples of figure 5.

The difference between disciplinary knowledge and cultural knowledge is sometimes subtle (any discipline shows cultural behaviour, and disciplines may be recognised within any culture), but it is important from a problem-solving point of view. When a goal is stated explicitly (disciplinary knowledge), it is easier to test whether specific adaptations to the disciplinary knowledge make sense (for example, a new atmospheric reaction theory provides better predictions and explanations than existing theories). In the case of cultural knowledge, realising change may be problematic: cultural perspectives in general are tacit and deeply rooted in a long period of group membership.
Cultural perspectives tend to be of a much higher complexity than would appear from a superficial analysis. For example, "knowing that others know" increases complexity. This phenomenon is an emerging property of high **a** perspectives: it makes cultural perspectives rather refractory.
A difference between disciplinary knowledge on the one hand and personal and cultural knowledge on the other is that for disciplinary knowledge an explicit infrastructure exists in which one acquires the knowledge of the discipline (for example, a school, textbooks, on-the-job training sites). As a result, disciplinary knowledge is easier to communicate than cultural or personal knowledge. In this infrastructure, cultural knowledge is acquired as well, but takes place rather implicitly.

A typology of environmental problems

The problem typology results from applying a combinatorial scheme on the qualitative interpretations of the three axes presented above. In the complexity/diversity plane of KDS the letters A, B, C and D are situated, reflecting the binary distinctions made on these axes (see figure 4b). The subscripts personal, disciplinary and cultural shift this ABCD typology along the adherence axis.

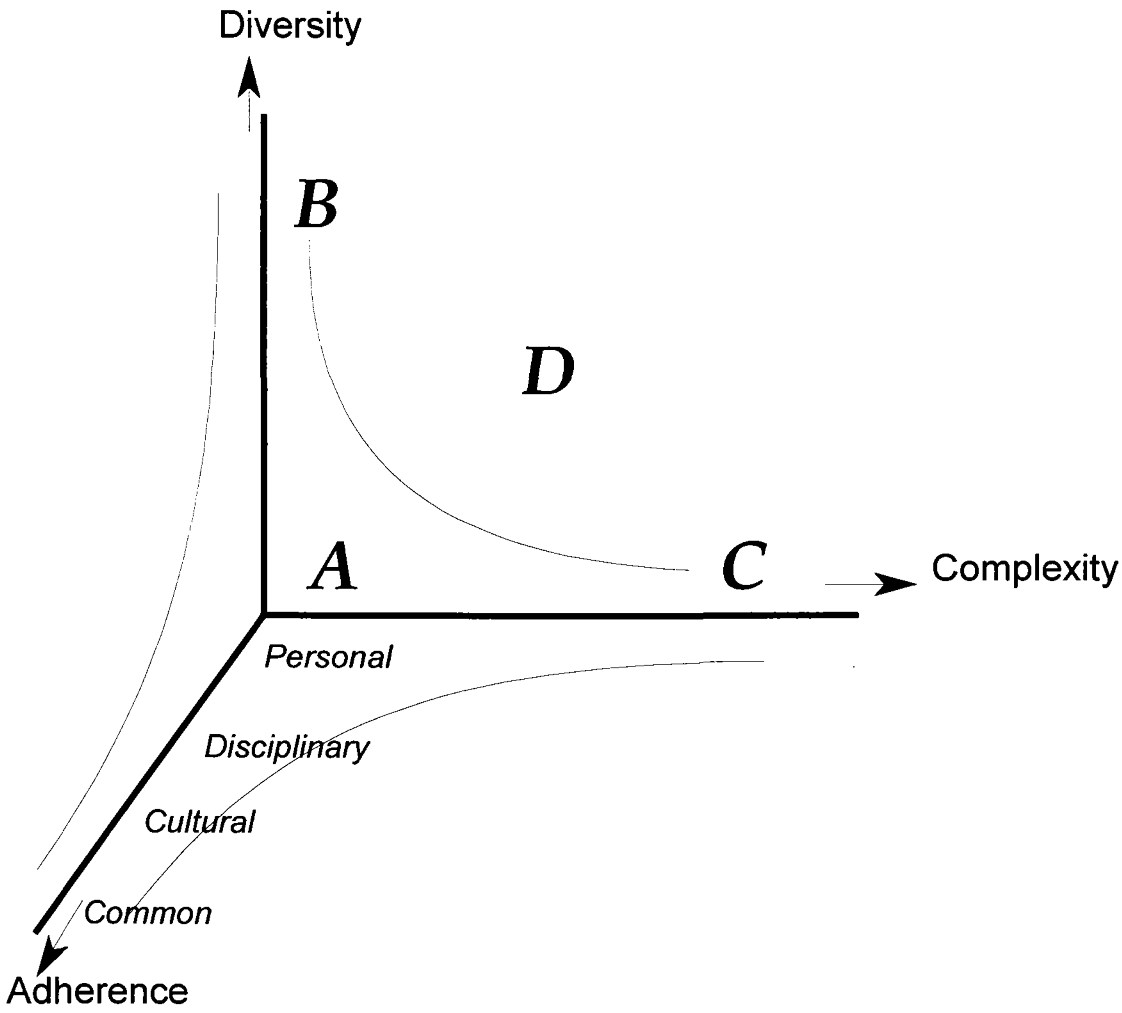


Figure 4b: The ABCD distinction.

A-type problems in principle are easy problems. Only one actor is involved in the problem-solving process. On top of this, the perspective of concern is not complex. Three subtypes exist: A_{pe}-type problems, A_{di}-type problems and A_{cu}-type problems. Take note that, although simple in principle, in high **a** regions of KDS it may require great efforts to solve the problem due to quantity and repetition effects.

B-type problems do not involve intrinsically complex perspectives: **c** is low. However, the involvement of several different perspectives (**d** being high) introduces a need for coherence. On top of the **d** relevant perspectives, a co-ordinating meta-perspective emerges: the **d**+1 perspective. This co-ordinating perspective will become more complex if diversity increases. Again, three subtypes exist:
B_{pe}-type problems: several simple personal perspectives must be co-ordinated.
B_{di}-type problems: several simple disciplinary perspectives must be co-ordinated. Examples are simple cases of multi-disciplinary problems.
B_{cu}-type problems: several cultural perspectives must be co-ordinated. An example is a fusion process between companies with a comparable but not identical corporate culture.

C-type problems involve only one perspective (**d** = 1). However, this perspective is complex (and recursively consists of many perspectives). The three subtypes are:
C_{pe}-type problems are those that require someone who possesses unique and complex knowledge, like an expert. Once an expert has been identified, solving the problem is easy. Note, however, that using the expert's answer may be easy, but understanding the perspectives that he/she uses during his/her reasoning processes in general is difficult (which is a logical consequence of the high **c** nature).
C_{di}-type problems are those that require trained members of one discipline. Participants are not unique (as in the expert case), but the knowledge required is intrinsically complex. An example is fixing a television set.
C_{cu}-type problems are those that require experienced members of a social group. For example, when visiting another culture, it may be important to first obtain thorough information about the culture by consulting a member.

D-type problems are worst cases: they involve finding or achieving coherence (high **d**) between complex (high **c**) perspectives. As in B-type situations, a **d**+1 meta-perspective emerges (this is the D-type perspective to be presented in Chapter 5). Unlike C-type situations, many cases require one to understand the individual perspectives in order to find the **d**+1 co-ordination perspective. This may require considerable cognitive effort. The three subtypes are: D_{pe} (involvement of several experts), D_{di} (involvement of several disciplines), and D_{cu} (involvement of several cultures).

Table 1 presents a survey of the problem types.

Table 1: A problem typology based on knowledge distribution.

COMPLEXITY	DIVERSITY	ADHERENCE	PROBLEM TYPE
low	1	personal	$A_{personal}$
low	1	disciplinary	$A_{disciplinary}$
low	1	cultural	$A_{cultural}$
low	> 1	personal	$B_{personal}$
low	> 1	disciplinary	$B_{disciplinary}$
low	> 1	cultural	$B_{cultural}$
high	1	personal	$C_{personal}$
high	1	disciplinary	$C_{disciplinary}$
high	1	cultural	$C_{cultural}$
high	> 1	personal	$D_{personal}$
high	> 1	disciplinary	$D_{disciplinary}$
high	> 1	cultural	$D_{cultural}$

3.5 METHODOLOGICAL CONSEQUENCES FOR PROBLEM SOLVING

In this section, we address the second research question: *does the problem typology presented in section 3 provide methodological guidelines for the subsequent problem-solving process?*

The answer to this question is in the affirmative. Different regions of KDS exhibit different knowledge distribution features. In attempts to solve a problem, these features are important: they imply different sources of complexity for the subsequent problem-solving process. Different sources of complexity in turn imply different problem-solving principles, and different knowledge and skills profiles of the intended problem solvers. In short: position and direction in KDS (or, more generally speaking, regions in KDS) are closely related to methodological problem-solving considerations.

Potential problems associated with the three attributes are presented below:

d: co-ordination

If **d** is large, many different points of view must be taken into account. Co-ordination (obtaining and maintaining coherence between these points of view) is likely to be a key item in the problem-solving process. This typically requires intensive communication.

c: obtaining an understanding of perspectives

If **c** is large, the perspective(s) involved consist(s) of many recursive perspectives. As a consequence, it is likely to take much effort to obtain an understanding of, or to change the perspective(s) of concern (e.g, understanding the perspectives of an expert may be difficult).

a: repetition and cultural effects

If **a** is large, repetition and cultural effects may be expected. Repetition effects simply result from the fact that many individuals are involved. Cultural effects are related to the observation that high **a** bodies of knowledge tend to be tacit and habitual. This makes it difficult to understand explicit and change cultural perspectives.

combinatorial cases

In combinatorial cases, the attribute-specific bottlenecks will present themselves in combination. For example, in a $D_{personal}$ situation, a large **c** and a large **d** manifest themselves in combination: complex perspectives must be understood, potentially changed, and co-ordinated (brought into coherence). This results in a need for a combinatorial problem-solving paradigm (which is not the same as sequentially applying unary techniques).

These observations will be used to relate each of the four main problem types (A B C and D) to problem-solving skills and archetypical problem solvers.

A-type problems do not require particular problem-solving skills; from a methodological point of view they are easy to solve.

A key aspect in solving B-type problems is *co-ordination*: coherence between low **c** perspectives of several actors must be found and maintained during the problem-solving process. A typical problem solver in B-type situations is a **co-ordinator**.

C-type problems require availability of a **specialist**. From a methodological point of view, problem-solving principles are dictated by a specialist: they are domain-specific.

D-type problem situations are "worst case". They are characterised by a lack of coherence and organisation in a complex, multi-perspective, interdependent context. Consequently, the key to a solution is a form of coherence between perspectives, in which the original problem is solved, *and* no new insurmountable problems are introduced. In finding such a solution, B-type, C-type and (potentially) high **a** problem characteristics will be encountered.

We call a prototypical problem solver for D-type problems a **knowledge broker**. (To our knowledge, the term "knowledge broker" was first introduced in [Hawkins (1985)]). We will call the problem-solving paradigm of environmental knowledge brokers *Environmental Knowledge Brokerage (EKB)*.

A professional environmental knowledge broker typically performs his/her problem-solving activities on behalf of the actor actually owning the problem: the primary problem owner. This actor is somehow responsible for the D-type environmental problem situation. Typical examples of this phenomenon are government officials, top managers, or representatives of the actors involved in the problem situation.

A knowledge broker is working in a problem context, involving many different perspectives: personal, disciplinary and cultural. A commonly used strategy in these situations is to train people as *generalists*. Generalists differ from specialists in that they

posses a global overview of several disciplinary and cultural perspectives. This puts them ahead in the process of understanding individual perspectives. Furthermore, possessing an overview of many disciplinary and cultural perspectives is a natural counterbalance for being pre-occupied.
However, the emerging and confronting nature of environmental problems turns anticipation into a difficult enterprise. In specific problem situations, anticipative strategies like "train generalists" may be of help. In general, however, rather than *possessing* advance knowledge, the ability to *obtain, communicate and integrate* knowledge within reasonable limits of time and effort is a crucial requirement for knowledge brokerage activities. This implies an orientation towards knowledge *processes* rather than towards knowledge *distributions* (possessing a large amount of advance knowledge).
In summary, a knowledge broker can be characterised as:

- a domain knowledge generalist;
- a knowledge management specialist; and
- a person, oriented with the knowledge possessed by the other actors involved.

An overview of the relation between problem types, methodological considerations and typical problem solvers is presented in table 2. Different types of generalists and "mixed" bodies of knowledge are presented in figure 5.
A mismatch between problem type and problem solver typically has a rather negative influence on the effectiveness and/or the efficiency of problem-solving processes. For example, appointing a general manager in a D_{di}-type problem situation would be an unwise cast, as would be appointing a domain expert. It is interesting to note that a specific type of problem solver is difficult to appoint for many of the specific problem types present in table 1. Sending either former top politicians or armies to ethnic problem sites may be common practice, but it is hard to conceive of the way in which this would eliminate cultural differences. Existing alternatives, however, are difficult to bring forward. Table 2 might be expanded into a matrix of twelve rows (by means of an explicit representation of the problem subtypes). Filling in such an expanded matrix in depth would be an interesting research activity indeed.

Table 2: Problem types, methodological principles and typical problem solvers.

PROBLEM TYPE	METHODOLOGICAL PRINCIPLES	TYPICAL PROBLEM SOLVER
A	Trivial	Trivial
B	Co-ordination	Co-ordinator / General manager
C	Domain-specific	Specialist
D	Knowledge modelling, knowledge communication and knowledge integration methods	Knowledge broker

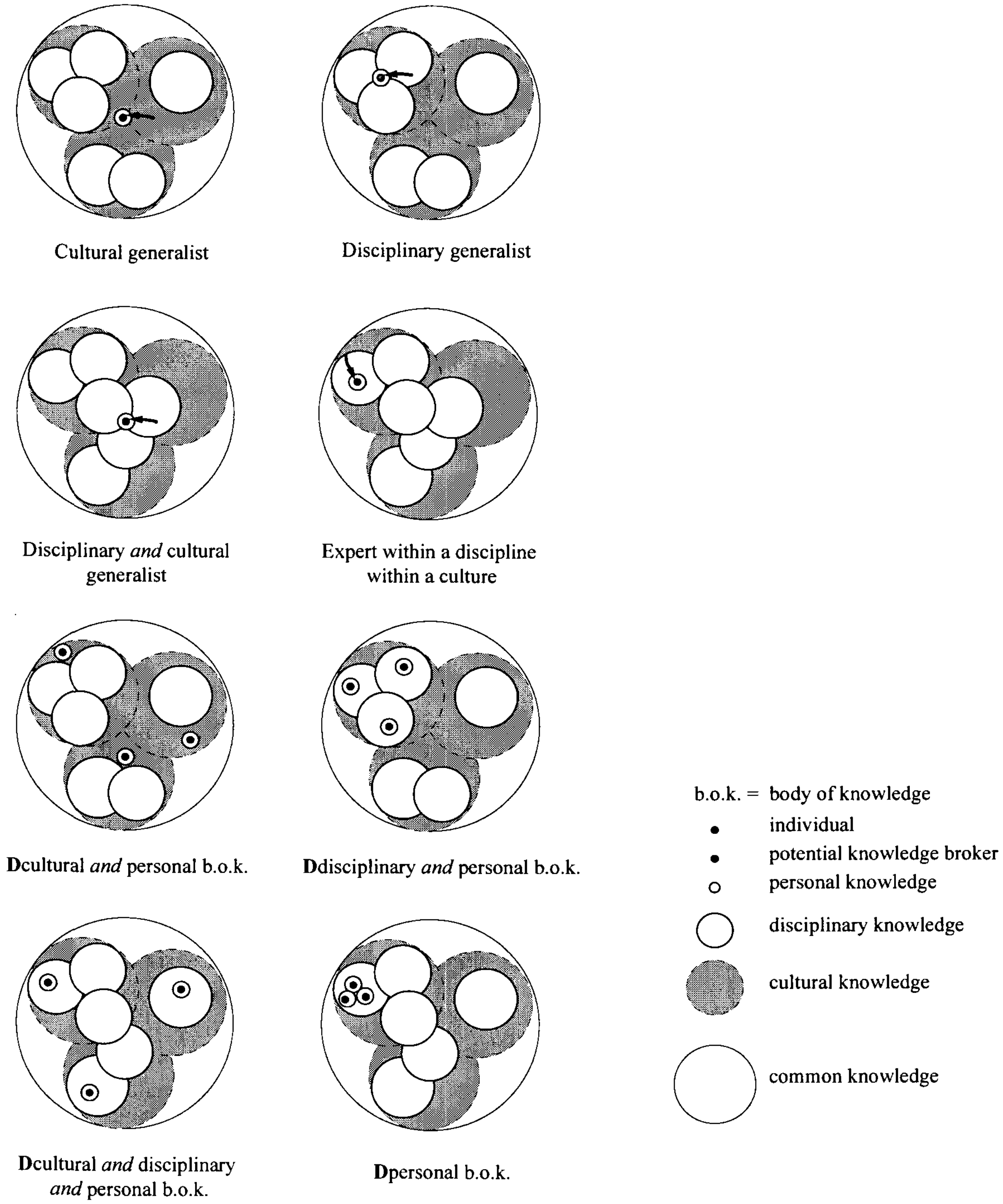


Figure 5: Different examples of bodies of knowledge (a qualitative representation).

3.6 DISCUSSION AND CONCLUSIONS

We investigated the relation between knowledge distributions and knowledge processes (of which problem-solving processes are a special case). More specifically, we addressed two research questions:

> *"Is it possible to classify situations of concern based on the way in which knowledge of relevance is distributed over individuals?"*

> *"If so, does this distinction provide (methodological) guidelines for the subsequent problem-solving process?"*

We did so on the basis of the KDS framework, a framework that defines and visualises knowledge distributions and knowledge processes on the basis of the notion of a perspective: a static notion of knowledge that is closely connected with an action potential.

KDS: a quantitative basis, a qualitative interpretation

KDS is defined in mathematical terms. This mathematical foundation suggests a level of precision that is not intended. The KDS framework exhibits a dualism: KDS is *defined* quantitatively, as a discrete space (indeed, it is a quantum space), but is *used* qualitatively and as a continuous space. It allows visualisation of both knowledge distributions and knowledge processes (*any* knowledge distribution and *any* knowledge process, as long as the definition of knowledge in terms of perspectives is accepted). Effort estimates of knowledge processes, the ABCD problem typology, and the methodological interpretation of problem types are examples of the qualitative use of our framework.
For now, we conclude that the KDS framework is a useful framework, in that it offers a platform to investigate knowledge distributions, knowledge processes and their relations. In Chapters 4 and 5 of this dissertation, many of the features of KDS will be used as a foundation for designing model relations, modelling steps and modelling strategies.

As an answer to the first research question: *"Is it possible to classify problem situations based on the way in which knowledge of relevance is distributed?"*, we presented the ABCD problem typology, which basically is a division of KDS into regions.
As an answer to the second question: *"Does such a classification provide (methodological) guidelines for the subsequent problem-solving process?"*, we showed that the typology helps in identifying potential bottlenecks for problem solving, in selecting appropriate problem-solving principles, and in associating a specific type of problem solver. The ABCD typology merely functions as a simplification of KDS. In principle, each region in KDS is related to specific problem solving bottlenecks, and consequently requires specific problem-solving principles and problem solvers. In this chapter, this observation is explored rather than elaborated. In principle, *any* method, methodology or discipline that supports knowledge processes (e.g. knowledge acquisition, knowledge modelling and knowledge transfer methods, and communication disciplines) can be assigned to regions in

KDS. For example, knowledge acquisition is a discipline that offers methods and techniques directed at obtaining an understanding of expertise knowledge, in order to distribute this knowledge (typically by means of some computer programme; see, for example, [Wielinga et al. (1990)], [Steels et al. (1994)]). These methods and techniques are closely related to $C_{personal}$ problem situations, and more specifically the trajectory [**c**=high, **a**=1, **d**=1] -> [**c**=high, **a**>1, **d**=1]. At this moment, also knowledge acquisition from multiple experts is an important research topic within this discipline. On the other hand, mass communication and public informing methods must be situated in $B_{disciplinary}$ and $B_{cultural}$ regions of KDS. In $D_{personal}$ situations, methods specifically suited for group communication between experts can be used (see, for example, [McGraw and Seale (1988)], [Dieng et al. (1994)]).
Completing this mapping requires a substantial research effort, but the result is likely to be a valuable tool in managing D-type (environmental) problem-solving processes.

The D-type nature of environmental problems

Environmental problems in many cases must be typed D. In worst case situations, they involve many different (**d** = high) societal and disciplinary actors (**a** = high) performing according to complex (**c** = high) points of view. Solving them is likely to cost much effort, and methodological guidance is scarce.
As dedicated problem solvers, we introduced the environmental knowledge broker. Knowledge brokerage is related to many different disciplines (see, for example, [Checkland and Scholes (1991)], [Wigg (1993)]). Basically, environmental knowledge brokers can be characterised by three features: they are oriented with knowledge possessed by other actors; they are environmental generalists; and they are specialists with respect to supporting knowledge processes (cf. the knowledge professional, as described in [Wigg (1993)]). Especially the first and last of these characteristics are underestimated aspects in environmental problem solving: they emphasise *dynamic* aspects (the problem-solving process, the ability to change knowledge distributions) rather than *static* aspects (possessing a large quantity of advance knowledge).
In many cases, D-type problem-solving processes are managed by general managers (B-type problem solvers) or by specialists (C-type problem solvers), or by both. In the latter case, a well-known problem is that general managers and domain specialists are not known to communicate very well. In addition, our analysis makes clear that none of the three possibilities (general managers in isolation, specialists in isolation, or simply "adding" specialists and a general manager) addresses the specific problems of a D-type situation. This is so because at the start of a D-type problem-solving process a deadlock situation exists. In order to understand the relation between the **d** perspectives, the perspectives must be investigated; in order to know what perspectives to investigate, their relations must be known. As a result, a combinatorial (and likely incremental) problem-solving paradigm is required which explicitly deals with this deadlock situation.
In a sense, the knowledge and skills profile of an environmental knowledge broker is a mixture of two worlds. As a generalist, he/she is able to quickly obtain a global understanding of the domain knowledge of the specialists involved. As a knowledge

manager, he/she is focused on understanding and bringing into coherence complex perspectives. In combination, these two facets eliminate (or at least diminish) the deadlock situation mentioned above.

Methodological support for D-type problem solving

D-type environmental problems exist, and continue to emerge at an increasing rate. Many individuals at this moment function as knowledge brokers *"avant la lettre"*, in attempts to solve these problems. The problem-solving approaches in use are largely dictated by experience and common sense. As a general rule, this is a sound basis for any intentional activity. However, in order to be further effective and efficient in D-type problem solving, an increased effort in methodological research with respect to D-type problem-solving processes is required. This might turn knowledge brokerage into a profession (although, admittedly, some aspects of knowledge brokerage activities will remain difficult to explain).

Contemporary environmental science clearly shows the importance of understanding systems with a large overall complexity. Examples are integral chain management [Association (1992)], national and international emission inventories, and life cycle analyses [Heijungs et al. (1992)]. These examples, although directed towards complex systems, show that existing models and approaches in environmental science too often concentrate on physical aspects of environmental problems. In this chapter, we emphasised knowledge distribution aspects. However, in a D-type situation the perspectives of actors are related to each other (the situation is D-type, rather than a set of C-types). Actors use perspectives to act in a shared physical environment, or to inform or affect each other by means of communication. As a result, a methodology directed at D-type environmental problem solving must encompass methods that integrate intentional aspects (involving perspectives of actors), physical aspects and communicational aspects of environmental problem situations. At this moment, knowledge of such integral approaches is limited. This is a serious bottleneck: a technically feasible solution that is not acceptable for the actors intended or not communicated in an effective way, simply does not solve the problem.

As in many cases, the main result of the research presented in this chapter is a new research question: what are appropriate methods for solving D-type problems in general, and environmental problems in particular? The research to be presented in subsequent chapters is directed at further development of environmental knowledge brokerage as a paradigm for D-type environmental problem solving. The central issue is the development and use of *Trinity*: a model-based approach to support knowledge brokers in D-type problem solving. *Trinity* is based on a theory of problems and problem solving, supports several stages of intentional activities, addresses the deadlock situation previously mentioned, and provides a conceptual modelling language that enables one to represent intentional actors, communication processes and actions in the physical environment in one integral model.

CHAPTER 4

A GENERIC THEORY OF QUALITATIVE MODELLING

4.1 INTRODUCTION

In this chapter, we will develop a generic theory of qualitative modelling processes. This theory presents guidelines for the design of qualitative modelling languages in general. The theory will form the basis for developing the *Trinity* modelling language: a qualitative language providing modelling methods that support multi-actor problem-solving processes. The *Trinity* modelling language will be presented in Chapter 5 (Part IV: Methods of this dissertation).

Many qualitative modelling paradigms exist. Examples are the Yourdon approach for modelling information systems, the IDEF approach in systems analysis, and the flowsheeting paradigm in process engineering.
In the Yourdon approach, Data Flow Diagrams (DFDs) are models of information processing operations. Information processes are represented by means of bubbles. These processes take data as input, and result in data as output. These data flows are represented by means of arrows. The data output of process a may be the data input of process b, which introduces the notion of linked information processes (and thus more complex models).
In process engineering, the flowsheeting modelling paradigm provides an example: process unit operations (like heat exchangers, blenders, et cetera) are connected (related) by means of material and/or energy flows.

These modelling paradigms share the feature of using symbols to refer to certain types of concepts. Concepts specify or delimit objects. These objects may be related to each other in many different ways (for example, in time, space, or any other dimension). Relational terms typically are represented by means of arrows or lines (see also [Meehan (1988) Chapter 5]).

In many cases, these qualitative modelling paradigms have a strong empirical foundation: typically they result from a need to operate in a more structured way when analysing or

designing a software product, a factory or another complex object or artefact. As a general rule, this is a sound reason to develop such a paradigm. Indeed, our own attempts to provide model-based support for multi-actor problem solving (resulting in the *Trinity* methodology) are motivated by a need to operate more structuredly in multi-actor environmental problem solving.
In spite of this rather bottom-up development, many qualitative modelling paradigms seem to share certain features: although the domains of application widely differ, notions like states, processes, relations, temporal ordering, the presence of alternatives and recurrent situations are present in many, if not most, paradigms. In addition, the *use* of such a qualitative modelling paradigm implicitly involves sequences of relatively small adaptations to a model, adaptations that have a strong generic nature.
In summary, although a historical analysis might raise the suspicion that every paradigm is unique, a (superficial) comparison suggests quite the opposite: it might very well be possible to develop a theory of qualitative modelling processes that is independent of the domain of application. Having at one's disposal such a theory would be beneficial in a number of ways:

1. it would provide a deeper insight into qualitative modelling processes in general;
2. it would enable us to understand and compare existing qualitative modelling paradigms in terms of a generic conceptual vocabulary; and
3. it would guide the design of new qualitative modelling paradigms.

This chapter presents a generic theory of qualitative modelling processes. First, a description of a *model relation* is presented in terms of the *meaning triangle*: a well-known theoretical concept that relates a symbol and an object by means of a concept. According to the theory to be presented, modelling is a process of turning a model relation into a new model relation, rather than turning a model into a new model. This difference is both crucial and subtle: we will show that some modelling processes do not change the model.
An important implication is that a theory of modelling processes should build upon a clear description of what a *model relation* is. In order to obtain such a description, the notion of a model relation is elaborated, resulting in four different types of model relations.
Subsequently, these four types of model relations are used to derive a typology of twelve *primitive modelling steps*: modelling procedures that enable us to change an input model relation into an output model relation. A modelling process can now be described as a (sequence of) modelling step(s). Each of the modelling steps will be discussed and explained separately.
The set of primitive modelling steps enables one to analyse and design complex modelling processes in terms of sequences of primitive steps.
Some sequences are recurrent (typical). This offers the possibility to develop the notion of a *modelling strategy*: a typical sequence of primitive modelling steps that enables one to meet a specific goal. For example, change the level of detail; increase the scope; extend the scope at the expense of level of detail.

The generic theory of qualitative modelling processes to be presented in this chapter provides guidelines for the design of qualitative modelling approaches. A qualitative modelling approach, designed in line with these guidelines, enables one to change both level of detail and scope of a model, to use multiple representations of one and the same referent, and to construct and use generic models. On top of this, the language will support the use of many modelling strategies, from simple to highly complex. At the end of this chapter, these guidelines will be presented. In Chapter 5 they will be used to design the *Trinity* modelling language to support D-type intentional activities (this type of use of the theory complies with the third benefit distinguished above).

4.2 MODELS, REFERENTS AND MODEL RELATIONS

By means of a model it is possible to explore (form expectations about) an object without actually interfering with this object. For example, a tailor's dummy can be used to make a coat that fits the body being modelled by the dummy. Or a mathematical model of a bridge may be used to calculate the strength of this bridge. In both examples someone using the model assumes that, although object and model are different things, the model mirrors features of the object it represents. In case of the dummy, the morphological features of the dummy are believed to reflect the morphological features of a real body. In case of the bridge, the strength calculated with the model is believed to reflect the strength of the real bridge. A *model relation* exists between model and referent.

The meaning triangle

The two examples presented above reflect a generic pattern, that first was described by Aristotle: the *meaning triangle* [Sowa (1984) p. 11, p. 310], [Kassangola (1989)]. The meaning triangle (figure 1) relates an *object* (referent, extension, the "thing" being modelled), a *concept* (intension, thought, idea, sense, quality, mental entity[30]) and a *symbol* (word, language element). The object is the entity of concern that is being explored without actually interfering with it; the concept is a mental interpretation of this object; and the symbol is a human artefact, referring to (representing) the concept. The mapping between object and concept is commonly referred to as *conceptualisation*; the mapping between concept and symbol is commonly referred to as *representation* (for example, speaking, drawing, writing, i.e. communication).

---

[30] It is common to distinguish "mental representations" and "mental models" as well. When we use the concept "model", however, we refer to a representation that is represented **explicitly**, i.e. perceptible for anybody, for example by means of a scheme, a diagram, in speech, in writing. A relaxed interpretation of "model" would be that the representation, at least in principle, should be translatable into an explicit form. This would include representation media like the "Language of Thought" (the "voice" we use to think) and mental images. When we refer to mental representations or mental models, this will be stated explicitly.

The meaning triangle can be used as a description of a model relation: the object is the referent; the symbol is the model; and the concept relates model and referent (realises the model relation, see figure 1).

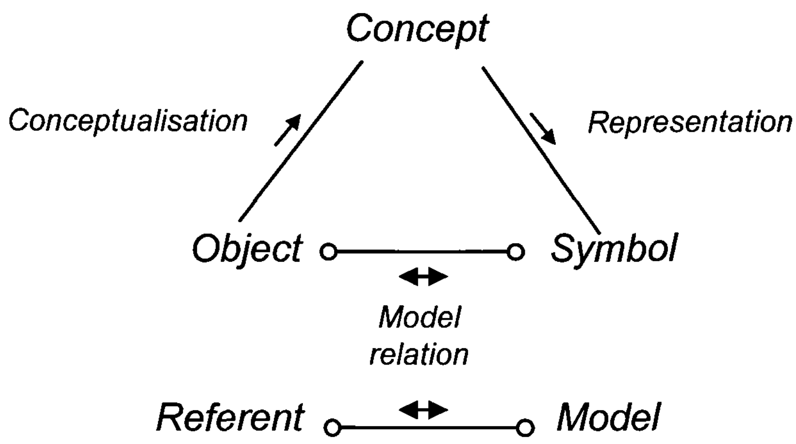


Figure 1: The meaning triangle and a model relation.

The meaning triangle makes clear that object (referent) and symbol (model) are related indirectly, by means of a concept. A model is a representation of a conceptualisation of a referent. Model relations are virtual, mental entities that are linked to a referent and a model. They are not *present* in the referent, nor in the model. They are rather attributed to a referent-model pair by either the modeller or the interpreter of a model. We will represent a model relation as a line between a referent and a model.

Types of model relations

Model relations can be either simple or complex. A *simple model relation* relates one referent to one symbol. A *complex model relation* relates one or more referents to one or more symbols. A complex model relation can be interpreted as consisting of several simple model relations. Three different types of complex model relations can be distinguished: *parallel* relations, *multi-referent* relations and *multi-representation* relations (figure 2). The types of model relations are closely related with the three dimensions of Knowledge Distribution Space[31].

[31] The three types of complex model relations comply with the three dimensions of Knowledge Distribution Space (Chapter 3). The *complexity* of a perspective is the number of perspectives that are distinguished in this perspective, which implies a *parallel* relation. The *adherence* of a perspective is the number of individuals that adhere to it, which implies a multi-referent relation. The *diversity* of a body of knowledge is the number of different perspectives that apply to a referent, which implies a multi-representation relation. Finally, an atomic perspective simply is a perspective of which further decomposition in terms of perspectives is not known, which implies a simple model relation. Indeed, KDS enables one to model bodies of knowledge by means of a

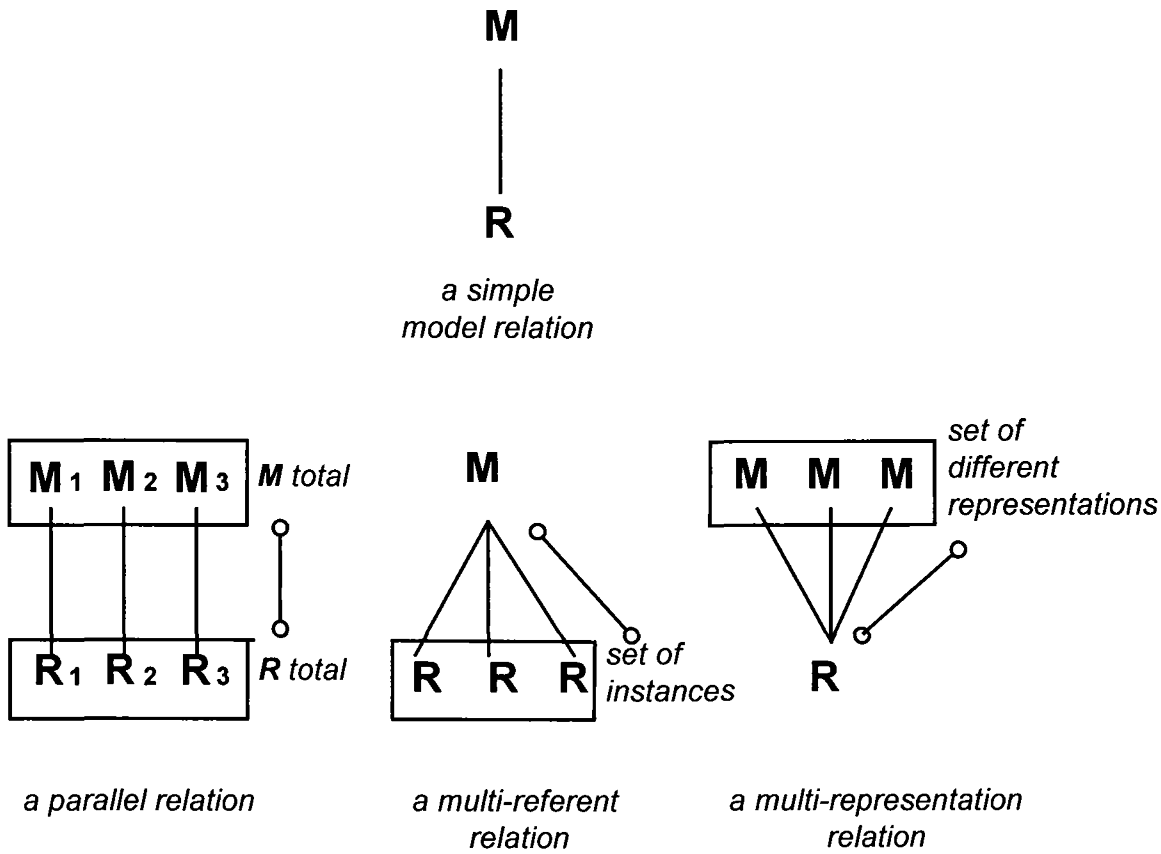


Figure 2: Model relations.

In a *parallel* relation, the number of referents equals the number of models. Each of the model parts has a separate referent part it mirrors. These partial model relations in combination constitute a simple model relation at a higher level of abstraction (relating R*total* and M*total*). The simple relations constituting a parallel relation are not necessarily parallel in time. For example, a chemical reaction, modelled by A → B, consists of three temporally ordered parts: two states and a conversion process.
In a *multi-referent* relation, one model models more than one referent: the cardinality of the model is greater than 1. In the extreme case that the concepts constituting the model relations are completely different, the model is a *homonym* (one symbol, several meanings,

point, and moving through KDS can be interpreted as a modelling process. The same holds true for the *Trinity* modelling language to be presented in Chapter 5, and experiments in *Trinity* modelling to be presented in the Experiments part of this dissertation. *Trinity* modelling can also be understood in terms of moving through a three-dimensional KDS-like space. These relations between the different parts of this dissertation will re-appear in the general discussion of this dissertation.

see figure 3 upper left). For example, the symbol "busy" means both engaged in action (as in "he is busy") and full of distracting detail (as in "a busy design"). In terms of the meaning triangle: the referents are different, the concepts are different, but the representations are the same. In the extreme case that the concepts constituting the model relations are the same, the model is *generic*, as it models the members of a class (genus) rather than one individual referent (figure 3, upper right). For example, the symbol "queen" is a generic model, as it is the representation part of a multi-referent relation. The symbol "Beatrix, queen of the Netherlands" on the other hand is not. Generic models are efficient to use, as the *same* model can be used to guide interaction with *several* referents (the members of a genus).

In a *multi-representation* situation, several models refer to one referent. In the extreme case that the concepts constituting the model relations are different, the models represent *alternative viewpoints* with respect to the same referent (figure 3, bottom left). For example, my Deux-Chevaux can be modelled by means of the symbol "a classic car" or by means of the symbol "a bunch of rusty metal". In terms of the meaning triangle: the referent is the same, but both the concept and the representation are different. In the extreme case that the concepts constituting the simple model relations cannot be distinguished from each other, the models are different *languages* or *synonyms* that refer to the same concept (figure 3, bottom right). For example, both the symbol "round" and "circular" refer to the same concept: they are synonyms. The symbols "arbre", "tree", Baum" and "boom" all four refer to the same concept, albeit in a different language. In terms of the meaning triangle: the referent is the same, the concept is the same, but the representation is different.

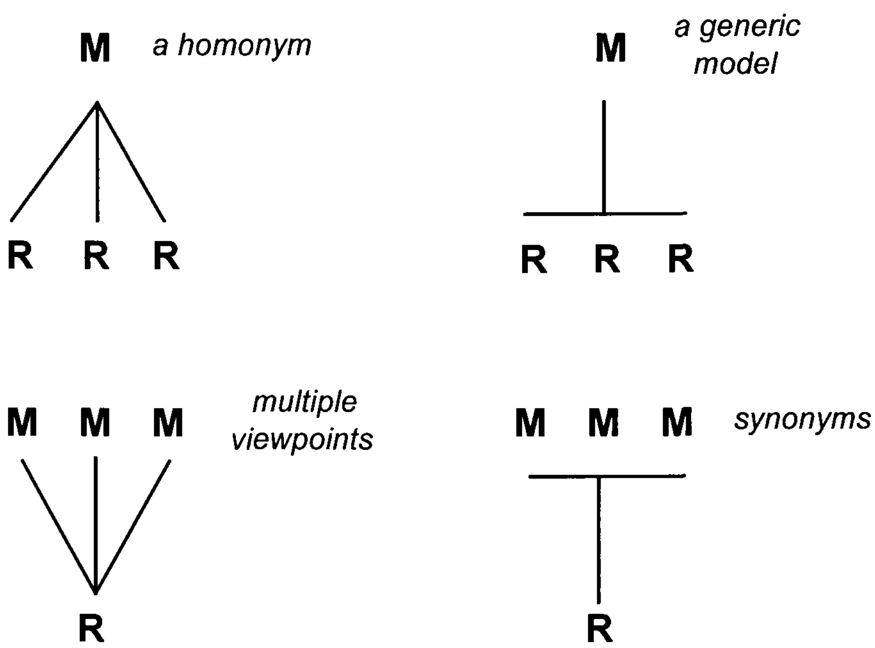


Figure 3: Extremes in multi-referent and multi-representation model relations.

4.3 MODELLING

A model relation relates a referent and a model by means of a concept. A model relation is satisfactory if the modeller is prepared to interact *directly* with the referent, on the basis of the insight offered by the model[32]. In terms of the examples mentioned before, the modeller is ready to build the actual bridge, or to let a real person wear the coat.

According to the theory presented in this paper, modelling processes change a model *relation*, rather than a model (a representation). This is both a subtle and a crucial difference. Although in many situations modelling processes change the model, in many other cases a modelling process results in changes that are not visible in the model (the representation) at all. For example, when a modeller applies a generic model to a new member of the genus, the model does not change, but the referent (and the model relation) does. Another example is the situation in which a modeller decides that a symbol will have a different meaning: again the representation does not change, but the model relation does.

[32] Note that this is exactly in agreement with Chapter 2, where it is stated that perspective construction, and hence a modelling process, stops at the very moment that the model is bidirectionally coupled. "Bidirectionally coupled" implies that the modeller is confident that the model models his environment and that the action will result in some improvement.

In order to obtain a thorough understanding of modelling processes, it is required to include all the ways in which a model relation can be changed. We will use the definitions of model relations as described in section 2 and visualised in figure 2 as a starting point for a description of qualitative modelling processes.

4.3.1 Modelling processes and modelling steps

We call a modelling process the process of converting a model relation into another model relation. A modelling process now can be represented in a general manner as follows:

$$\mathbf{ModProcess}\ (\mathbf{MR}_{\boldsymbol{initial}}) = \mathbf{MR}_{\boldsymbol{final}} \qquad (1)$$

in which $MR_{initial}$ is the original model relation, and MR_{final} is the resulting model relation. Modelling processes are self-contained: a complex modelling process can be interpreted as being composed of several simpler modelling processes. The simplest modelling process is called a *modelling step.*

A typology of modelling steps

Different types of modelling steps can be distinguished. We will develop a typology of primitive modelling steps on the basis of the model relations as presented in section 2, figure 2. This typology is founded upon three different attributes:

1. the type of input model relation;
2. the type of output model relation; and
3. the direction of change.

The type of input model relation refers to whether the modelling step *operates on* a simple, parallel, multi-referent or multi-representation relation.
The type of output model relation refers to whether the modelling step *results in* a simple, parallel, multi-referent or multi-representation relation.

Finally, *the direction of change* refers to the change in complexity of the model relation, i.e. the difference in the number of simple model relations that constitute the input and output relation, respectively. Complexity either *increases* or *decreases*[33]. For example, changing a simple model relation into a parallel relation increases complexity.

By means of applying a combinatorial scheme to the three criteria, we obtain a typology of possible ways to change a model relation. The combinatorial scheme results in 4*4*2=32

[33] The third possibility, *remains equal,* is neglected as this is not possible in one primitive step: it requires, for example, a sequence of extension and restriction, or abstraction and specification.

combinations. A selection of them is presented in table 1 (for an overview of the complete scheme, see Appendix A).

Table 1: A selection of the typology of ways to change a model relation.

INPUT RELATION	OUTPUT RELATION	DIRECTION
Simple	Parallel	Increase
Parallel	Simple	Decrease
Simple	Multi-referent	Increase
Multi-referent	Simple	Decrease
Simple	Multi-representation	Increase
Multi-representation	Simple	Decrease

This typology of 32 ways to change a model relation, however, is purely syntactical: it emphasises morphological, structural changes in a model relation. From a *semantic* point of view, two additional types of change can be distinguished: the *"building blocks"* type and the *"transformation"* type.
The "building blocks" type changes the identity of the model relation by means of either *extending* the original model relation (simple model relations are added), or *restricting* the original model relation (simple model relations are deleted). According to the "building blocks" approach, modelling is a process of playing *Lego* with simple model relations (the building blocks). For example, a model of the computational kernel of a computer programme can be extended with a model of a user interface. The resulting model relation encompasses both the original one (the computational kernel) *plus* an extra one (the user interface).
The "transformation" type preserves the identity of the model relation by means of either *specifying* the original model relation (turning the original model relation into several partial model relations), or *abstracting* the original model relation (turning several partial model relations into one "whole"). The original model relation is *transformed* into a new model relation. The difference between a "transformation" type of change and a "building blocks" type of change is that in a transformation a *conservation principle* applies. For example, a global model of the computational kernel of a computer programme can be specified into a more detailed model of the same kernel. The referent of the resulting model relation and the original one are identical (this is the conservation).

An example will further explain the difference between the "building blocks" type of change and the "transformation" type. Imagine a child modelling a horse out of clay. The child may start with a piece of clay representing a body, and then *add* legs, a tail and a head. The child may be dissatisfied with the head, *delete* the original head and *add* a new one. This would be a "building blocks" approach. The "transformation" approach would be

to use one piece of clay and *remodel* it, until a satisfactory result is obtained. Throughout the modelling process, the clay models the horse as a whole (although the model improves, or at least changes, in time).
The example reveals that in practice both approaches are likely to be used in combination. A head, for example, is being "transformed" out of a certain amount of clay (a transformation) and then added (an addition) to the body.

The semantic dimension is a crucial prerequisite to be able to distinguish *specifications* from *extensions*, and *abstractions* from *restrictions* (see figure 4). This is not possible from a purely structural point of view: both an extension and a specification turn a simple model relation into a complex one, and both an abstraction and a restriction do the inverse.

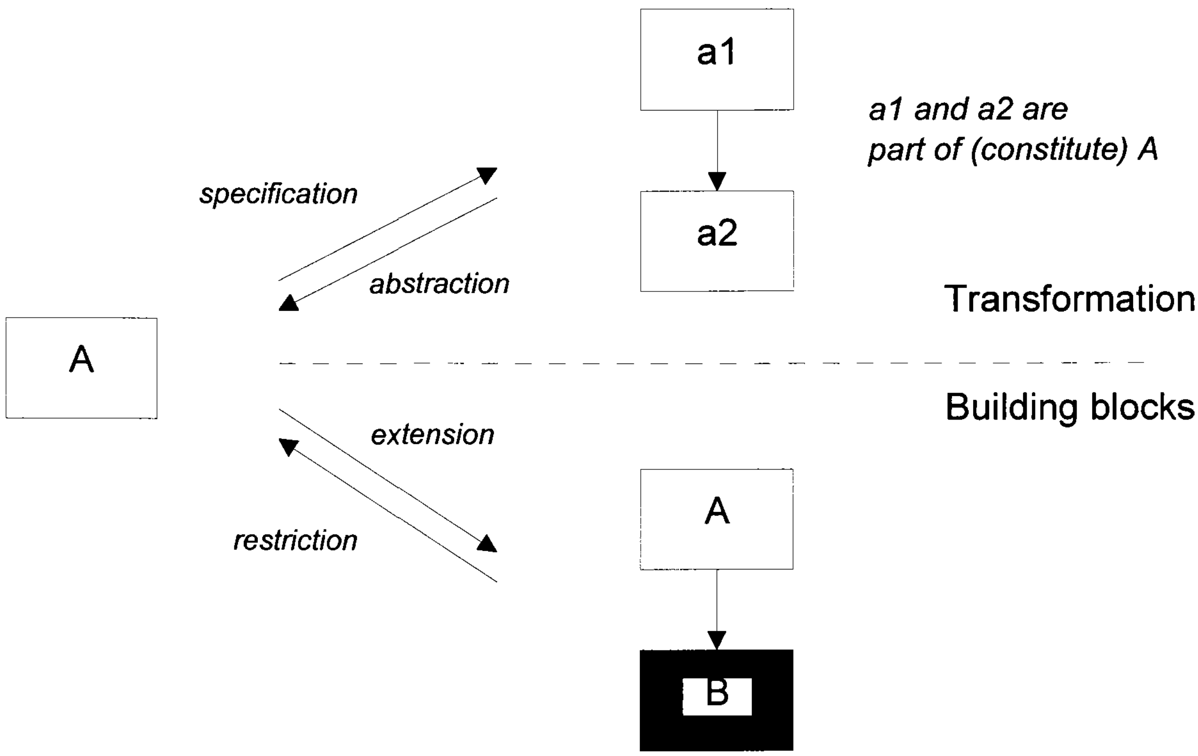


Figure 4: Two modelling processes, structurally the same, semantically different.

Although the resulting models of figure 4 are quite different, without taking into account the semantic dimension (which is indicated by the colour) the two modelling processes cannot be distinguished from each other: their structure is identical. The semantic dimension, therefore, must be included into table 1, which results in a typology of 64 *modelling steps*. A selection of this typology (the same selection as in table 1) is presented in table 2.

Table 2: A typology of primitive modelling steps.

INPUT RELATION	OUTPUT RELATION	DIRECTION	TYPE OF CHANGE	NAME OF MODELLING STEP
Simple	Parallel	Increase	Building blocks	Parallel extension
Parallel	Simple	Decrease	Building blocks	Parallel restriction
Simple	Parallel	Increase	Transformation	Parallel specification
Parallel	Simple	Decrease	Transformation	Parallel abstraction
Simple	Multi-referent	Increase	Building blocks	Multi-referent[34] extension
Multi-referent	Simple	Decrease	Building blocks	Multi-referent restriction
Simple	Multi-referent	Increase	Transformation	Multi-referent specification
Multi-referent	Simple	Decrease	Transformation	Multi-referent abstraction
Simple	Multi-representation	Increase	Building blocks	Multi-representation[35] extension
Multi-representation	Simple	Decrease	Building blocks	Multi-representation restriction
Simple	Multi-representation	Increase	Transformation	Multi-representation specification
Multi-representation	Simple	Decrease	Transformation	Multi-representation abstraction

Table 2 presents only 12 of the 64 possible ways. Only changes that take a simple relation either as input or as output are included. Changes that turn a complex model relation into another complex relation (which covers the remaining 54) are left out. The reason for this

[34] A shorthand name for *multi-referent* is *referent*. For example, a referent extension is a synonym for a multi-referent extension.

[35] A shorthand name for *multi-representation* is *representation*. For example, a representation extension is a synonym for a multi-representation extension.

is that this type of change (from complex to complex) can be realised by means of sequencing members of the twelve types of table 2. Therefore, we call these twelve *primitive*. By means of sequencing, any type of model relation can be converted into any other type of model relation. This issue will be elaborated upon in section 4.4.1 (Non-primitive modelling steps).

In naming the modelling steps, the following convention is used. As was shown in figure 4, a building blocks step that increases complexity is called an *extension*; a building blocks step that decreases complexity is called a *restriction*. A transformation step that increases complexity is called a *specification*; a transformation step that decreases complexity is called an *abstraction*. The *type* of the complex relation (that is either input or output of the modelling step) is added in front. A schematic representation of each of the types of primitive modelling steps is presented in figures 5a-5c. Each of the steps will be explained separately.

In principle, the terms "extension" and "restriction" are slightly misguiding; they hide the fact that an extension *merges* two or more model relations, resulting in a new one, and that a restriction *splits* a model relation, resulting in two or more new model relations. The difference between extensions/restrictions on the one hand and merge/split on the other is subtle, however, and depends on the way in which the modeller sees the modelling process. Consider the analogue with playing *Lego*: when you make a wall higher, you are likely to see this as an *extension* of the wall (the building blocks that you are adding are small). If, on the other hand, you add a roof to a wall, then you are likely to see this as a *merge* operation (as both parts are substantial). The third possible interpretation, that you are adding a wall to a brick, is rather unlikely.

4.3.2 Parallel steps

Parallel steps are visualised in figure 5a[36]. The transition model in the middle makes clear that M1 and M2 *are part of and constitute* M, and R1 and *R2 are part of and constitute* R.

A *parallel extension* is a modelling step in which both the model and the referent are extended, resulting in a new overall model and overall referent, consisting of partial models and referents (note the "is-part-of" relations in figure 5a). A *parallel restriction* is the inverse situation: part of the model is deleted, simultaneously resulting in a restriction of the original referent of the model. The existence of M and R and the overall model relation relating them to one another also disappear.

An example of a parallel extension is the situation in which the symbol "car", referring to a specific car, is extended with the symbol "trailer", resulting in the overall symbol "car with trailer". Removing the symbol "trailer" from "car with trailer" (resulting in "car") would be

[36] Models and referents that were not present in the original relation (at the left side) are printed in bold face.

a parallel restriction. An example of parallel extensions and restrictions from a functional point of view is provided by a range of Swiss pocket knives, from cheap to expensive.

A *parallel specification* is a modelling step in which an overall model-referent pair is divided into several model-referent pairs, in combination constituting the original relation. The original model relation is transformed into several model relations at a more detailed level. For example, the model relation associated with the symbol "bicycle" can be detailed, resulting in model relations associated with the symbols "frame", "wheels", et cetera. A *parallel abstraction* would be the inverse operation.

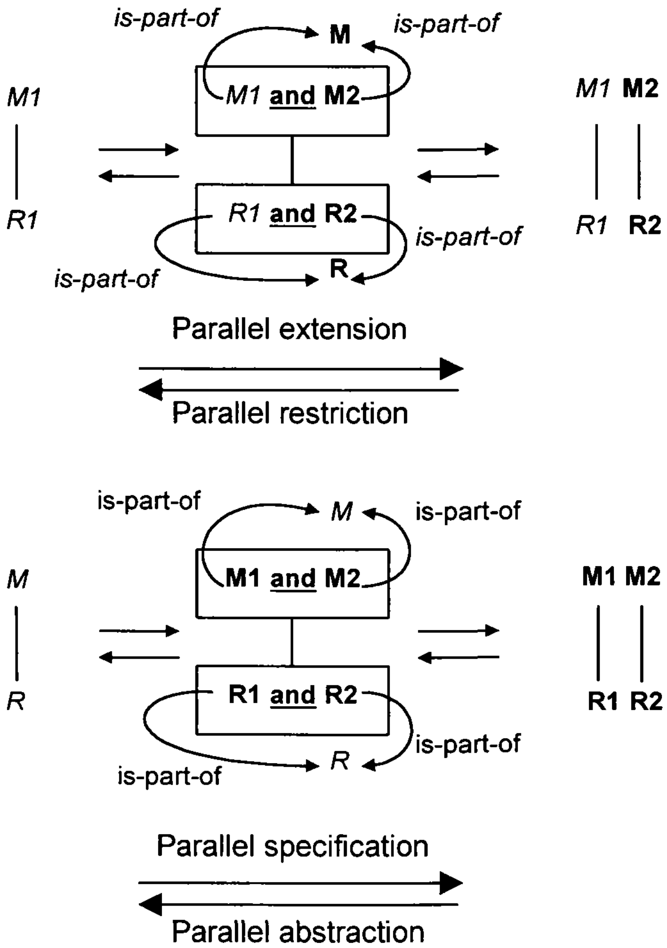


Figure 5a: Parallel modelling steps.

Parallel steps may introduce *emerging and vanishing properties*: properties of "the whole", that are not properties of the parts in isolation, and properties of the parts, that are not properties of the whole (see also Chapter 2). For example, the function of opening a bottle of wine is an extensive quality of the model relation symbolised by "corkscrew". Parts of a corkscrew do not have this quality, although each part has a function. The partial functions are part of (make up) the overall function to open a bottle of wine. Another example is the symbol "car with trailer": the symbol "with" reflects that the overall concept "car with trailer" as a whole is more than the sum of "car" and "trailer": a "car with trailer", for example, enables one to transport large goods over long distances, a feature that is not included in the sum of "car" (which enables to cover long distances) and "trailer" (which enables to transport large goods). It is an emerging property.

4.3.3 Multi-referent steps

Multi-referent steps are presented in figure 5b. In a multi-referent relation, several referents are modelled by one and the same model. Multi-referent steps change the number of referents that are modelled by a model. They are mental steps, and invisible in the model: a change in the model relation is not reflected by a change in the model (note that the "model" parts of the model relations presented in figure 5b do not change). They facilitate, however, a style of modelling that is based upon the construction and subsequent *repeated* use of generic models: a highly efficient and effective approach, provided that the right generic model is available. In practice, this requires that a sufficiently large library of generic models, covering the whole application domain be available.

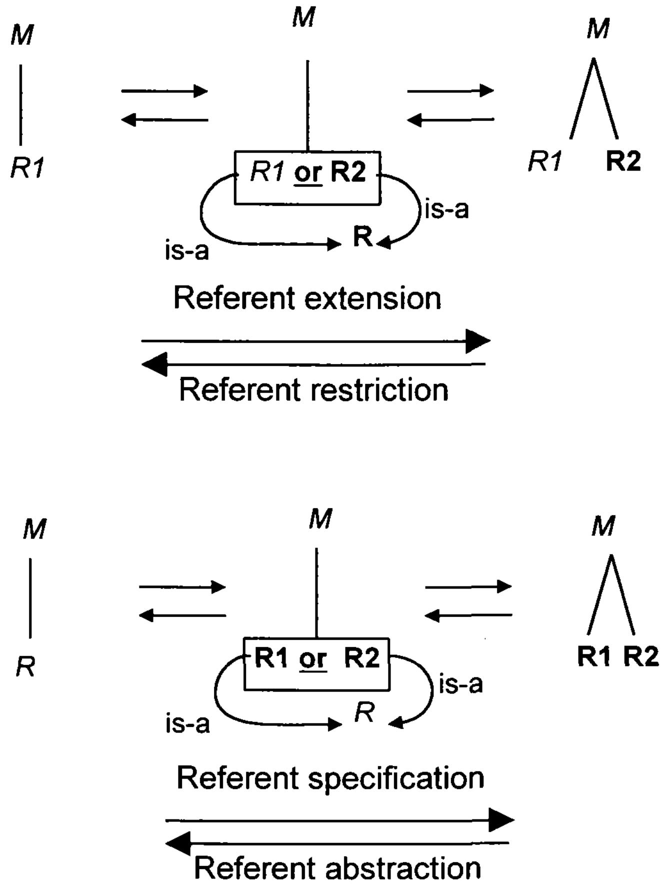


Figure 5b: Referent modelling steps.

A *referent extension* adds a referent to the model relation. A *referent restriction* deletes a referent from a multi-referent relation (for example, a selection process).
An example is crossing a river. Both a water bicycle and a rowing boat are referents (instances) of the generic model (class) "means to cross the river". If the situation would be that both a water bicycle and a rowing boat are available, a referent restriction is required (which implies a selection criterion).

A *referent abstraction* is a modelling step in which several referents are classified into one generic referent. For example, in the initial model relation, the symbol "screw" might refer to 10 different types of screws. After a referent abstraction, the symbol "screw" might refer to large screws and small screws (or any other more abstract typology of screws). A *referent specification* is the inverse operation: the typology of referents becomes more specific.
Another example is the introduction of the ECU (European Currency Unit). At this moment, much discussion is going on in order to determine whether national versions of the ECU are allowed to exhibit a national symbol. This would result in a Dutch version, a

French version, a German version, et cetera. For someone collecting coins, this difference does matter. For someone interested in buying an ice cream on the Costa Brava, this difference does not matter, however, and therefore can be abstracted.

4.3.4 Multi-representation steps

Multi-representation steps are presented in figure 5c. In a multi-representation relation *several* models exist of the *same* referent. Multi-representation steps enable one to explore several viewpoints (or, in case of identical concepts, to use several languages) with respect to one and the same referent. They enable a style of modelling that is based on introducing and discarding alternative viewpoints. In multi-representation steps *the referent* remains the same, whereas in multi-referent steps *the model* remains the same. Remember that a model is used to explore (gain an understanding of) a referent without actually interacting with this referent. Multi-representation steps change the number of alternative explorations, whereas multi-referent steps change the number of referents that are modelled by a model.

A *representation extension* adds a model to a referent. A *representation restriction* deletes a representation of a referent. An example is my Deux-Chevaux. A likely model obviously is "a classic and remarkable car". However, "a bunch of corroding materials" would be an alternative model of the same referent. Note that both viewpoints induce quite different action potentials.

A *representation abstraction* results in a generic representation of several models of one and the same referent. A *representation specification* specifies a model of a specific referent into a number of different alternatives. For example, the descriptions (models) "nice", and "good" might be used to refer to a specific situation. This can be abstracted to the model "positive situation".

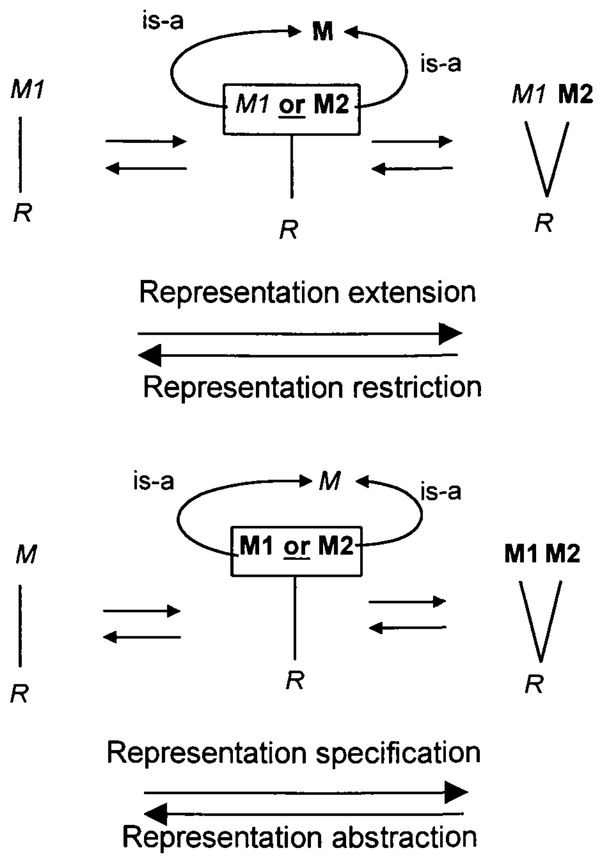


Figure 5c: Representation modelling steps.

4.4 SEQUENCING MODELLING STEPS

The introduction of primitive modelling steps enables one to define a modelling process as a sequence of these steps (see also equation 1). In this section, we will address two issues related to sequencing primitive modelling steps: *non-primitive* modelling processes and *typical* sequences.

4.4.1 Non-primitive modelling processes

When looking at table 2, it seems that certain modelling processes cannot take place. Modelling steps that convert a complex model relation into another complex model relation (for example, extending a parallel relation into a more complex parallel relation, or turning a parallel relation into a multi-representation relation) are not provided. However, by means of sequencing primitive modelling steps it is very well possible to do so. Non-primitive modelling steps are sequences of primitive modelling steps. In order to explain this, figure 6 visualises the twelve primitive steps; non-primitive modelling steps (52 of the 64 possible combinations) are summarised by means of the arrows with a cross.

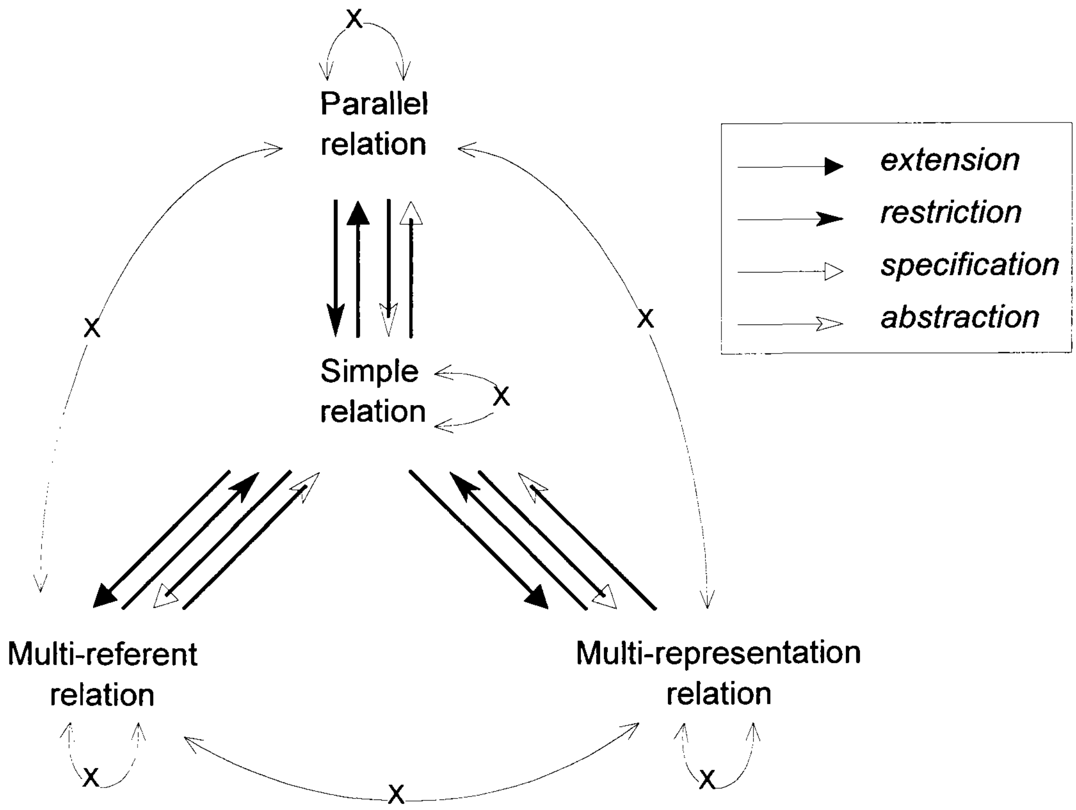


Figure 6: Constraints in sequencing modelling steps (the *tetrahedron* figure).

We will refer to figure 6 as the "tetrahedron figure". It is like a railroad map: any modelling process can be abstracted to a route in this figure. The simple model relation plays a crucial intermediate role: it functions as a bridge (to use an analogue: the simple relation is like a "hub" for an airline company). Some examples will explain this crucial role further.

The first example is about changing the type of a complex model. By means of using a simple relation as an intermediate, it is possible to change the type of a complex model relation. For example, a multi-referent relation can be *abstracted* to a simple model relation, and this simple model relation can be *specified* to a parallel relation. An alternative sequence would be to *restrict* the multi-referent relation to a simple model relation, and then to *extend* this simple model relation to a parallel relation. The example is presented in figure 7a below.

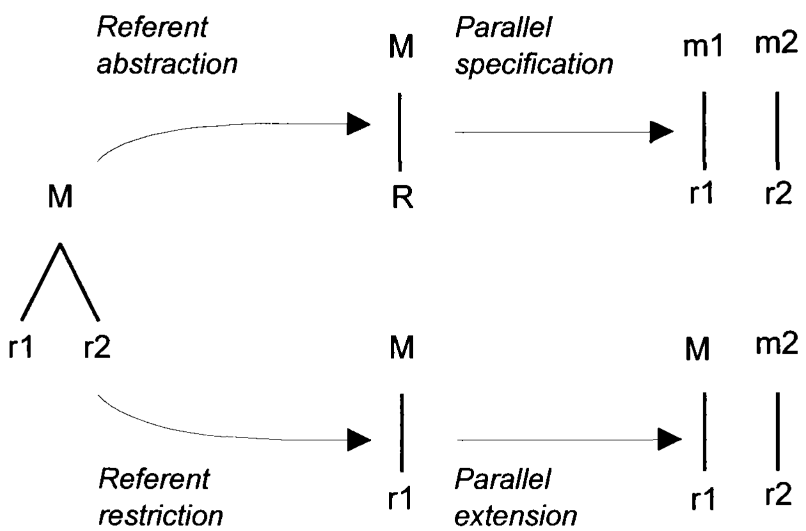


Figure 7a: Sequencing referent and parallel steps via an intermediate simple relation enables one to convert a multi-referent relation into a parallel one.

The next example shows the way in which extending a parallel relation, resulting in a more complex parallel relation, can be understood in terms of a sequence of primitive modelling steps. Consider a referent modelled by means of the symbols “a” and “b” (a parallel model relation). A possible sequence to extend this relation is:

a, b → (parallel abstraction) → x → (parallel extension) → x, c
→ (parallel abstraction) → y

In a different notation:

a and b constitute x
x and c constitute y
This is equivalent to:
a and b and c constitute y

The parallel relation *a, b* is extended to the parallel relation *a, b, c*.

The third example shows the way in which a specification of the simple relation modelled by *y* to three parallel parts can be modelled in terms of primitive steps (this is the inverse of the example above). The sequence is:

y → (parallel specification) → x, c → (parallel restriction) → x
→ (parallel specification) → a, b

In a different notation:
x and c constitute y
a and b constitute x
Which is equivalent to:
a and b and c constitute y

As a real-world example, consider a simple model relation of which the model part is the symbol "Tree", referring to a tree. This simple model relation can be specified, resulting in a parallel relation modelled by "Trunk" and "Crown". A crown in turn is known to consist of branches and leaves. The example is presented in figure 7b.

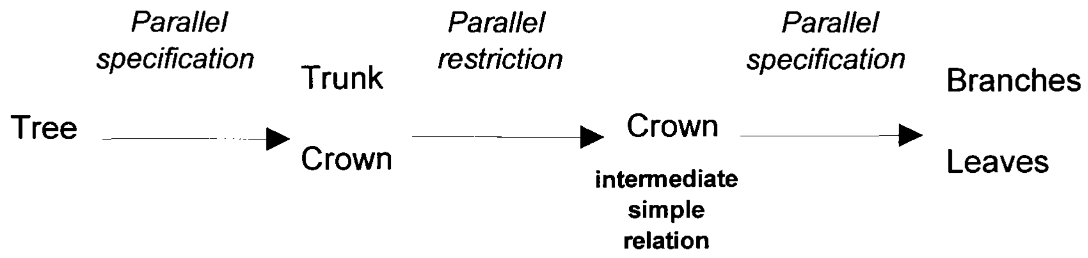


Figure 7b: Specifying a complex relation via an intermediate simple relation.

In a different notation:

Trunk and Crown constitute Tree
Branches and Leaves constitute Crown

These relations in combination enable one to infer that:

Trunk and Branches and Leaves constitute Tree

Now, as a last example, consider the situation in which you become aware that the model you are modelling is based upon totally wrong assumptions. It is not allowed to discard the model relation (resulting in "nothing") and subsequently start a new modelling process, as the twelve primitive steps do not provide a route from something to nothing. After a more careful analysis, it becomes clear that this is not what actually happens within a line of reasoning. What happens is that, rather than abstaining from a model relation and subsequently starting a new modelling process, a new alternative develops *next to* the model relation you are working with. The original relation is abstracted, resulting in a simple model relation, which is extended to a multi-representation relation. At the very moment you decide to switch, the multi-representation relation is restricted to a simple one (the original relation is discarded), and the modelling process continues. A "one alternative" situation develops into a "two alternatives" situation, after which the original alternative is discarded. Again, the simple model relation performs a crucial role as a bridge.

Consider, for example, the situation in which you would like to drive a cabriolet (the referent), and you are thinking about purchasing a Porsche to do so (the point of view). On the way to the Porsche dealer, a Deux-Chevaux overtakes you: an alternative emerges. The model relation becomes multi-representation. You make the decision between the Porsche and the Deux-Chevaux and you continue on your way to the Citroën dealer. The example is presented in figure 7c.

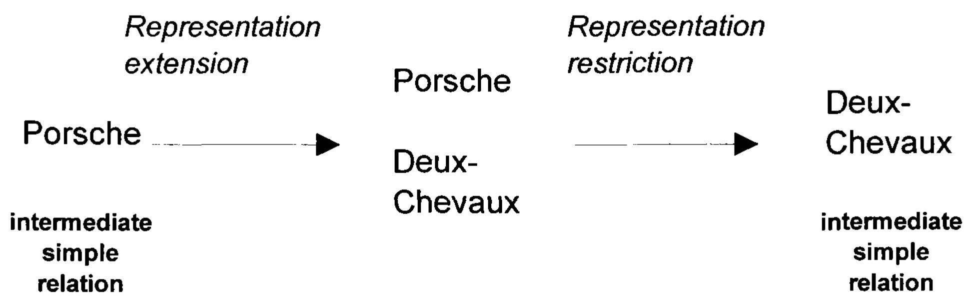


Figure 7c: The Deux-Chevaux beats the Porsche.

The examples above show that, by means of sequencing primitive modelling steps, complex modelling processes can be performed, turning any model relation into any other model relation. At each moment between two modelling steps, the model relation must be simple, parallel, multi-referent or multi-representation. The simple model relation performs a crucial intermediate role.

4.4.2 Typical sequences: modelling strategies

Complex modelling processes can be analysed or designed in terms of the types of primitive modelling steps they consist of. This is important, as it allows for the distinction of *modelling strategies*: characteristic (typical, recurrent) sequences of types of primitive modelling steps that can be used to attain a specific goal (for example, "obtain a bird's eye view"). Several strategies in combination may form a more complex modelling strategy.

Modelling strategies offer a means to think about and discuss modelling processes in a less detailed manner than modelling steps do. Many different strategies can be distinguished. In each strategy, primitive modelling steps are "mixed" in different formulas. In terms of the tetrahedron figure (the railroad plan, figure 6), strategies form different but recurrent travelling schemes. An overview of "families" of strategies is presented in table 3. Three "families" are distinguished, in line with table 2. They are explained below.

Table 3: Families of modelling strategies.

Family 1: the structural family
- *parallel strategies*
- *multi-referent strategies*
- *multi-representation strategies*

Family 2: the semantic family
- *building blocks strategies*
- *transformation strategies*

Family 3: the complexity family
- *expansion strategies*
- *reduction strategies*
- *balancing strategies*

4.4.2.1 Family 1: the structural family

The structural family emphasises the way in which the structure of a model relation changes. This family can be divided into three subtypes: parallel strategies, multi-referent strategies and multi-representation strategies. They are distinguished in figure 8.

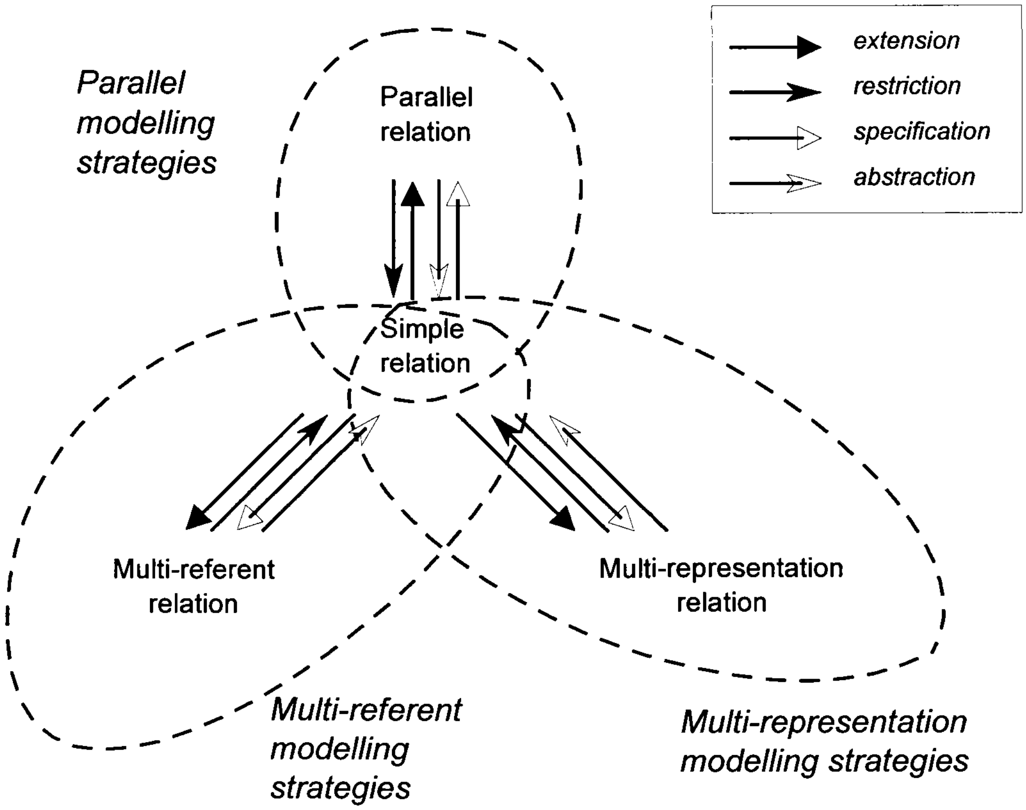


Figure 8: Strategies of the structural family.

Parallel modelling strategies maintain a (multiple) "one model - one referent" relation throughout the modelling process: the number of viewpoints is one, and the number of referents is one. The scope can be extended or restricted, and the level of detail can be abstracted or specified. Some parallel strategies are represented in figure 9 (in italics), and elaborated upon below. Note that, in contrast with the tetrahedron figure, in figure 9 arrows represent intermediate model relations, and modelling steps are represented in text.

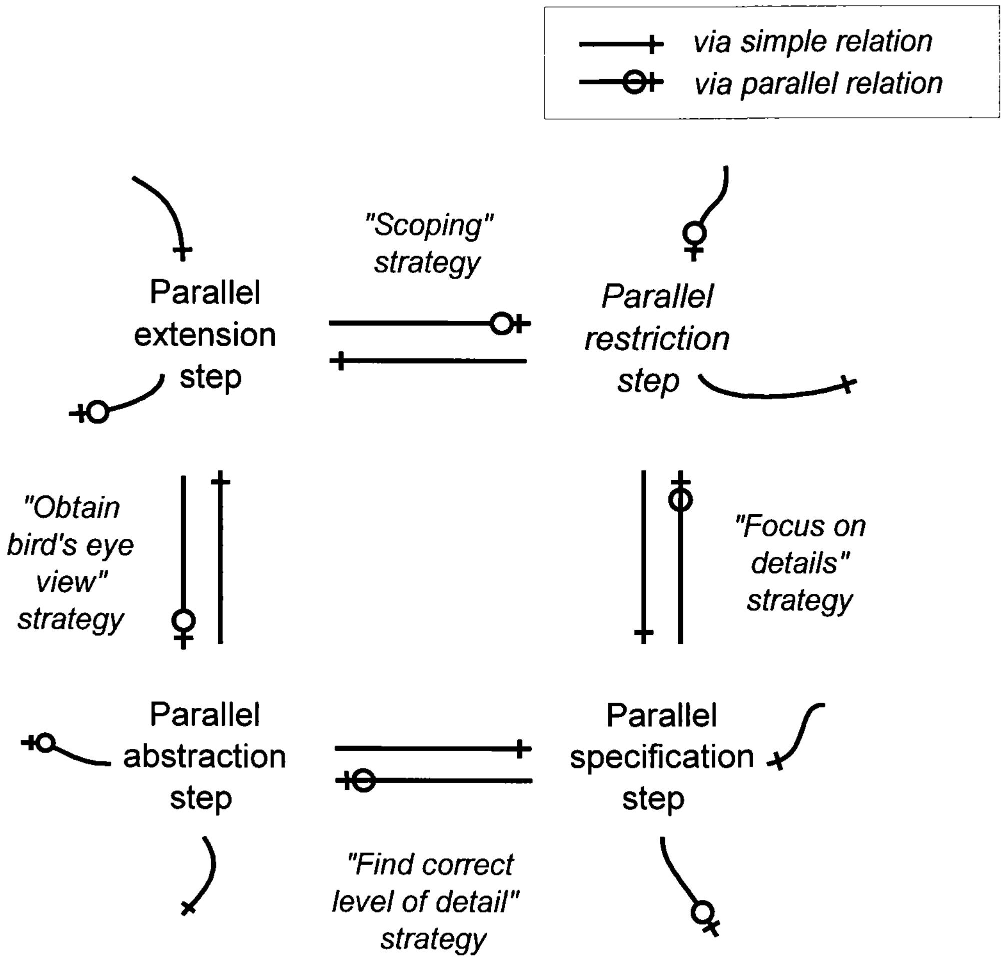


Figure 9: Parallel modelling processes.

Parallel restriction steps and parallel extension steps in combination result in a "scoping" strategy: the "coverage" of a model changes.
The "find correct level of detail" strategy alternates parallel abstraction and parallel specification steps, enabling one to explore several levels of detail (changing the granularity of the model relation).
Parallel extension steps, alternated with parallel abstraction steps, offer the opportunity to extend the scope at the expense of the level of detail. This results in a bird's eye view. Therefore, this strategy is called the "bird's eye" strategy.
The inverse strategy of the "bird's eye" strategy is the "focus on details" strategy: by means of a combination of parallel restrictions and parallel specifications, part of a more complex referent is modelled into more detail. In plain English: the level of detail is increased at the expense of scope.

Referent modelling strategies change the number of referents of a model (the model does not change). The strategy is "model-driven" (theoretical) rather than "observation-driven" (empirical): the model is used to recognise a referent.
The model used in a referent strategy, in principle, is a *generic* model: it models the elements of a set (a "genus") of referents. A referent strategy leaves the model unchanged: it is a rather invisible strategy. Some referent strategies are presented in figure 10 and explained below.

A "change referents" strategy is a sequence of referent extensions and referent restrictions. The set of referents modelled by the generic model is extended and restricted[37]. Examples are recognising alternative application domains of a model (extension) and selecting between them (restriction).
A "find correct level of generalisation" strategy is a sequence of referent abstraction and referent specification steps. For example, when you like to revise Deux-Chevaux, you are likely to be knowledgeable about the different types of Deux-Chevaux that exist. When you are talking about a complex revision of a Deux-Chevaux with another hobbyist, and you did not explicitly mention the type (you are talking about “my Deux-Chevaux”), peculiarities of the revision process will help your partner in conversation to infer the specific type you are referring to (by means of a referent specification step). In case she makes a wrong inference, a referent abstraction step is required to return, and another specification may be tried.

A "referent generalisation" strategy is the process of adding a referent to a model, and subsequently recognising that both referents are instantiations of a more general type of referent. For example, a Deux-Chevaux and a DS are both referents of the generic model "automobile", and can be transformed into the more general referent "Citroën": the Deux-Chevaux is a Citroën, the DS is a Citroën, and a Citroën is an automobile.
A "referent specialisation" strategy is the process of recognising that one of the referents of a multi-referent relation can be specialised into a set of referents. For example, two referents of the generic model "automobile" are "Citroën" and "Volvo". This relation can be restricted to the relation "automobile" referring to "Citroën", and subsequently the referent "Citroën" can be specified into "Deux-Chevaux" and "DS". Figure 11 presents some of the examples in a graphical form.

[37] It is interesting to note that the use of a *quantitative* model, for example, the formula F=m.a, in a specific situation amounts to applying a referent restriction strategy to the generic model F=m.a. From this point of view, quantitative models are a subset of qualitative models: they are generic models with an extremely large number of referents.

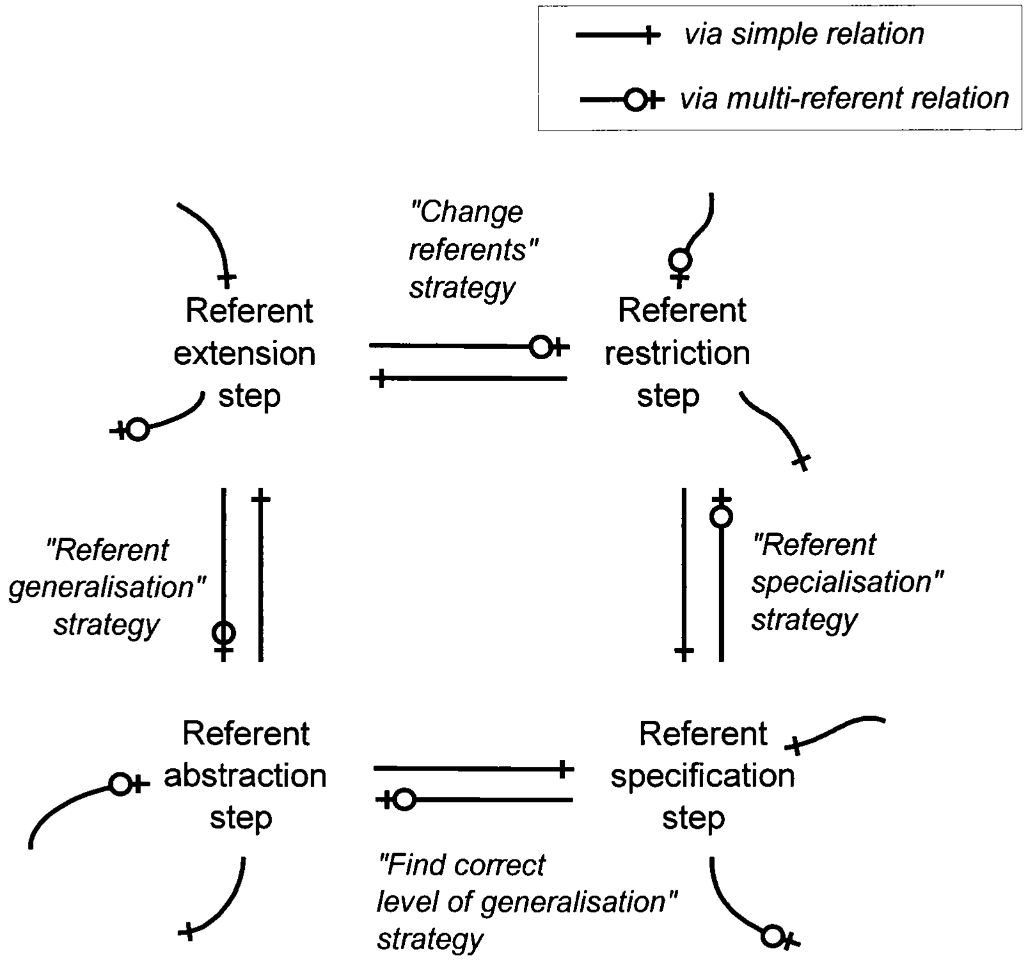


Figure 10: Referent modelling strategies.

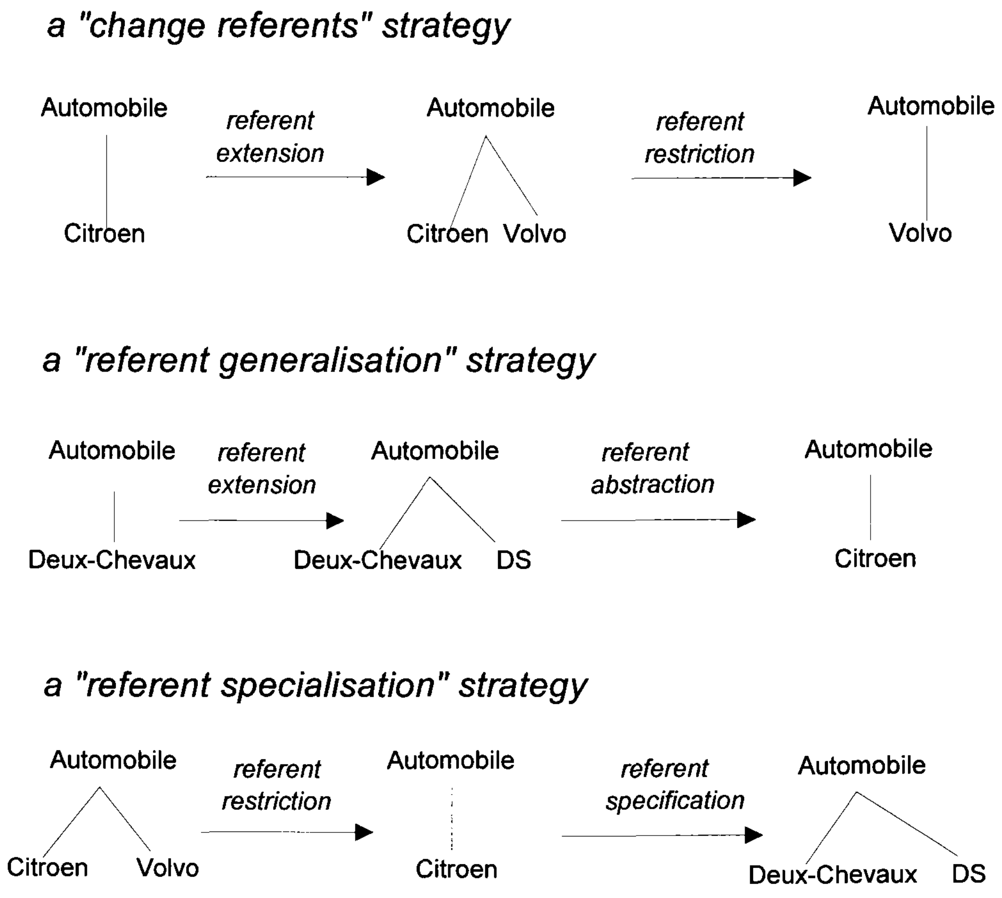


Figure 11: Examples of applying a referent strategy.

Representation strategies change the number of representations (viewpoints or languages) to refer to a specific referent. They are not explained any further.

4.4.2.2 Family 2: the semantic family

The semantic family emphasises the way in which the *meaning* of a model relation changes, rather than its *structure*. This family consists of two main types of strategies: building blocks strategies and transformation strategies.

A **building blocks strategy** is a sequence in which extension steps are alternated with restriction steps. Building blocks strategies are routes in the tetrahedron figure that follow the arrows with the black heads (i.e. extensions and restrictions, see figure 12).

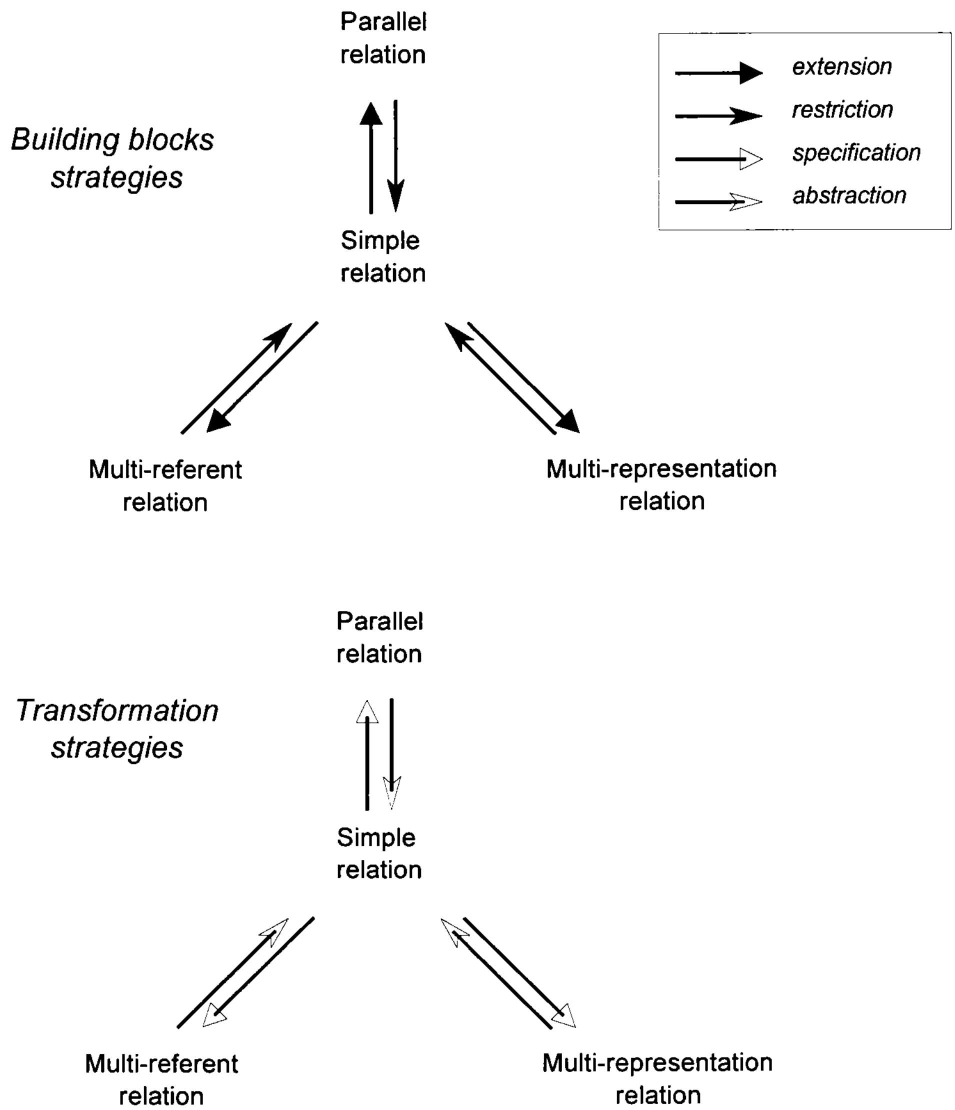


Figure 12: Building blocks and transformation strategies.

A **transformation strategy** is a sequence of abstraction and specification steps. Basically, the same model relation is being described on different levels of abstraction, until a satisfactory model results. Transformation strategies are routes in the tetrahedron figure that follow the arrows with the white heads (i.e. specifications and abstractions, see figure 12).

4.4.2.3 Family 3: the complexity family

Strategies of the complexity family emphasise the way in which the complexity of a model relation changes. In order to be able to distinguish strategies of the complexity family, a procedure is required to "measure", or at least order the complexity of model relations. A convenient feature of the definition of a model relation, as presented in figure 2, is that it provides the elements for such a procedure. The mechanism is based on the notion of *structural complexity* (**s**) of a model relation: ***s** equals the number of simple model relations that constitute a complex model relation.*

The simple model relations, constituting a complex model relation, may in *their* turn be complex at a lower systemic level. For example, a wheel is a part of a bicycle, but a wheel consists in turn of several elements (a lower systemic level in terms of composition). A world journey may be modelled in terms of partial journeys, which in their turn may be modelled in terms of even shorter journeys (a lower systemic level in terms of time or episode). An intentional activity may be modelled in terms of other intentional activities, which in turn may be modelled in terms of other intentional activities. In order to obtain an accurate estimate of the complexity of a model relation, *all* systemic levels of relevance must be taken into account. At a certain point, the model relations are considered to be atomic (simple and not complex at a lower systemic level): further specifying their structural complexity is considered to be useless. For example, for a bicycle mechanic it makes sense to distinguish a tire, a pedal and so on of a bicycle. It does not make sense for him, however, to distinguish separate atoms of a bicycle (the reason is pragmatic: he cannot fix atoms, but he can fix a pedal.).

The sum of simple relations in a complex model relation on all systemic levels, is called the *overall complexity* (**o**) of a model relation[38]. Note that for models that consist of only one systemic level, **o** reduces to **s**.

When applying a primitive modelling step to a model relation, this results in a new model relation with a different overall complexity. In line with this, three types of complexity strategies can be distinguished: **expansion strategies** (they increase **o**), **reduction strategies** (they decrease **o**) and **balancing strategies** (they keep **o** at approximately the same value).

Extensions by definition *add* model relations: they increase **s**, and therefore **o**, and as such are by definition part of expansion strategies. Restrictions by definition delete model relations: they decrease **s**, and therefore **o**, and as such are part of reduction strategies.

For abstractions and specifications the distinction is not so straightforward. As is the case with extensions and restrictions, an abstraction reduces **s**, and a specification increases **s**. It would, therefore, seem correct to say that abstractions reduce **o** and specifications increase **o**. However, it must be kept in mind that **o** is defined as the number of simple model relations on **all** systemic levels. And it is possible indeed to use models of the same referent, but on different levels of abstraction (i.e. multi-level models).

For example, it is possible to abstract a complex model (a bottom-up approach), and to *throw* the original, more specific model *away*. The new level of abstraction is the new atomic level. In this case, abstraction results in a *reduction* of **o**. On the other hand, it is

[38] Note that this notion is very similar to the structure of a Knowledge Distribution, as presented in Chapter 3.

possible to abstract a complex model and *preserve* the original, more specific model as a refinement at a lower systemic level. In this case, the original level of abstraction remains the atomic level, and **o** increases! A new, more abstract systemic level is added to the model. The same argument holds for top-down approaches, in which new systemic levels are added at lower systemic levels.
As a result, in situations where the number of systemic levels is kept at the same value (it is likely that only one systemic level is distinguished), reduction strategies consist of sequences of restrictions and abstractions, and expansion strategies consist of sequences of extensions and specifications. In multi-level modelling situations, restrictions are part of reduction strategies, and extensions are part of expansion strategies. Abstractions and specifications may be part of both, depending on whether the original level of abstraction is discarded or maintained as part of the model.

Expansion strategies at one systemic level are routes in the tetrahedron figure that contain many arrows with a "triangle head" (see figure 13): as a result, **s** increases. Reduction strategies are routes in the tetrahedron figure that consist of many arrows with a "four-cornered head" (figure 13); as a result, **s** decreases.

The third complexity strategy is the balancing strategy. This strategy is founded upon the observation that only a certain amount of complexity can be understood (a "cognitive threshold" exists). When this amount is reached, a model must be made more simple at certain places in order to be able to make it more complex at other places (hence *balancing strategies*).
A *balancing* strategy maintains **o** at (near) a certain level. For example, extensions at a high systemic level can be alternated with abstractions at a low systemic level. This results in a model that covers a larger scope, though the model relations that are considered to be atomic are coarser. This, for example, is (should be) the difference between the point of view of a member from the advisory board of a multinational and a specialist in repairing television sets.

For models at only one level of abstraction, balancing strategies are routes in which the number of "triangle head" arrows and "four-cornered head" arrows is approximately equal. For models at several systemic levels, balancing strategies are mixtures of expansion and reduction strategies. Note that, for example, a "bird's eye view" strategy (a parallel strategy) is also a simple form of a balancing strategy. This shows that strategies may belong to different families. In contrast with the other families a balancing strategy, in its most complex form, may use *all* the twelve steps that are available.

In summary: a **balancing strategy** is a sequence in which overall complexity is kept (close) to a certain value.

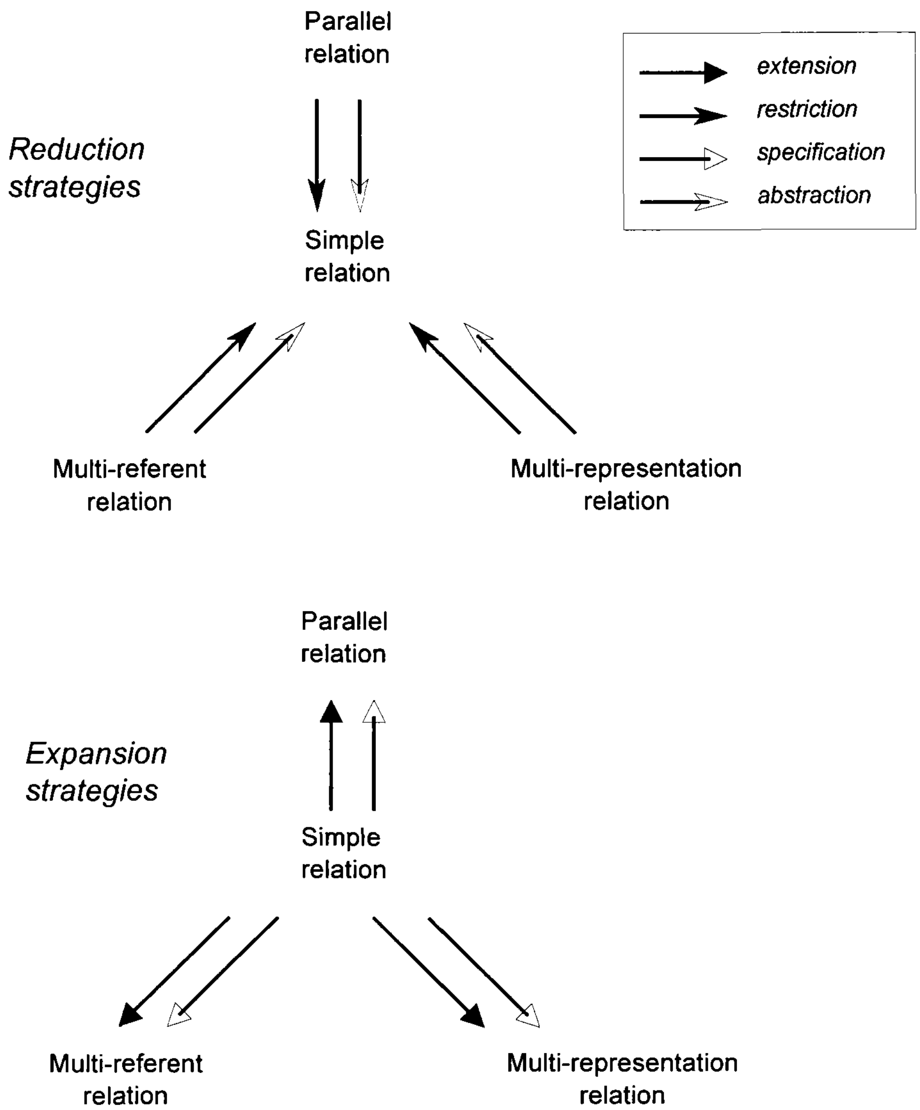


Figure 13: Reduction and expansion strategies with respect to **s** (models with a fixed number of systemic levels).

Balancing strategies have a strong intuitive appeal: at a certain point increasing **o** must be compensated for by decreasing **o** somewhere else. A balancing strategy prevents one from exceeding some "cognitive threshold". The maxims "you can know a lot about a little" and "all you can say about almost everything is next to nothing" express this notion.
The types of modelling steps seem to be "interchangeable" with respect to the cognitive threshold to a certain extent. For example, for models at one systemic level abstraction facilitates specification, but also extension; extension above the threshold requires first of all either abstraction or restriction. Especially complex modelling processes are likely to belong to the family of balancing strategies.

People prefer simple models, as they are easier to comprehend. An interpretation of "simple models" in terms of this paper would be: models of which the model relation has a low overall structural complexity (**o**). However, as Einstein observed:

"Theories should be as simple as possible. But not any simpler."

We suggest that the same holds true for models. Complex situations are complex because they require many different concepts (model relations) to be taken into account: overall structural complexity is high (near the cognitive threshold). Making a model simpler by means of using a reduction strategy may result in a model that is easier to understand, but does not facilitate purposeful action anymore[39]. To use an analogue: playing the piano surely would be very simple if only one key were left on it. One might wonder, however, whether people would enjoy playing this piano, or listen to the music.

When the required overall complexity of a model relation exceeds the cognitive threshold of individuals, co-operation is the only way to proceed. In earlier work (Chapter 3), we identified this type of situation as "D-type". D-type environmental problems are *examples par excellence* of the necessity to co-operate. The overall model relation must be shared by several individuals and groups.

Here a link with the KDS framework becomes manifest. KDS enables one to model bodies of knowledge, being systemic constructions of perspectives, in terms of distribution characteristics. However, when interpreting a large body of knowledge (for example, the knowledge required to develop, produce and distribute television sets), the complexity of this body of knowledge is likely to exceed the cognitive threshold. It may be possible to obtain a global overview of the knowledge of concern, but it is impossible for one person to fully comprehend it. For example, a chief executive officer of a multinational in sound and vision equipment is not likely to actually know how to produce a television set in all its details. The only way out is by means of the balancing concept: detailed knowledge is not known, therefore overview can be obtained (note that this is a bird's eye view).

The point where making use of the balancing concept becomes unavoidable can intuitively be visualised in KDS. As before (see Chapter 3), we will use a mathematical derivation. However, as before, the results will only be used qualitatively.
Let us assume that the maximum amount of knowledge that can be understood by one person, $\mathbf{c}_{max}$, equals the amount of knowledge (in terms of perspectives on all systemic levels) possessed by an extremely gifted human expert: this corresponds with an extreme position in the $C_{personal}$ type region of KDS ($\mathbf{a} = 1$, $\mathbf{d} = 1$, $\mathbf{c} = \mathbf{c}_{max}$).

[39] Nonetheless, this is a very popular strategy. Therefore, we reserve the name "one button strategy" for using a reduction strategy in a complex situation that requires distinction of far more "buttons" to manage it properly.

An iso-plane in KDS (see Chapter 3) connects the points that refer to a body of knowledge consisting of the same amount of perspectives (**c.a.d** = constant). It is impossible to possess more knowledge than an extremely gifted human expert. Therefore, when being confronted with a body of knowledge (corresponding with a part of society), it is impossible to know more than $\mathbf{c}_{max}$ perspectives. For example, when interpreting the body of knowledge of several experts in combination, only an overview can be obtained. If the number of experts increases, this overview necessarily becomes more superficial. This nicely corresponds with the curvature of the iso-plane, going through the position in KDS where the extremely gifted expert resides:

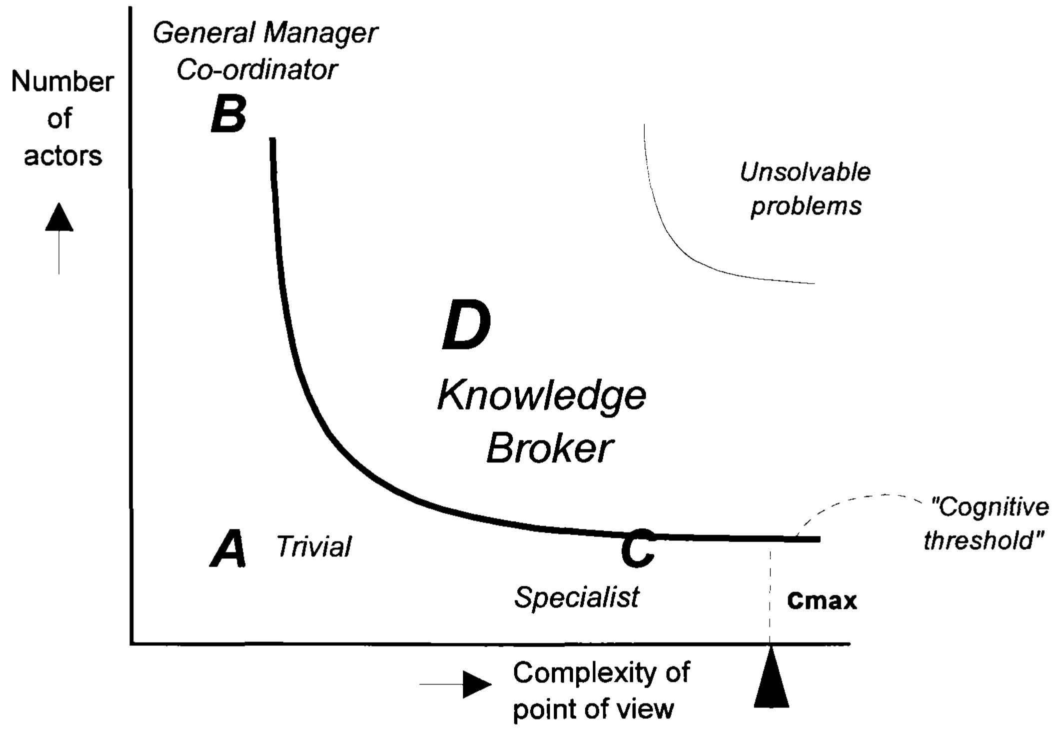
$$\mathbf{c.a.d} = \mathbf{c}_{\mathbf{max}} \qquad \mathbf{(2)}$$

However, when interpreting a body of knowledge with an adherence greater than 1, the task to obtain an overview becomes more easy. The knowledge of several individuals belonging to a social group (constituting an actor) is assumed to be the same[40], and therefore can be "copied". As a result, this does not occupy a great part of the cognitive capacity of the interpreter (limited by the cognitive threshold $\mathbf{c}_{max}$). Therefore, we will neglect this contribution. As a result, bodies of knowledge that in principle can be understood completely by an interpreter are positioned in the following region of KDS:

$$\mathbf{c.d} < \mathbf{c}_{\mathbf{max}} \qquad \mathbf{(3)}$$

A two-dimensional cross-section of the cognitive threshold is presented in figure 14. This makes clear that in the D-type area in KDS, balancing strategies are a necessity.

[40] It is an assumption, though!



The grey area is where an interpreter of this body of knowledge must make a trade-off between level of detail and level of overview. This corresponds with the D-type area and with the realm of the knowledge broker.

Figure 14: The cognitive threshold.
The cognitive threshold is an intuitive concept that separates distributed bodies of knowledge which, *in principle,* can be understood completely from bodies of knowledge which can only be understood by means of a *balancing mechanism* (level of detail is traded off for overview). Its equation is $\mathbf{c.d} < \mathbf{c}_{max}$, which implies that the plane is at the same position for any value of **a**.

4.5 DISCUSSION AND CONCLUSIONS

A model enables one to explore (gain an understanding of) a referent, without directly interacting with this referent. The modelling process stops when the modeller is satisfied with the quality of the model relation: he/she is ready to interact *directly* with the referent, on the basis of the insight offered by the model.

In this chapter, a qualitative modelling process is described as a process of changing a *model relation* (rather than a model), by means of small steps, until the modeller is satisfied. Model relations are either simple, parallel, multi-referent or multi-representation. Twelve different types of primitive modelling steps enable a modeller to turn any model relation into any other relation. They offer the possibility to cover several referents with one model (a generic model), to use different representations of the same referent (a multiple viewpoint), to change the scope and level of detail, or any combination of these. Modelling strategies are typical sequences of modelling steps.
In order to substantiate the claims that the twelve modelling steps are primitive and complete, first, we will identify the degrees of freedom of a model relation. Second, we will a) verify whether each type of modelling step matches with a degree of freedom, and b) check that the number of degrees of freedom equals the number of types of modelling steps. If the first test succeeds, the steps are primitive, and if the second test succeeds, the typology of twelve steps is complete. This approach substantiates that the modelling steps are complete and primitive with respect to model relations as defined in this Chapter (see figure 2).

The degrees of freedom can be derived from figure 2: there are three structurally different dimensions (parallel, multi-referent and multi-representation), two complexity dimensions (increase and decrease), and two semantically different ways to change a model relation ("building blocks" type and "transformation" type). In combination, this results in twelve degrees of freedom. Indeed, each of them is covered by a modelling step (see table 2 and the tetrahedron figure), and there are twelve of them.

A related question that might be posed is: Why use the set of primitive steps defined in this chapter? Why not define another set? Again, the answer to this question is the definition of model relations used in this Chapter (see figure 2). The twelve modelling steps are a logical consequence: they are derived from the types of model relations, rather than constructed.

In the introductory section of this chapter, we mentioned three important reasons for developing a theory of qualitative modelling processes:

1. such a theory provides deeper insight into qualitative modelling processes in general;
2. such a theory would enable one to understand and compare existing qualitative modelling paradigms in terms of a generic conceptual vocabulary; and
3. such a theory guides the design of new qualitative modelling paradigms.

At this point, at least in our opinion, deeper insight into qualitative modelling processes in general is achieved (the first reason). The notions of model relations, modelling steps and modelling strategies offer a conceptual vocabulary to analyse, synthesise and discuss qualitative modelling processes.

With respect to the second reason, it is striking that modelling approaches like Yourdon's DFDs and the IDEF paradigm can be described in terms of our theory: expanding a "bubble" in a DFD would be a parallel specification; extending an IDEF model with an extra rectangle would be a parallel extension, et cetera. An even more striking example is the ontological modelling approach as presented by Fridsma [Fridsma, Gennari and Musen (1997)], in which our notion of parallel modelling steps can be recognised completely, albeit in other terms. In addition, they present a rudimentary version of our notion of modelling strategies (a replacement they interpret as a deletion followed by an addition, which in principle is a very simple example of sequencing modelling steps, resulting in a parallel building blocks strategy). These examples show that it is possible to understand (parts of) several qualitative modelling paradigms in terms of the theory presented in this chapter. The notions of generic models (multi-referent models) and multiple representations, however, generally are given little attention within these paradigms.

The third (and main) reason for developing the theory of qualitative modelling processes was that it provides *guidelines* for designing new qualitative modelling paradigms. These guidelines are summarised below in four steps:

> **Step 1:** *Specify the qualities (concepts and relations) that you want to be able to refer to (that you consider to be relevant to distinguish in your referents).*
>
> **Step 2:** *Select symbols (a language) to refer to these qualities.*
>
> **Step 3:** *Design ways to refer to complex qualities (parallel, multi-referent and multi-representation model relations).*
>
> **Step 4:** *Design the twelve modelling steps in terms of the modelling language.*

A qualitative modelling approach, designed in confirmity with these four steps, enables one to change both the level of detail and the scope of a model, to use multiple representations of one and the same referent, and to construct and use generic models. On top of this, the language will support the use of all the modelling strategies presented earlier, from simple to highly complex.

At several places we used the *Lego* analogon. With hindsight it is easy to understand why *Lego* is such a popular toy: *Lego* enables one to use simple, parallel, multi-referent and multi-representation relations, and allows for performing all twelve types of primitive modelling steps. For example, a *Lego* block can be added to a model (a parallel extension); deleted from a model (a parallel restriction); several blocks can be interpreted as a larger "whole" (a parallel abstraction); and a "whole" can be interpreted as a couple of smaller wholes (a parallel specification). A *Lego* model can refer to a genus (for example, a model of "a dog"). Several models can be constructed that refer to the same referent (multiple representations). In summary: *Lego* is a very flexible qualitative modelling tool. *Lego* is limited, however, in the type of referent it enables one to model: typically, these are relatively static, physical structures. The qualities of interest are morphological. This dissertation is not about explaining the success of *Lego* but about providing model-based support for D-type problem solving. A modelling language that supports D-type problem solving should allow for representing and adapting models of D-type perspectives, which implies D-type "as is" and "to be" situations as well as D-type scripts (see Chapters 2 and 3). In Chapter 5, the *Trinity* modelling language will be presented: a modelling language especially designed to support D-type problem solving. The *Trinity* modelling language will be designed completely in compliance with the theory as presented in this chapter.

PART IV: METHODS

CHAPTER 5

Trinity: MODELLING METHODS TO SUPPORT MULTI-ACTOR PROBLEM SOLVING

"What I have been talking about is knowledge. Knowledge, perhaps, is not a good word for this. Perhaps one would rather say my "image" of the world... It is my image that largely governs my behavior." [Kenneth E. Boulding (1956)]

5.1 INTRODUCTION

So far in this dissertation, a philosophical and theoretical foundation has been laid for the *methods* layer of the *Trinity* methodology. In this chapter, these methods will be worked out.

In general, the *methods* layer of a methodology encompasses generic "prescriptions" of what can be done in order to successfully finish an application of this methodology in specific situations. The methods layer makes the philosophy and the theory layer operational: it provides the conceptual toolbox of the methodology (see also the introductory chapter of this dissertation).

For *Trinity*, this means that modelling methods will be presented that support D-type problem solving. First, the essential elements of the philosophical and theoretical basis of *Trinity* are recalled (section 5.2). Second, in section 5.3, the global contours of the *Trinity* methods layer will be presented. In section 5.4, the *Trinity* modelling language, as a more refined account of *Trinity* methods, will be described in detail. In section 5.5, several *Trinity* modelling strategies will be discussed. In section 5.6, practical guidelines will be presented. In section 5.7, several modes of using *Trinity* are distinguished. The chapter ends with a discussion and conclusions (section 5.8).

5.2 PHILOSOPHICAL AND THEORETICAL BASIS OF *TRINITY*

In this section, the links between the Philosophical background and the Theory parts of this dissertation, and the *Trinity* modelling methods (to be presented in this chapter), will be illuminated. In order to do so, the essence of each of the philosophical and theoretical chapters so far will be reiterated.

Chapter 2: model-based support for problem solving

In Chapter 2, the Philosophical background part, the notions of *problem solving* and *model-based support* were investigated in depth. The central concept in Chapter 2 was the **perspective**: a body of knowledge that motivates and guides an intentional action, and that consists of three parts ("as is", script and "to be"). Problem solving was defined as perspective construction. A problem manifests itself (or rather: emerges) as a mismatch between the environment and the intentions of an actor: the problem owner. To be more precise (see Chapter 2):

> *Problem solving is the attempt of an actor (a problem owner, possibly a group of individuals) to re-establish correspondence between its environment and its intentions. This attempt manifests itself as a process of developing an incomplete perspective into a perspective that models both the actor's environment and the actor's intention. The very moment that such a perspective is obtained, the correspondence between environment and intentions is re-established: the problem is solved; the actor can act intentionally.*

Still according to the philosophy presented in Chapter 2, the notion of problem solving is embedded in the notion of an *intentional activity*, an activity consisting of the following stages:

stage 1:	acknowledge situation of concern
stage 2:	construct perspective
stage 2a:	*analyse situation "as is"*
stage 2b:	*synthesise script (a plan for taking action)*
stage 2c:	*predict situation "to be"*
stage 3:	implement script (act)
stage 4:	evaluate situation "to be"

Figure 1a: A stage-based model of an intentional activity.

If stage 2 is non-trivial (requires substantial cognitive effort; an appropriate perspective is not readily available), it corresponds with a problem-solving process. Intentional activities were described to be self-contained: the model can be re-applied recursively (at a lower systemic level) and sequentially, resulting in a recursive spiral of intentional activities (figure 1b).

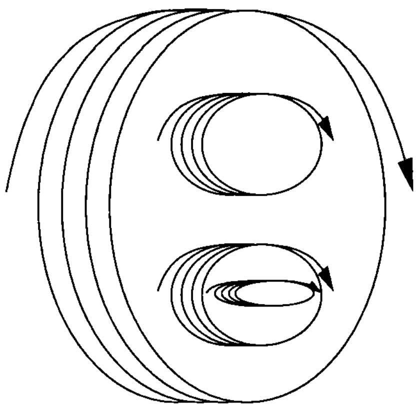


Figure 1b: A recursive spiral of intentional activities.
The notion of intentional activities is self-contained, and may be interpreted as sequentially and concurrently consisting of intentional activities at a lower systemic level.

Trinity straightforwardly adopts the definitions of intentional activities and problem-solving processes, and the central role of the concept of a perspective as described in Chapter 2.

Chapter 3: Knowledge distributions and knowledge processes

In Chapter 3, the first chapter of the Theory part, the notion of KDS (Knowledge Distribution Space) was introduced. KDS offers a means to visualise knowledge distributions and knowledge processes, provides a foundation for a qualitative problem typology on the basis of knowledge distributions, and enables one to relate knowledge distribution characteristics to methodical consequences.

Trinity is a modelling methodology that is especially designed to support problem-solving processes in the D-region of KDS (figure 1c): the intended user is a knowledge broker (see also Chapter 3). *Trinity* provides ways to model complex systems of several actors, acting in a shared environment. Stepping through KDS implies that either the *interpretation* of the body of knowledge (the knowledge distribution) of concern changes, or the very body of knowledge *itself* changes. Inspired by KDS, *Trinity* modelling methods will also provide ways (modelling steps) to change either the interpretation or the very body of knowledge being modelled itself.

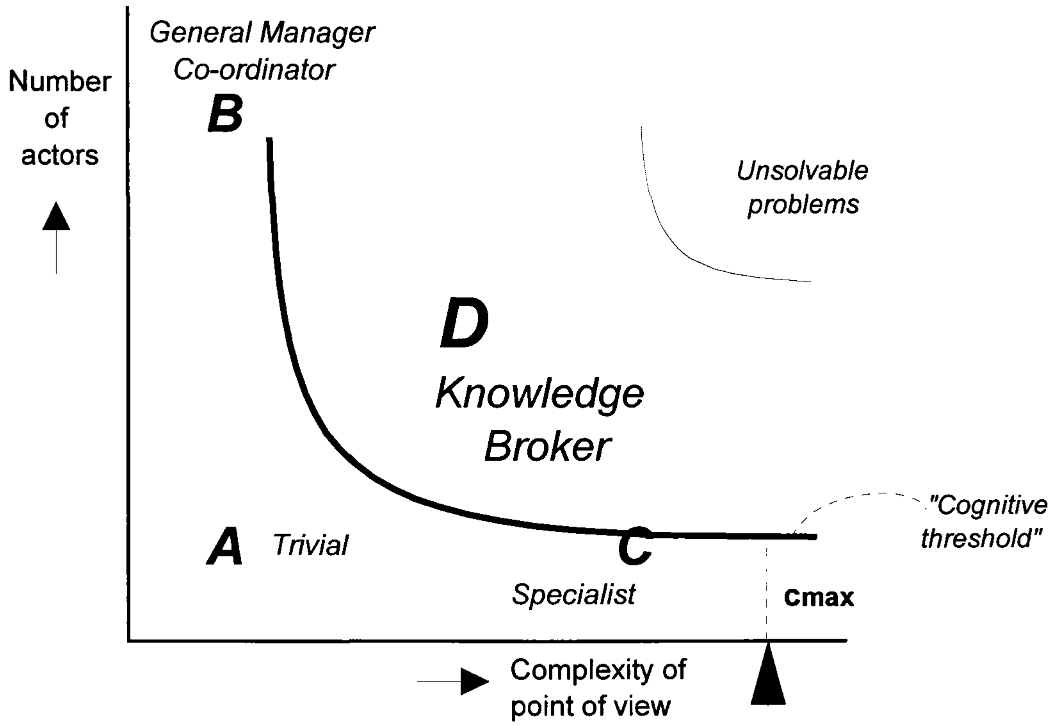


Figure 1c: *Trinity* is directed at supporting knowledge brokers in D-type problem solving. The fact that knowledge brokers operate at the "difficult" side of the *cognitive threshold* (see Chapter 4) implies that their interpretation of the knowledge of concern is necessarily more abstract. Their atomic level (see Chapter 4) is at a higher level of abstraction.

Chapter 4: Models, modelling processes, modelling strategies

Finally, in Chapter 4, a theory of qualitative modelling was presented. The *Trinity* modelling language, to be presented in this chapter, is designed completely in accordance with this theory. Notions like *Trinity* model primitives, *Trinity* models, *Trinity* modelling steps and *Trinity* modelling strategies, that will be presented in subsequent sections, are straightforward consequences of applying the theory of Chapter 4.

The discussion so far reveals the tight relationships between the Philosophical background and the Theory parts of this dissertation and the *Trinity* modelling language to be presented below:

- *Trinity* is directed at D-type problem solving;
- the central concept in *Trinity* is the perspective;
- *Trinity* interprets problem solving as perspective construction;
- *Trinity* provides model-based support for perspective construction; and
- *Trinity* is designed according to the systemic theory of qualitative modelling.

In subsequent sections the *Trinity* modelling language will be presented.

5.3 THE *TRINITY* APPROACH: A BIRD'S EYE VIEW

According to the Philosophy layer of the *Trinity* methodology, it is beneficial for the process of perspective construction to be supported by a *modelling process*. A model is a representation of "something else" (the *referent* of the model). The modelling process enables one to gain an understanding of this referent without directly interacting with it. This means that several alternatives for interaction with the referent (several perspectives) can be explored without changing this referent. At the very moment that the modeller is satisfied with the model, direct interaction with the referent can start. At that very moment, the model is assumed to be a correct representation of both the environment and the intention of the problem owner (see also Chapter 2).

5.3.1 Models in *Trinity*

Trinity provides a modelling language that is specifically designed to support perspective construction in multi-actor (to be more specific: D-type) situations. During the acknowledgement stage of an intentional activity, a first, incomplete and ill-defined perspective emerges. The outcome of the problem-solving process is an explicit qualitative model of a perspective that motivates and guides actions that are expected to result in an improved situation: the difference between the model "as is" and the model "to be" *motivates* taking action, and the script *guides* these actions (the implementation process, stage 3). Evaluation (stage 4) is supported by the models as well, as during this stage the actual situation "to be" is being compared with the predicted situation "to be". Stage 4 is where experiential learning takes place.
This implies that, although *Trinity* models are *constructed* during stage 2, they *support* the complete intentional activity of 4 stages.

The notion of "intentional activity" is recursive (see figure 1b): during such an activity, a stage may turn out to be problematic, which introduces a shift in attention. A new recursive intentional activity (at a lower systemic level) is encountered, and *Trinity* may be applied again[41]. A graphical summary of the *Trinity* approach from a bird's eye view is presented in figure 2a.

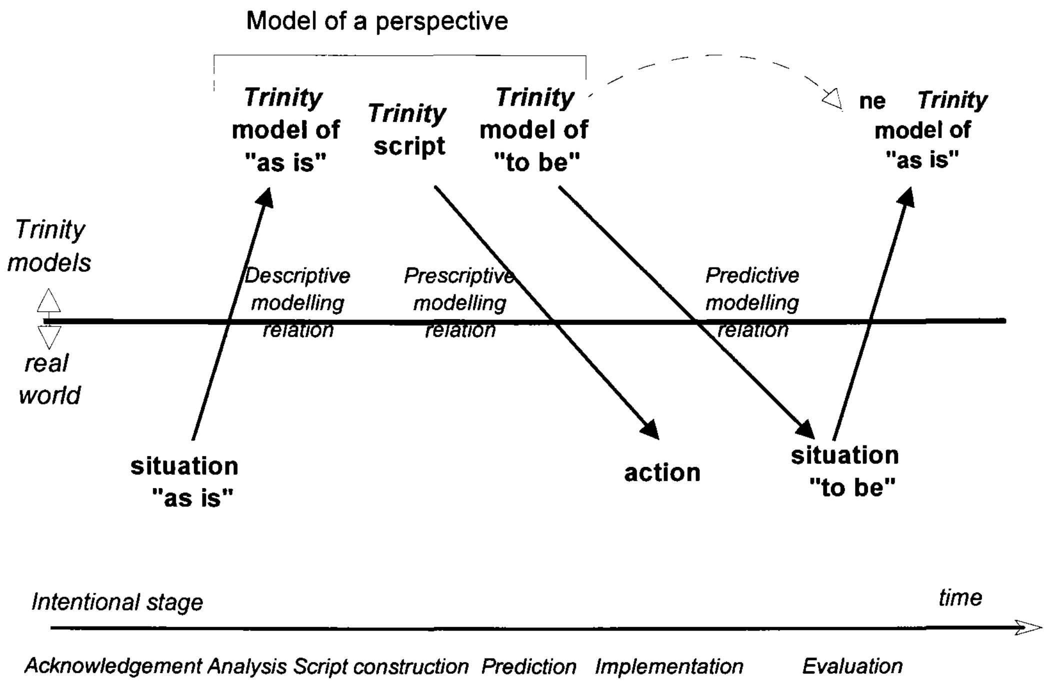


Figure 2a: The *Trinity* approach: a bird's eye view.

During the problem-solving process, the threefold *Trinity* model at each moment in time reflects the adaptations and refinements made to the perspective: the "time series" of models reflects the evolution history of the perspective. From a strictly pragmatic action-oriented point of view, this time series is of no interest: it is the result (the last perspective in the series) that counts. The final perspective fills in the experienced lack of knowledge that was responsible for the problematic situation.

The model relation between perspective and environment is trivalent: it consists of a *descriptive* part (describing the situation "as is", i.e. without intervention), a *prescriptive* part (the script prescribes actions) and a *predictive* part (the expected outcome, the expected "to be" situation).

[41] As was explained in Chapter 2, we do not adhere to the point of view that complex intentional activities should be interpreted as one cycle. Rather, each intentional activity within a larger overall intentional activity can be described as a cycle in turn, and may be supported by *Trinity*. At which systemic level cycles are distinguished is partly a matter of preference of the problem solver, and partly dictated by the problem-solving process itself (as sub-problems are likely to emerge).

Although it is very well possible to order in time the real situation "as is", the execution of a script and the situation "to be", resulting from actually executing the script, it is very difficult to do so with the three cognitive stages (analysis, script synthesis, prediction) associated with them. Therefore, we consider a problem-solving process to be a mixture of all three stages, rather than a sequence (see also the bucket analogy in Chapter 2). During acknowledgement, the first (uncertain, incomplete) notion of a perspective starts to exist. During the following three stages, this first notion is refined until the perspective is considered to be sufficiently elaborated upon to justify taking action according to the script part. In complex problem-solving processes, the perspective is the outcome of a complex mixture of all three activities.

5.3.2 The problem-solving process

Figure 2b puts the use of *Trinity* in another perspective: it presents a problem solver, engaged in an attempt to intentionally change a D-type situation of concern. In doing so, he/she uses *Trinity*.

The figure highlights three core activities:

1. *Trinity* modelling;
2. interaction with the problem context[42]; and
3. knowledge acquisition.

[42] The problem context is the situation "as is", the transition process as well as the situation "to be", i.e. it is the referent of the developing perspective.

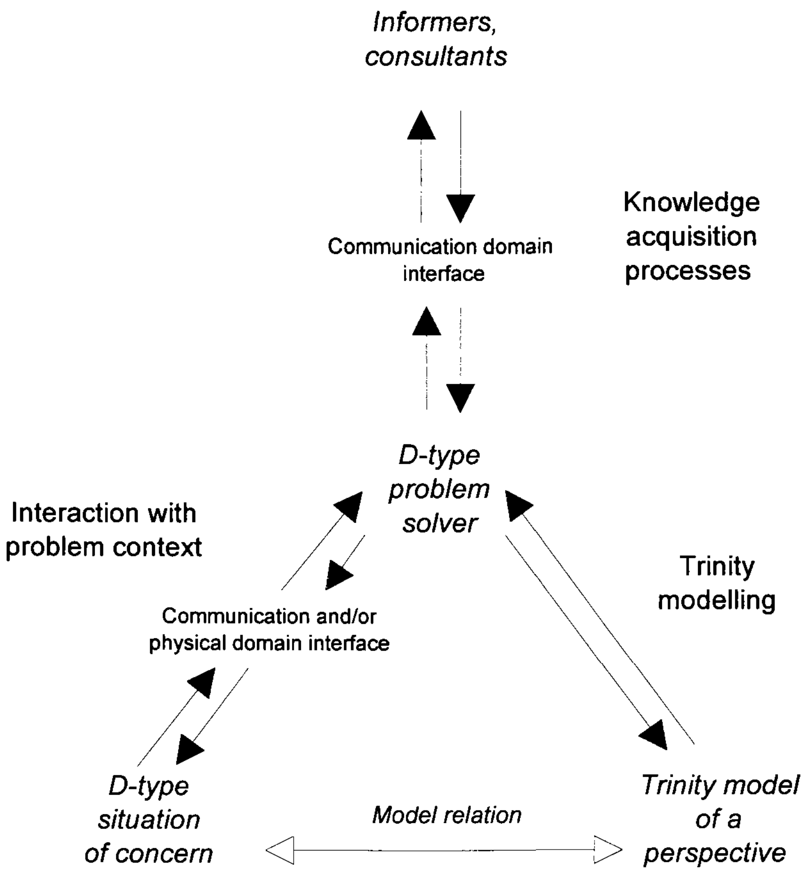


Figure 2b: A D-type problem solver using *Trinity* is engaged in three core activities.

The lower part of figure 2b (the triangle) resembles the meaning triangle (see figure 1 of the theory of qualitative modelling processes, Chapter 4). During D-type perspective construction (the analysis, script synthesis and prediction stage), typically complex knowledge acquisition and communication processes take place: the problem solver communicates with many different consultants and informers. On top of the D-type situation in the referent, a D-type situation may manifest itself during this knowledge acquisition process. This will be elaborated further below.

When being confronted with a multi-actor problem context, typically the first sensation is one of confusion. What actors are part of it, and why, according to whom, what do they do and for what reason? What are their points of view? Are these points of view (in)commensurable? Do different "cognitive maps" exist? What are the "degrees of freedom" of actors involved to change their points of view and their actions? What would be better situations, are these situations realistic and what are feasible routes to attain these situations? Who might help? A myriad of questions emerges. Confusion is abundant. A lack of knowledge exists.

This provides an important handle toward developing means to support multi-actor problem-solving processes. It is not a feasible route to simplify the multi-actor problem context by means of ignoring parts of it: a problem context emerges, and simply is what it is. Or, perhaps more to the point: a problem context is what a problem owner thinks it to be. The problem owner is the individual or the group that considers the multi-actor situation to be a situation of concern: intervention may be required in order to improve it, and a problem owner is seriously considering the possibility of intervention, i.e. taking action[43].

A problem context is confronting: typically it emerges, rather than is deliberately constructed. At least in principle it is possible for a problem owner, however, to improve his *interpretation* of this problem context (according to *his* value system, and in terms of a *perspective*). Improvement of an interpretation is assessed in terms of the degree to which it is known what should be done: confusion is reduced, improvement potential[44] (action potential) is obtained.

Interpretations change as a result of *knowledge acquisition processes*. We use the word "knowledge acquisition" in this dissertation in a very general sense: *any* intentional activity directed at gaining knowledge (i.e. at reducing confusion and obtaining improvement potential) is a knowledge acquisition process. Knowledge acquisition processes are directed at changing an ill-defined and vague interpretation of a problem context (an ill-defined and vague perspective, resulting from the acknowledgement stage) into a preferably well-defined and clear perspective (although in really complex situations "well-defined and clear" may not be an attainable goal). The process stops at the very moment that the confusion is reduced to a reasonable, acceptable level: the problem owner feels that he has a (more or less clear) idea about how to interpret the problem context; he understands the actual situation, knows what he should do, has an idea about the results of these actions and considers these results to be an improvement with respect to the original situation. The problem owner possesses a perspective that (in his/her eyes) is sufficiently elaborated upon to justify intentional action.

A large number of different methods exist that may support knowledge acquisition processes. Typically, knowledge acquisition methods are described in terms of more or less well-defined steps with more or less well-defined outcomes. Structured observations, bilateral interviews, experiments, literature reviews, multi-expert meetings, workshops, decision supporting methods (for example multi-criteria analyses, voting methods), SWOT analyses (Strengths, Weaknesses, Opportunities and Threats), brainstorming, Delphi's, cognitive mapping methods, or using plain common sense, these all are examples of means by which to obtain a better understanding of complex referents. Many of these

[43] According to this criterion, not many of the persons who are talking about "environmental problems" are actually problem owners: they are not seriously considering the possibility of taking action.

[44] The way to improve may very well be: do nothing.

examples form a family of knowledge acquisition methods of their own. And yet, all these (families of) methods merely cover a subset of all the things a problem owner can do in order to obtain a better understanding of the problem context of concern, i.e. to improve his/her perspective.

We consider the availability of knowledge acquisition methods to be of the greatest importance for multi-actor problem solving. Indeed, we use these methods ourselves as building blocks in designing context-dedicated problem-solving approaches in specific multi-actor problem situations. Notwithstanding this, it is not our ambition to develop yet other methods that describe context-dependent knowledge acquisition processes. We do not think that this is the right level of granularity to offer support for multi-actor problem-solving processes in general (see also the introductory chapter). Neither do we think that a lack of knowledge acquisition methods is the key bottleneck that prevents multi-actor problem solving to be effective and efficient (although a practical set of rules of thumb with respect to their applicability, their strong points and their weaknesses would be a great help).

In this chapter we will rather make an attempt to develop generic methods that may support *many different* multi-actor problem-solving processes. That is, methods that are not too restrictive with respect to peculiarities of the multi-actor problem context of concern, nor with respect to the specific stage in the problem-solving process of concern[45]. The methods will not emphasise or prescribe specific knowledge acquisition methods: this issue is deliberately left out, as for this purpose the methods mentioned above (and many more) provide an excellent (albeit scattered) library. Knowledge acquisition methods are included indirectly, as they are to be used *within* (are selected and embedded in) the modelling methods: they guide the processes that constitute the knowledge acquisition steps within the problem-solving process as a whole.

Instead, *Trinity* methods will focus on the *complement* of knowledge acquisition methods, i.e. the inputs and outputs of knowledge acquisition processes: the interpretation of (parts of) the problem context of concern in terms of models of perspectives. The methods to be presented in this chapter enable one to ***represent*** a problem owner's interpretation of his multi-actor problem context in terms of a perspective, and this during the ***complete*** development (evolution) from an ill-defined and incomplete interpretation (i.e. the situation of confusion mentioned above) via a possibly large number of intermediate versions towards an interpretation in which confusion has disappeared to such a degree that the problem owner feels that he is able to take purposeful action (see figure 2c). The *Trinity* modelling language does describe the very modelling steps themselves as well, but this only in superficial terms.

---

[45] In terms of the spectrum "specific approaches" and "generic approaches" ("specific" implying much support in few situations, and "generic" implying little support in a wide range of situations), the *Trinity* methodology is to be situated at the generic side. The benefit of this is twofold: it covers a wide range of potential applications, and it does not require to change methodology during a (complex) multi-actor problem-solving process. The potential draw-back of this choice, i.e. that the methodology may provide too little support in specific situations (genericity may result in triviality), is counterbalanced by the pleasant feature that other methods (for example, knowledge acquisition methods, or textual attachments to parts of a model) can be embedded at an "if-needed" basis within this "backbone" methodology.

Such a representation can, for example, exist in a mental way or can be made explicit by means of natural language (e.g. a policy document, a plan or an oral presentation). Explicit representations have the advantage over implicit (mental) representations in that they can be shared: this opens the possibility to exchange, discuss, rethink and adapt this representation. However, a natural language representation does not specifically address or emphasise the complexity and the peculiarities of multi-actor problem contexts. Therefore, in order to provide a generic means to support multi-actor problem solving, we considered it important to design a dedicated *modelling language* that enables us to represent and adapt (intermediate) interpretations of multi-actor problem contexts. The pivot of the *Trinity* methodology is this modelling language. This modelling language enables us to represent developing perspectives, and hence provides model-based support for multi-actor problem solving.

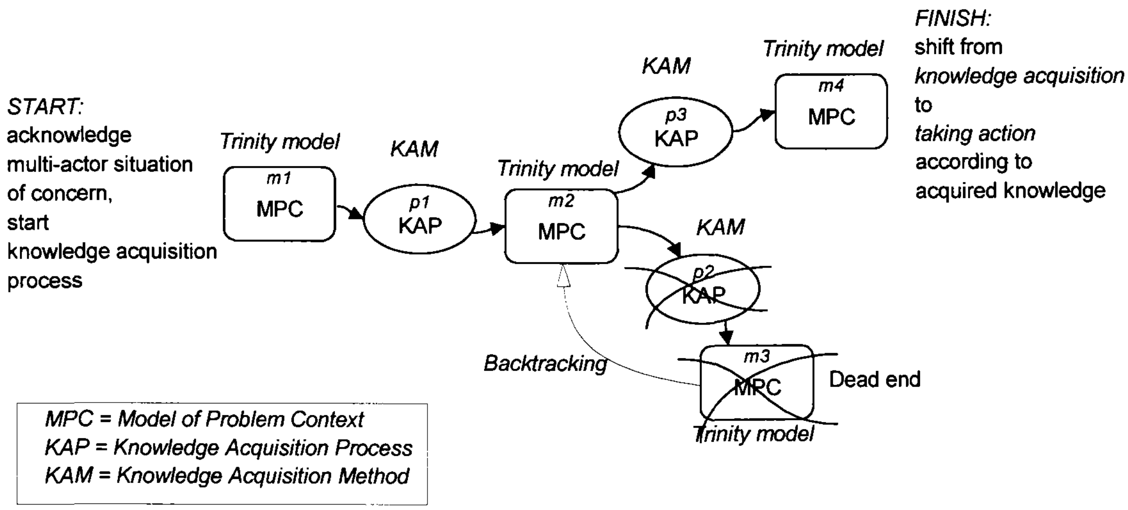


Figure 2c: A sequence of *Trinity* models represents an evolving interpretation of a problem context in terms of a perspective.

Each intermediate *Trinity* model *reflects* the knowledge the problem solver has acquired so far. On the other hand, the model *suggests routes* to continue the knowledge acquisition process. The model has an important *heuristic* function on top of a *representational* function (for an overview of functions of models, see [Diepenmaat (1993a)], and Chapter 2). For example, the model may suggest that actors are missing or that specific parts require further investigation. The knowledge acquisition process and the *Trinity* modelling process support and guide each other.

In many cases, actors that are part of the situation of concern also assume the role of informer/consultant: they play a dual role. This implies that the process of perspective construction may influence the problem context: a phenomenon that has been well recognised in many fields of multi-actor research. For example, asking an actor why he behaves in a specific manner may cause a reflective process in this actor, which in turn may result in different perspectives, hence different behaviour.

5.4 THE *TRINITY* MODELLING LANGUAGE

In this section, the *Trinity* language will be presented: the conceptual tool to model an evolving perspective of a multi-actor referent. The *Trinity* language is designed according to the theory of qualitative modelling processes, presented in Chapter 4 of this dissertation. In line with this theory, the design of the *Trinity* modelling language consists of four steps:

> **Step 1:** specify the qualities (concepts and relations) you want to be able to distinguish in a referent;
> **Step 2:** select symbols to refer to these qualities;
> **Step 3:** design ways to represent complex model relations; and
> **Step 4:** design modelling steps that enable one to change these model relations.

These steps will be worked out in paragraphs 5.4.1 - 5.4.4, respectively.

5.4.1 Specify qualities of interest (Step 1)

Different interpretations of environmental problem contexts exist (see, for example, [de Groot (1992)], [la Rivière (1991)]). According to the ***Trinity* principle**, three different, yet tightly related, domains must be distinguished in multi-actor problem contexts. They are:

1. the physical domain;
2. the knowledge domain; and
3. the communication domain.

The *Trinity* principle states that real-world phenomena are recursive systems of (or at least: can be understood as; can be modelled as recursive systems of) processes and states in these three domains (hence *Trinity*[46]). Actors act according to private points of view (perspectives), which are subject to change. They do so in a shared environment, consisting of physical phenomena and communication phenomena.

In order to support D-type environmental problem-solving processes, it is important to provide knowledge brokers with conceptual tools that support the process of obtaining an understanding of such complex three-domain problem contexts. For this very reason, *Trinity* provides a modelling language.

---

[46] Actually, the name *Trinity* has a threefold meaning. This meaning is explained in depth in Appendix D.

In line with the *Trinity* principle, the qualities of interest (the concepts considered to be relevant to be distinguished in a referent) can be classified according to two dimensions. The first dimension is: they belong to the *physical domain*, the *knowledge domain* or the *communication domain*. The second dimension is: they qualify as a *state* or as a *process*.

The application of a combinatorial scheme to the two dimensions results in six different qualities: *physical states, physical processes, knowledge states, knowledge processes, communication states* and *communication processes*. Note that these six types of qualities (or combinations of them) cover "everything that can be modelled" from a *Trinity* point of view. It is assumed that these six qualities enable one to have a complete coverage of real-world phenomena. Other qualities (with the exception of relational qualities like "causation", see further) would fall outside the scope of a *Trinity* model. Below, the meaning of these six qualities within the *Trinity* approach will be explained further.

The physical domain

Physical we call all the things that, at least in principle, are observable (sensory perceptible) by anybody.
Examples of *physical states* are: the presence of a building, a specific concentration of ozone in the atmosphere, the presence of noise.
Examples of *physical processes* are: quantum effects in atoms, a chemical reaction, the breaking of a window, the rotation of the planet Earth, the expansion of the universe.

The knowledge domain

Within the limits of our theory, *knowledge* is a potential to act intentionally in either of the three domains.
Knowledge states are *perspectives*: parcels of knowledge possessed by either an individual or a number of individuals, that enable this individual or group to act purposefully. Perspectives enable intentional, purposeful action because they consist of a descriptive part, a prescriptive part and a predictive part. The actor models his actual situation with the descriptive "as is" part of the perspective; executes the script part; and expects the predictive "to be" part to be a correct model of the outcome of his actions. For example, when applying the "lawn mower" perspective, this perspective refers to a state with long grass, followed by a mowing process, followed by a state with short grass. The person mowing the lawn observes long grass ("as is"); starts mowing (script execution); and expects short grass ("to be").
Knowledge processes change the set of perspectives that are available for an actor, and as such change the action potential of this actor. Examples of knowledge processes are learning (i.e. learning the "lawn mower" perspective), revising (i.e. improving the "lawn mower" perspective) and forgetting.

The communication domain

Communication is a process of information exchange between actors with the intention to change the action potential of participating actors (i.e. to inform or influence each other).

Actually, it is not perspectives that are exchanged. Rather, a sender represents parts of perspectives in the physical domain (by means of speech, writing, or any other means of communication), and a receiver interprets these representations. Agreement about perspectives (the success of communication) is tested in pragmatic situations, by means of comparing expected and real behaviour of participants in communication. For example, when I ask you to give me the red book on the third shelf, and you hand the red book to me, I can be confident that the transfer was successful (see also [Winograd and Flores (1986)].

A *communication domain state* is a static representation of knowledge used during a communication process. Examples are: a library, a report, a speech, a book, a model.

A *communication domain process* is a process that changes the communication domain. Examples are: writing a letter, collecting a library, uttering words, but also burning a book or destroying a floppy disk.

Communication domain phenomena *refer to* the knowledge domain, but are *embodied* in the *physical* domain.

States and processes

States are the inputs and outputs of processes. Processes transform states: they change a domain. A process typically takes one or more states of a domain as input, and results in one or more states of the same domain as output. An exception is a process in the communication domain: here it is possible that the process has no communication domain state as input. For example, when I write a note, the resulting communication domain state (the note) results from a communication domain process (the writing process), but this process did not take a communication domain state as an input. However, when I am revising part of this dissertation, the input state is the original text, and the output state is the new (revised) text.

According to the *Trinity* point of view, states and processes do not exist in an absolute sense. Examples in the physical domain are: the presence of a Deux-Chevaux is a state, but at the same time this beautiful car is likely to corrode, which is a process; the pencil in my hand is a state, but at the same time the molecules constituting this pencil vibrate; the rivers in the Netherlands are states from a geographical point of view, but they flow at the same time, and from a historical perspective they even move. An example in the knowledge domain is: universities possess the knowledge to perform research activities (a state), but at the same time this knowledge continuously changes (a process). An example from the communication domain is: a library consists of books (a state), but the books actually present in the library change over time (a process).

Therefore, we define a state as something that is static *with respect to* the (intentional or autonomous) processes that one is interested in. Conversely, a process is something that changes (or results in) a state that one is interested in. From a philosophical point of view, this is a combination of a relativistic and a pragmatic interpretation (something *is* the action *potential* it offers to *someone*). In line with this relativistic nature of states and processes, *Trinity* provides ways to change states into state transitions, and vice versa (see the dynamic modelling steps further on).

5.4.2 Select symbols to refer to qualities (Step 2)

The second step in designing a modelling language is to associate each quality with a symbol. We call these symbols *model primitives*: they constitute the building blocks of a model. Six different qualities are distinguished, which would imply six different symbols. However, it is not required to select different symbols for the three types of processes (in different domains), because the domain of a process is the same as the domain of its input and output states. As a result, only four different model primitives are required.
We will use an *ellipse* to refer to a process (in any of the three domains); a *hexagon* to refer to a perspective (a knowledge domain state), a *rounded box* to refer to a communication domain state, and a *rectangle* to refer to a physical state. Figure 3 presents examples.

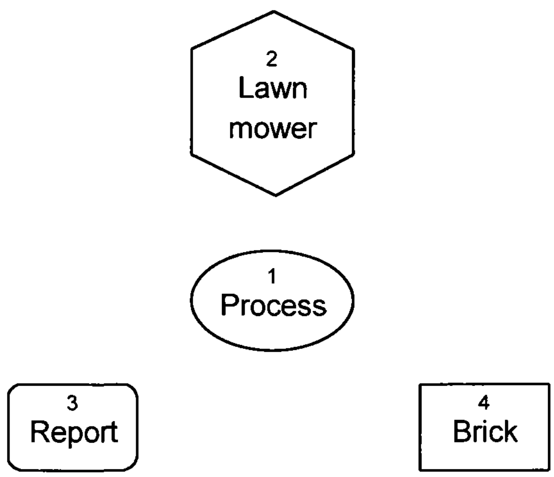


Figure 3: Reference model primitives.

The examples illustrate that the model primitives can be numbered, and are provided with a name that should enable one to understand the semantic correspondence (i.e. to relate the symbol to its referent). For this reason, they are called *reference model primitives*. A process (ellipse) typically has an "active" name like "mows lawn" or "breaks". A communication domain state (rounded box) typically has a name that refers to a physical communication means, like "book", "report", "speech", or "library". A physical state (rectangle) typically has a name like "tower", "bridge", or "molecule". A hexagon (perspective) typically is named after the purposeful action that it will motivate and guide: for example, the "research performer" perspective consists of the knowledge required to perform research; the "lawn mower" perspective consists of the knowledge required to

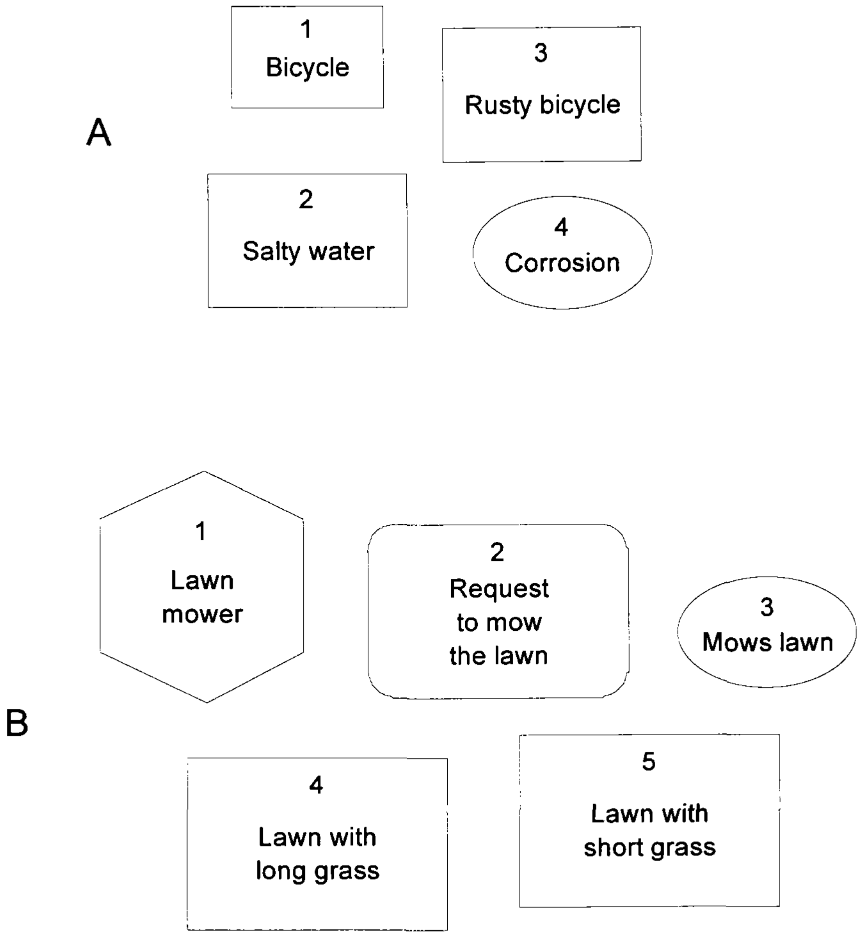
mow a lawn; and the "air traveller" perspective consists of the knowledge required to be able to travel by air. Note that these names correspond with the name of the actor who possesses the perspective and is able to act accordingly.
In order to facilitate semantic correspondence (to relate a symbol with its referent), model primitives can be further described in "natural language" by means of a textual attachment. The number, in combination with the name, provides a reference to this textual attachment. Another way to elaborate upon a model primitive is to detail it on a lower systemic level by means of parallel specifications (see further on).

5.4.3 Design ways to represent complex model relations (Step 3)

The examples in figure 3 illustrate the way in which simple model relations, relating one model primitive with one referent, can be modelled. However, the modelling language should enable one to represent more complex model relations as well. Complex model relations consist of several simple model relations. In Chapter 4, we distinguished three different types of complex model relations: *parallel* relations (a model consisting of several model primitives, each of them referring to a part of the overall referent of this model); *multi-referent* relations (one model referring to several referents); and *multi-representation* relations (one referent being modelled by several models, i.e. several viewpoints). Each of these three types of complex model relations requires specific representation conventions. They are presented below.

5.4.3.1 Parallel relations

Representing a parallel model relation requires the combination of several model primitives into one complex model. The model primitives in combination constitute a "whole": they are *part of* the complex relation.
For example, the state "Bicycle", the state "Salty water", the process "Corrosion" and the state "Rusty bicycle" in principle offer the opportunity to model a bicycle becoming rusty because of exposure to salty water. A second example is the situation in which a lawn mower (the person, that is) is requested to mow the lawn, and subsequently does so. A rounded box "request to mow the lawn", a rectangle "lawn with long grass", a hexagon "lawn mower", an ellipse "mows lawn" and a rectangle "lawn with short grass", in principle, offer the opportunity to model this situation.
The resulting sets of reference model primitives, however, hardly constitute clear models as a whole (see figure 4). Some additional conventions are required to be able to construct models consisting of several reference model primitives. From a knowledge theoretical point of view, descriptions (hence models) require two types of primitives: concepts and relations between them (see also [Meehan (1988)]. This enables one to represent clauses like: x *is caused by* y, x *is left from* y, x *is greener than* z. Reference model primitives represent concepts, but a means to refer to relations is still missing. In *Trinity* (at this moment) only one type of relation is distinguished: the **causal relation**.



Figures 4a and 4b: Complex models that do not make sense.

The first example (figure 4a) consists of model primitives of only one domain (the physical domain). It emphasises that a means to represent a *temporal order between reference model primitives* (for example, a causal relation) is lacking. The second example (figure 4b) consists of model primitives of several domains. It emphasises that, on top of a temporal ordering, a clear way to *link different domains together*, must still be developed.

Temporal ordering: causation

In order to be able to represent temporal ordering, an additional model primitive is provided: the *arrow*.

The arrow refers to a *causal relationship.* Causation takes time: arrows represent this time span. However, not every temporal relation is causal. Event x may happen later than event y, but they may be totally unrelated.

A causes B means that if we would intervene and prevent A from existing at time a, B would not be present at a later time b. This interpretation enables one to represent that A is *necessary* to cause B, although the existence of A may be *insufficient* to cause B. Consider, for example, the use of an automatic cash dispenser. Most people would agree that inserting a credit card would cause the possession of money. Indeed, this is *necessary*. However, this is not *sufficient*: the machine should be filled with money, the machine should operate properly, the card should not be damaged or expired, et cetera. Another example is lighting a candlestick with a match: the presence of oxygen is required, and yet normally this is omitted in a "candlelighter" perspective.
By means of the arrow, it is possible to turn the example presented in figure 4a into a clear model. This is presented in figure 5a. Salty water and a bicycle, in combination, cause a corrosion process (note that the presence of oxygen is missing!), which causes a rusty bicycle. The ellipse in figure 5a refers to an **autonomous** process, rather than to an **intentional** action. This is clear from the fact that the ellipse is not connected with a perspective (the process is not guided and motivated by a perspective). Intentional processes (actions) *are* connected with a perspective. This will be explained in a moment.

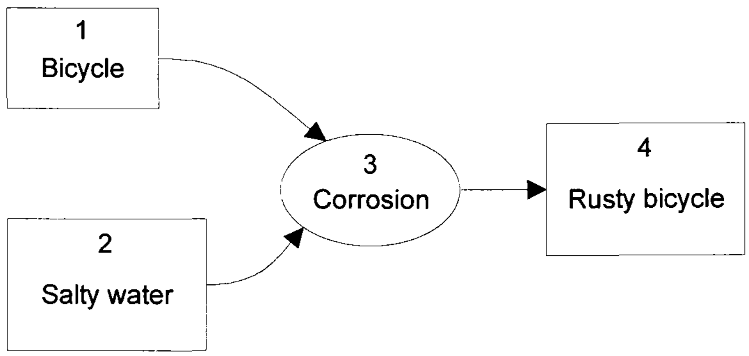


Figure 5a: Causation in *Trinity* models: an autonomous process.

The rules for connecting reference model primitives, belonging to the same domain, by means of causation arrows are:

1. An arrow can be maximally connected with two reference model primitives: one at the arrow's head and one at the arrow's tail. The one at the arrow's tail is said to cause the one at the arrow's head.

2. The boundaries of a reference model primitive can be connected with the heads or tails of any number of arrows. This is similar to a logical "and": all the incoming arrows in combination cause this reference model primitive, and all the outcoming arrows are caused by it.

An arrow that is connected with only one reference model primitive, or a reference model primitive that is only connected with incoming or outgoing arrows, is at the system boundary of a model.

Linking domains: intentionally acting actors

In figure 4b model primitives referring to *different* domains are present. Somehow, the knowledge domain, the communication domain and the physical domain must be integrated into a whole that makes sense. The concept that enables us to do so is the notion of *an actor, intentionally acting in an environment*. This notion can be modelled by means of using reference model primitives and arrows in combination. The lawn mower example of figure 4b is used as an example to illustrate this (see figure 5b).

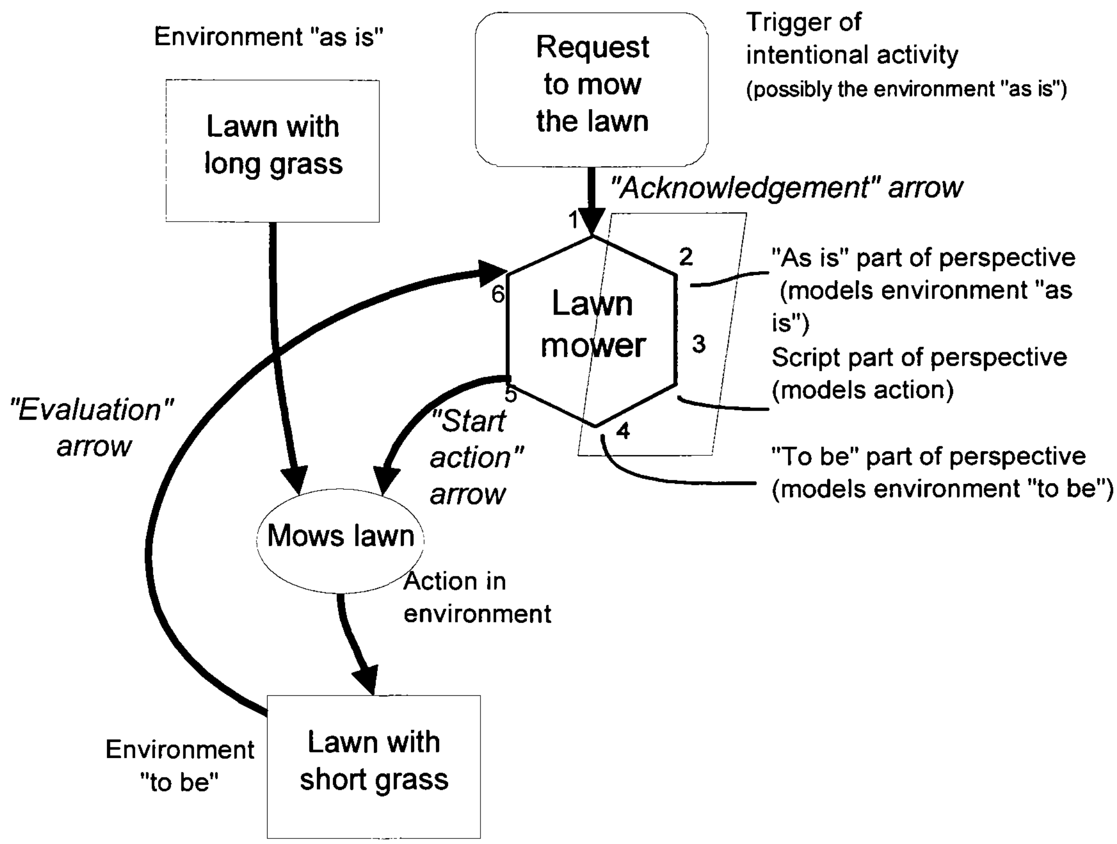


Figure 5b: Representing intentional activities.

A request to mow the lawn (a communication domain state) causes the person who is to mow the lawn to acknowledge that some action is required: the actor becomes intentional (the acknowledgement arrow, connected to position 1 of a hexagon). The lawn mower possesses the lawn mower perspective (modelled by the hexagon). Possession of this perspective causes him (the "start action" arrow departing from position 5) to actually start mowing (a physical domain process, modelled by the ellipse: in contrast with the bicycle corrosion example, this ellipse refers to an intentional action and not to an autonomous process). The mowing process turns a physical state "as is" (modelled by the input rectangle) into a physical state "to be" (modelled by the output rectangle). The physical state “to be” results in an evaluation: the actor verifies whether expectations (the “to be” part of the perspective) and results agree. The acknowledgement arrow, in a sense, is the inverse of the evaluation arrow: acknowledgement starts an intentional cycle, and evaluation ends an intentional cycle (although in case of a mismatch between predicted and actual “to be” further cycles may be called for, resulting in a spiral).

The six corner positions of a *hexagon* are given a special meaning: they correspond with the six stages of an intentional activity (the three perspective construction stages are separated). More specifically:

Position 1 (at the top) refers to the *end* of the *acknowledgement* stage: a situation that potentially requires taking action has been acknowledged. The intentional activity has started.

Position 2 (clockwise) refers to the "*as is*" part of the perspective (i.e. the result of analysis).

Position 3 refers to the *script* part of the perspective (i.e. the result of the script construction stage).

Position 4 refers to the "*to be*" part of the perspective (i.e. the result of the prediction stage).

Position 5 refers to the *start* of the *script implementation* process. Action in the environment begins.

Position 6 refers to the *end* of the *evaluation* stage. The intentional activity ends (but another one may start).

Note that positions 1, 5 and 6, on the one hand, and positions 2, 3 and 4, on the other, are different. Positions 1, 5 and 6 have to do with starting an intentional activity, starting an action and ending an intentional activity, respectively. They are the interface between perspectives and driving forces (causations), so to speak. Positions 2, 3 and 4, on the other hand, represent the three parts of the perspective that makes the action intentional. This is in agreement with Chapter 3, where a simple philosophy of problem solving was presented, based on three concepts: *perspectives*, *intentions* and *environments*. The perspective (the knowledge) is represented by the hexagon, and, more specifically, the lower right side (shaded in figure 5b). The intention (the driving force) is represented by the causation arrows “acknowledge”, “start action” and “evaluate” (if they would be absent, an intention would not exist). Finally, the environment is modelled by means of the reference model primitives around the hexagon.

The lawn mower model represented in figure 5b is well balanced: it is quite natural to imagine the lawn mower receiving a request, and to see him start mowing the lawn by means of a grass mower, which results in short grass. The model enables one to understand the links between the intention, the perspective and the environment. It is a model of an intentional activity indeed. To be more specific: the model is *pragmatically correct*. In figure 5c, two examples are presented that are modelled *syntactically correctly* (all connections between reference model primitives and arrows are legal), *semantically correctly* (every reference model primitive refers to a referent of the appropriate domain and type (state or process), and every arrow refers to a causation), but *pragmatically incorrectly*.

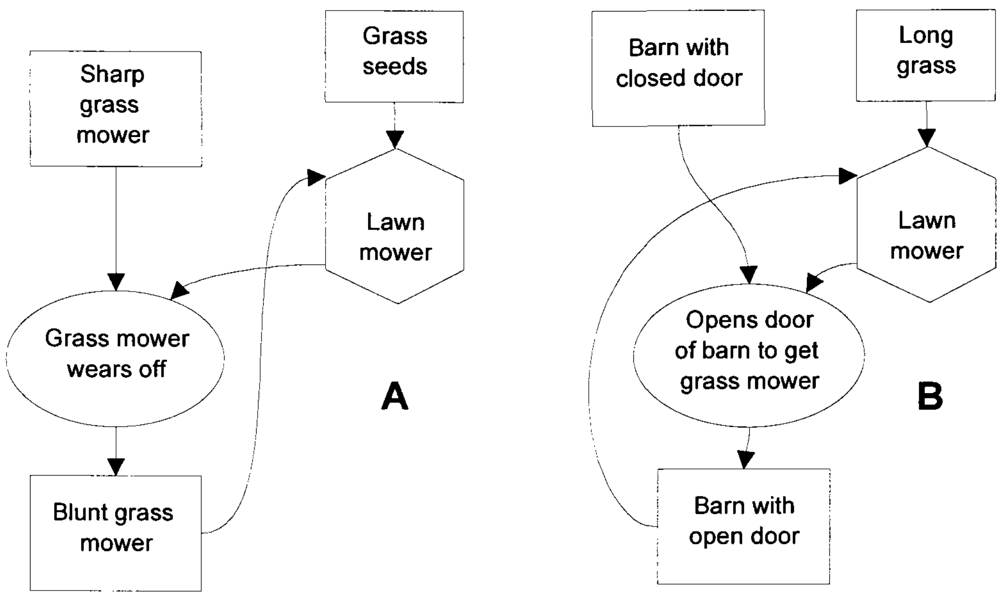


Figure 5c: Pragmatically incorrect models of intentional activities.

Example A is *semantically correct*: grass seeds may cause grass to grow, which may cause the "Lawn mower" perspective to become applied, which may cause a lawn mower to mow, which may cause a grass mower (a machine) to wear off. Each of the model primitives can be given a referent, and each of the arrows can be given a causal meaning. However, the model is *not* pragmatically correct: the rectangle "Sharp grass mower", the ellipse "Grass mower wears off" and the rectangle "Blunt grass mower" are not referents of the "as is" part, the "script" part and the "to be" part of the "Lawn mower" perspective. In addition, the arrow causing the "Lawn mower" perspective to become applied cannot be

interpreted as an observation (though this should be an *observation* as the connected reference model primitive is a physical domain state).
In example B, applying the "Lawn mower" perspective causes the process "Opens door of barn to get grass mower": it is semantically correct. However, this process seems to be too specific to be coupled with the "Lawn mower" perspective: it is part of the lawn mower's preparatory actions, rather than an implementation of the script. Several systemic levels seem to be mixed. The rectangles "Barn with closed door" and "Barn with open door" are not instantiations of the "as is" and "to be" parts of the Lawn mower perspective. The arrow between the rectangle "Long grass" and the hexagon "Lawn mower" in this case can, however, be interpreted as an observation that causes the perspective to become applied.
Appendix B, *Trinity modelling conventions: pragmatic correctness*, provides an overview of syntactical, semantic and pragmatic conventions of the *Trinity* language.

A remarkable feature of models of intentional activities, as presented in figure 5b, is that they represent a model relation: the perspective (hexagon) refers to the environment (two rectangles and an ellipse), and both the perspective and the environment are represented in the model. This resembles the notion of endomorphic systems [Zeigler (1990) p. 16], i.e. systems in which some sub-object uses models of other sub-objects. One might wonder whether this is necessary: simply using a notation in which hexagon and ellipse are collapsed would result in a far more simple model (for example, "rectangle causes triangle causes rectangle", in which the triangle represents the intention, the perspective and the action). However, it must be kept in mind that when intervening in multi-actor problem contexts, two basic types of strategies can be adopted: strategies that aim at changing the environment directly or strategies that aim at changing the perspectives of actors. Models of intentional activities provide ways to *explicitly model both types of strategies*, as both perspectives and environments are present in the model. An example will explain this.

Consider the situation of a very dry and hot summer. The results of your efforts with respect to mowing the lawn are not very good: each time you mow it, the grass is more brown than before. However, you have a neighbour (with a green lawn, by the way) who knows a little trick that prevents this from happening. Mow more often, but not so short. The neighbour (to keep the example simple) can do either of two things: he can start mowing your lawn, obeying this simple rule, or he can teach you the trick. The first strategy would be directed at intervening in the environment. The second would be directed at changing your lawn mowing perspective.
Let us simulate what might happen from the neighbour's point of view. He acknowledges that the results of your method of mowing the lawn are not very good, and decides that he must do something about it: he becomes intentional (stage 1). After careful deliberation (stage 2: problem solving) he possesses an appropriate perspective. The results of his analysis are: you mow not often enough and you mow too short, resulting in brown grass. The script: he intends to advise you to mow more often and not so short, and you will take his advice. His prediction: you will mow the lawn using a new perspective, resulting in a

green lawn. A model of the neighbour *as an intentional actor* is presented in figure 5d. A model of the neighbour's *perspective* is presented in figure 5e. Running slightly ahead: the perspective presented in figure 5e is a *specification* of the environment as modelled in figure 5d (see also the text in figure 5d).

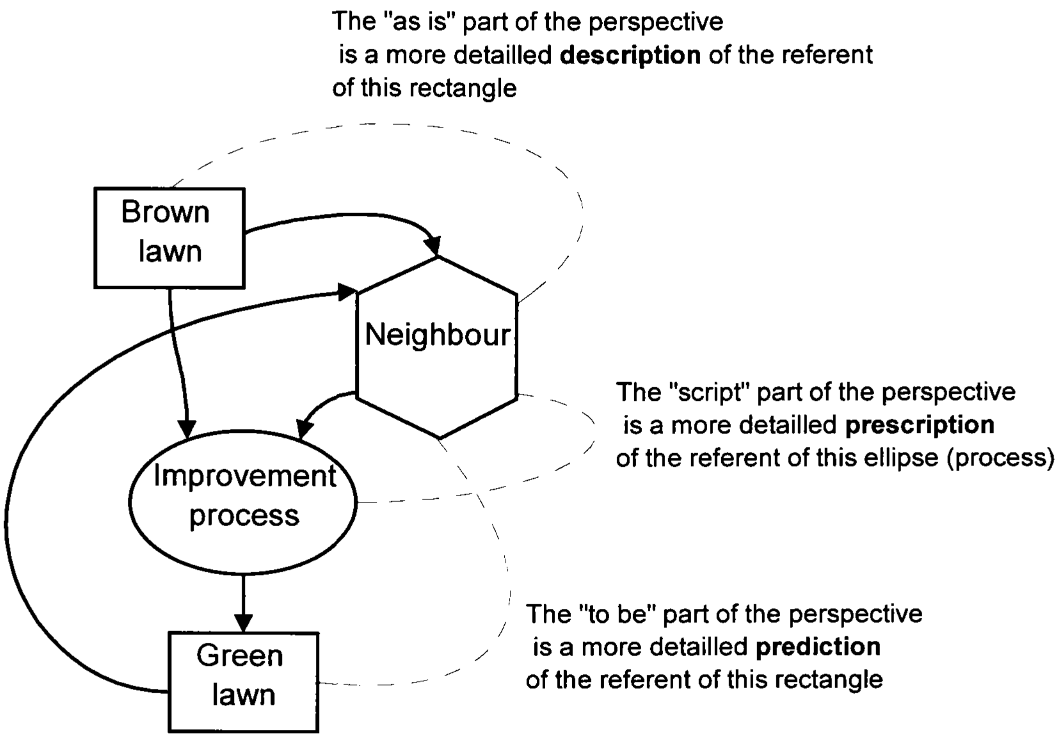


Figure 5d: The neighbour modelled as an intentional actor.

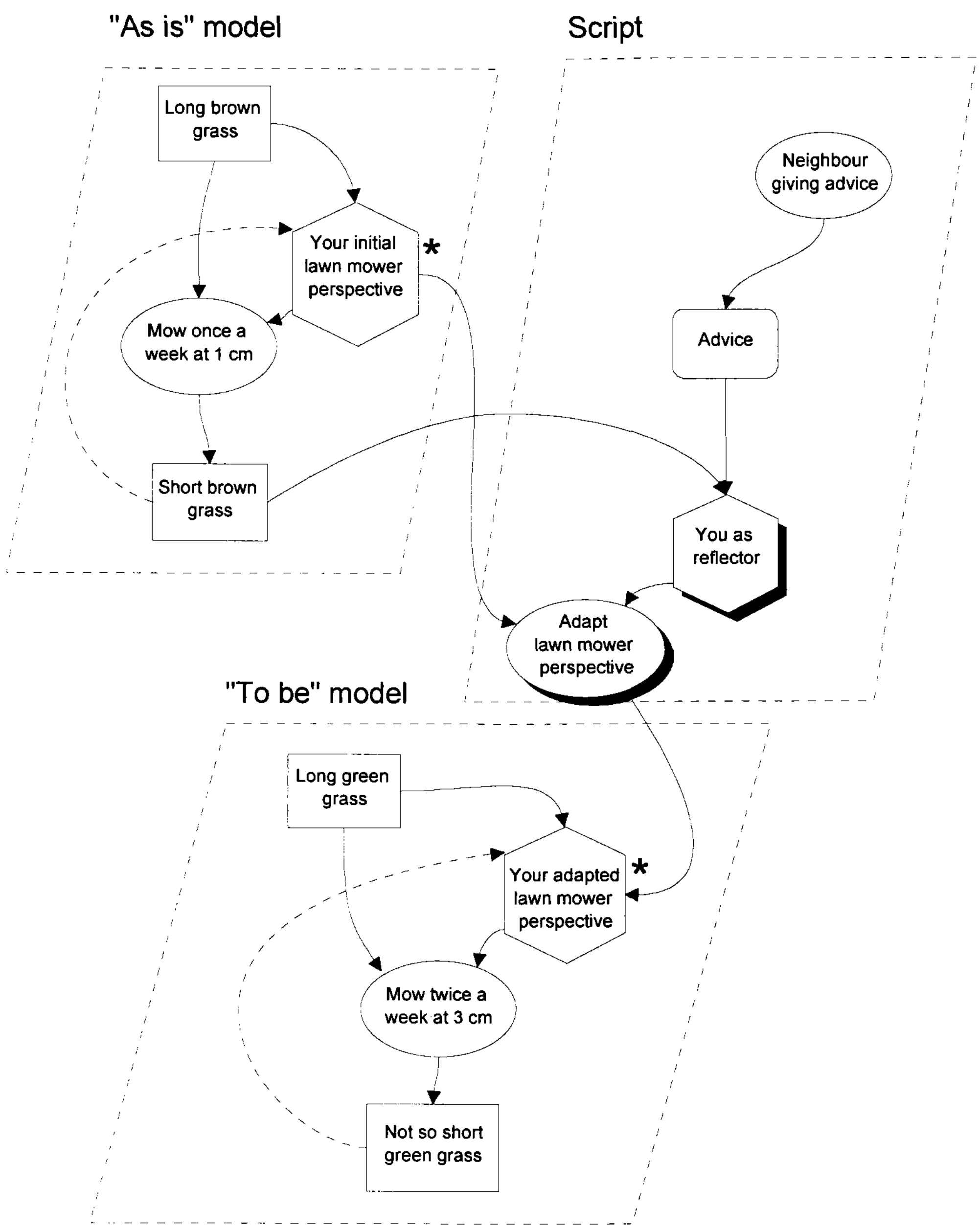


Figure 5e: A model of the *perspective* of the neighbour (which is a more detailed model of his environment).

Figure 5e presents an example of **reflection**: an intentional action that takes place in the knowledge domain. The input as well as the output of the shaded ellipse are *perspectives* rather than *physical states* (as is the case with the two "mow" ellipses). Note that in this reflective case the arrows depart or enter at the middle of a side of a perspective, rather than at a corner position (see the asterisks in figure 5e).

Another example is presented that illustrates the expressiveness of distinguishing *both* perspectives *and* environments in one model. Consider the situation that you, a policy maker, want to prevent batteries from ending up with the normal household garbage. Your perspective might be: in the "as is" situation, batteries end up with the normal garbage; the script specifies that you must initiate both an information campaign (a strategy aimed at changing the knowledge domain) and an action directed at providing battery buckets at street-corners (a strategy directed at changing the physical domain); the "to be" situation is expected to be that citizens throw their batteries into these buckets. A *Trinity* model enables one to model this perspective as a whole (figure 6).

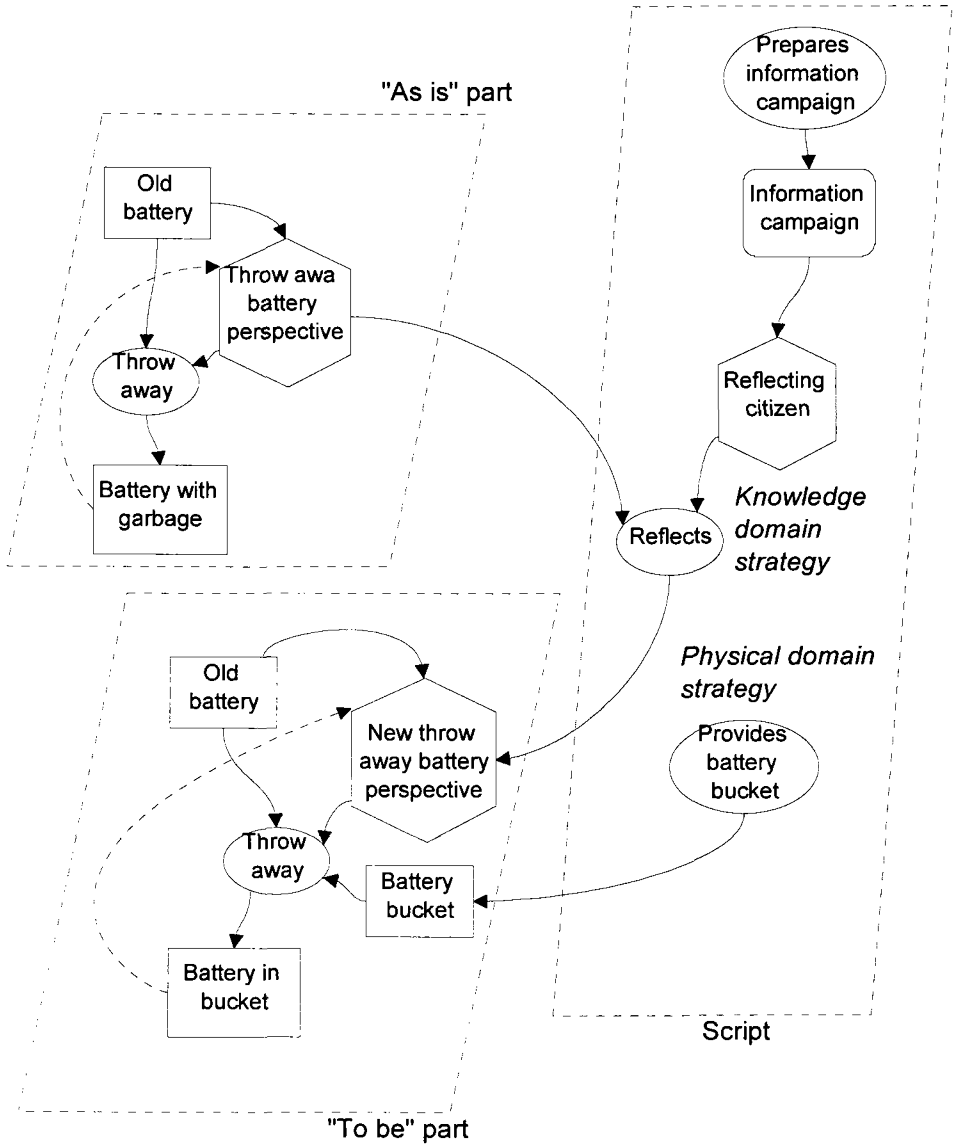


Figure 6: A model of a perspective, aiming at changing both the physical environment and the perspectives of citizens.

Of course, the perspective of figure 6 requires more elaboration. For example, preparing an information campaign and designing and distributing buckets are complex activities that can be detailed (additional actors and actions are required). Ways to do this will be presented in subsequent sections.
In summary, the rules for representing an actor engaged in an intentional activity are:

1. The central primitive in modelling an intentional activity is a hexagon (perspective).

2. This perspective must be caused by a reference model primitive of one of the three domains, at position 1. This arrow represents the acknowledgement stage. The arrow must be interpretable as an observation, a perception, or anything that triggers an actor.

3. The perspective must cause a process: the arrow departing from the implementation stage (corner 5) should cause an ellipse. The process to which this ellipse refers should be the referent of the script part of the perspective.

4. The process caused by the perspective should change one of the three domains (i.e. change an initial state into a final state, or introduce a new state). The initial state is the referent of the "as is" part of the perspective; the final state is the referent of the "to be" part of the perspective.

5. A state downstream of the process must be evaluated by the actor. This is represented by an arrow from this state to position 6 of the perspective. In the case that state(s) directly downstream of the implementation ellipse is/are both the output state(s) and the intended state(s) (see the dashed evaluation arrows in figures 5^e and 6), the evaluation arrow may be omitted (for reasons of convenience).

5.4.3.2 Multi-referent relations

In multi-referent model relations several referents are modelled by one and the same model. The model is said to be generic (applicable to the members of a genus, a class). For example, the formula F=m.a (Newton's law) can be applied in many different situations; the handbook of a Deux-Chevaux mechanic applies to many Deux-Chevaux. This is beneficial, as the same model can be used on several occasions. Examples of *Trinity* generic models will be presented when *Trinity* modelling strategies are discussed (see section 5.5).
Normally, multi-referent model relations are not visible in a model, as the referent part of a model relation typically is not represented. We will use a simple solution for making explicit the fact that a model is generic: we will represent this in the title of the model. For example, “Generic model of indoor environment”.

A special case is the situation in which only a smaller part of a larger model is generic. An example is a *time loop* in a *Trinity* model. A loop can be recognised as a circular path of arrows in a model. For example, at several stages in a complex production process the results so far are examined by a quality controller. This quality controller appears several times, but is represented only once. The example is presented in figure 7 (evaluation arrows are left out). We will use the convention that generic parts of a larger model are printed in bold (see figure 7).

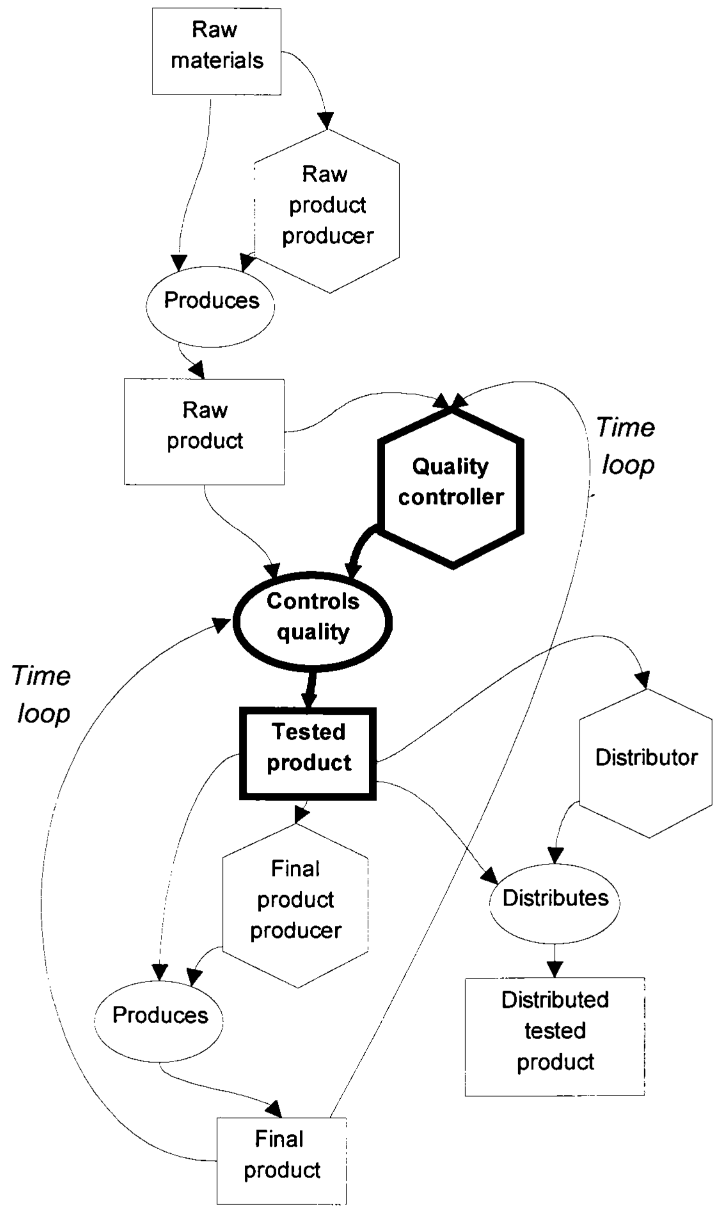


Figure 7: Mixing different levels of genericity.

Mixing several systemic levels in this way seems beneficial at first: it reduces complexity. After a closer look, however, the resulting model turns out to be ambiguous. The two arrows entering the hexagon "quality controller" are not a normal *"and"* (do not *in combination* cause the quality person to act intentionally), but rather are an *"exclusive or"*. *Either* the quality controller tests the raw product *or* the quality controller tests the final product. The same is true for the rectangle "Tested product"; its referent either goes to the distributor (in case of the tested final product) or to the producer of the final product (in case of the tested raw product). Another ambiguity is that the model does not exclude the possibility of an "eternal loop". The side effect of mixing different levels of genericity is that the model becomes underspecified. Additional representation conventions (like the bold printing) are required to keep the model unambiguous. Note that simply specifying the generic model part in two models, one for each referent, and representing both of them would obviate the ambiguities. An alternative route would be to abstract the two intentional activities "produce raw product" and "produce final product" to "produce product", which would result in a clear model consisting of only three intentionally acting actors, at a higher systemic level. Mixing several levels of genericity in one model should best be used only sparingly.

5.4.3.3 Multi-representation relations

In multi-representation model relations, one referent is being modelled from different points of view. This means that several models exist of the same referent. When looking at only one of these models (they are likely to be represented on separate pieces of paper), it is not clear that alternative models are existent as well. Therefore the title of alternative models should explicitly refer to the fact that the model is one out of many possible viewpoints (for example, "Model of indoor environment (viewpoint 1 of 3)").

A special case is the situation in which alternative models are the same for the larger part, but different for the smaller part. In these cases, it would seem to be efficient to represent the identical parts only once (to represent both alternatives in one model). Consider, for example, a plan for a large journey, of which a small part can take place either by taxi or by train. The two plans overlap to a considerable extent. Both alternatives can be inserted in the identical remainder of the plan. An example is presented in figure 8. Italic print indicates those parts of the model that are alternatives (see figure 8).

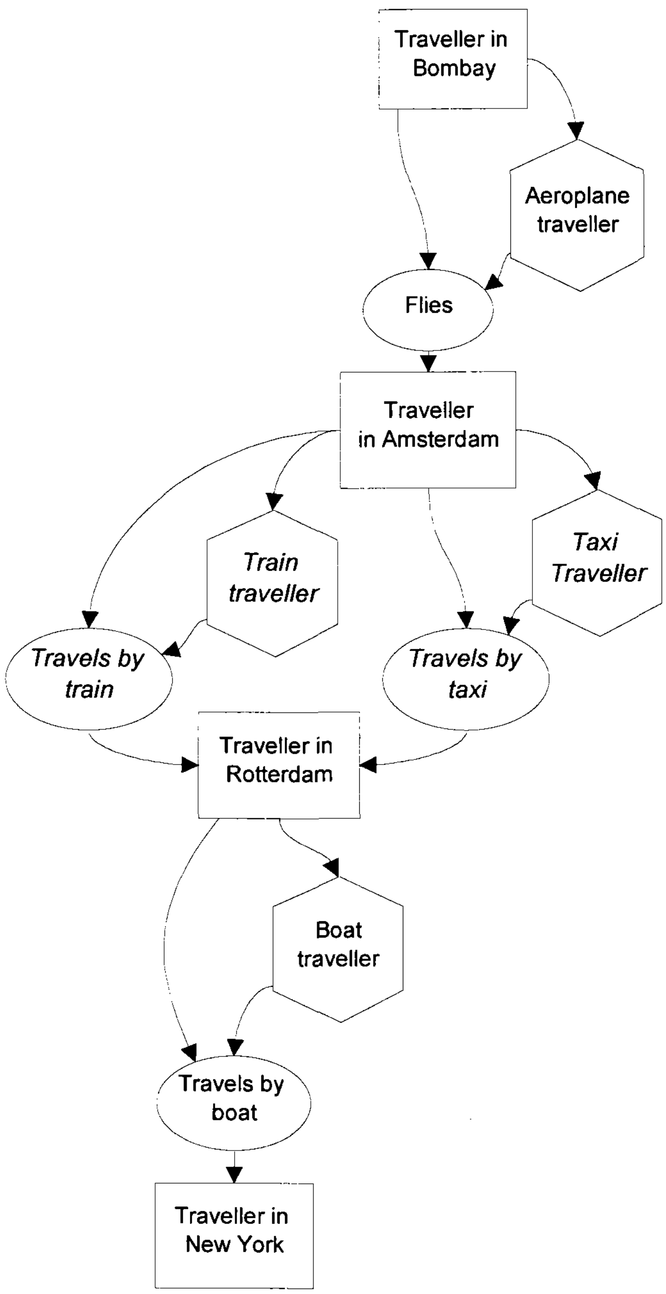


Figure 8: Combining different representations into one model.

After a closer look, this model also turns out to be underspecified (ambiguous). According to the normal convention, if several arrows enter a reference model primitive, they *in combination* cause the referent of this primitive (they form a logical "and"). However, the two routes are *alternatives*, and constitute an "exclusive or". The notation in italics for simple cases solves this ambiguity.

In order to remedy this situation, multiple representations as part of a larger model should be used sparingly. These situations are easy to avoid: option one is to take the notational overhead for granted and model both alternatives completely (their titles would have to reveal the presence of two viewpoints). Option two would be to abstract the two options to go from Amsterdam to Rotterdam to one option, that does not specify the specific means of conveyance (the model becomes multi-referent (generic), as it refers to two referents now).

5.4.4 Design modelling steps to change model relations (Step 4)

According to the theory of Chapter 4, the last step in designing *Trinity* is to design ways to change model relations. The representation conventions, presented in section 5.4.3, enable one to *represent* all three types of complex model relations (parallel, multi-referent and multi-representation). Representation conventions, however, do not support the very modelling *process* (i.e. the process of changing a model relation) itself. For this purpose, the theory of qualitative modelling processes (Chapter 4 of this dissertation) provides a typology of twelve primitive modelling steps. For convenience, the twelve modelling steps are presented once more in figure 9 and explained in table 1.

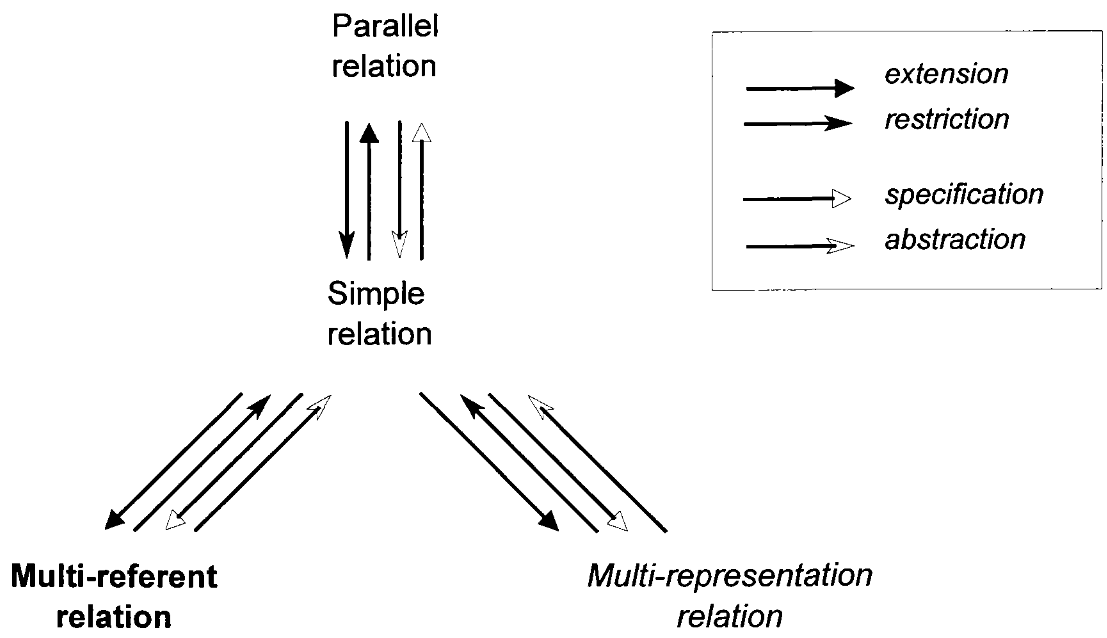


Figure 9: Modelling steps (the tetrahedron figure).

Table 1: A typology of primitive modelling steps.

Parallel modelling steps
(involving model representations consisting of >1 reference model primitive)

Parallel extension (adding a reference model primitive)
Parallel restriction (deleting a reference model primitive)
Parallel specification (modelling the same referent in more detail)
Parallel abstraction (modelling the same referent in less detail)

Referent modelling steps
(involving generic models, i.e. models with >1 referent)

Referent extension (adding a referent to a model)
Referent restriction (deleting a referent from a generic model)
Referent specification (subclassing a referent)
Referent abstraction (superclassing the referents of a generic model)

Representation modelling steps
(involving multiple viewpoints, i.e. referents with >1 model)

Representation extension (adding a viewpoint)
Representation restriction (deleting a viewpoint)
Representation specification (subclassing a viewpoint)
Representation abstraction (superclassing a set of viewpoints)

Below, the modelling steps of the *Trinity* language are presented.

5.4.4.1 Parallel modelling steps

Parallel modelling steps are perhaps the most intensively used modelling steps. *Trinity* provides a flexible library of parallel modelling steps. They change the number of reference model primitives in a model. For example, they add model primitives or detail a model primitive. The parallel library is elaborated below in detail.

Four different types of parallel modelling steps exist: extensions, restrictions, abstractions and specifications. Extensions and restrictions are inverse steps. The same holds for abstractions and specifications. Therefore we will design them in pairs. When applying a modelling step, the syntactical rules of parallel models should not be violated (see also section 5.4.3 and Appendix B). For this reason, parallel modelling steps will be designed in such a way that they are *syntax-preserving*: applying a parallel modelling step to a syntactically correct model will always result in a syntactically correct model. Preserving semantic correctness (and achieving pragmatic correctness, see also section 5.4.3 and Appendix B) is the task of the modeller.

Parallel extensions and restrictions

In semantic terms, parallel extensions increase, and parallel restrictions decrease the scope (the part of the three domains, distinguished by the *Trinity* principle) that is covered by a *Trinity* model. In syntactical terms, a parallel extension adds model primitives to, and a parallel restriction deletes model primitives from a model.

First, we will present some examples of parallel extensions/restrictions[47]. These examples will make clear the difference between two basic types of modelling steps: *static* and *dynamic* modelling steps (this distinction will appear again when discussing parallel abstractions and specifications). After that, a syntactical library of parallel extensions and restrictions will be presented.

When seeing a water tap, it is a plausible (theoretical) extension to model a water pipe as well (as this is normally the case), although this water pipe is hidden in the wall (and cannot be seen). It also is a plausible extension to assume the existence of water in this pipe. However, if the tap would not work, this last extension might be refuted (a restriction) on the basis of an empirical fact. These examples are *static*, as they do not change the episode covered by the model (figure 10).
When seeing water coming out of a water tap, it is a plausible step to model the process that opened the tap as well. This extends the episode of the model into the direction of the past. Also, it is a plausible step to expect someone closing the tap. This extends the episode of the model into the direction of the future. These examples change the model in terms of the time span (episode) it covers: they are called *dynamic* (figure 10).
Theoretical extensions are hypotheses. They are not "verified", like observations. Nonetheless, the modeller thinks these extensions are plausible, and he/she may be prepared to act accordingly. This is the reason why perspectives, for a large part resulting from theoretical extensions, may motivate and guide actions.

---

[47] Note that parallel extensions and restrictions, like all modelling steps, can be interpreted as intentional activities. They change the communication domain. The intention is triggered (acknowledged) by the initial model: this initial model, for some reason, is considered to be less appropriate than the final model. Modelling the modelling process, however, would imply a shift in problem context: rather than *the original referent* (i.e. the referent of the perspective being modelled), *the modelling process of this perspective* becomes the referent.

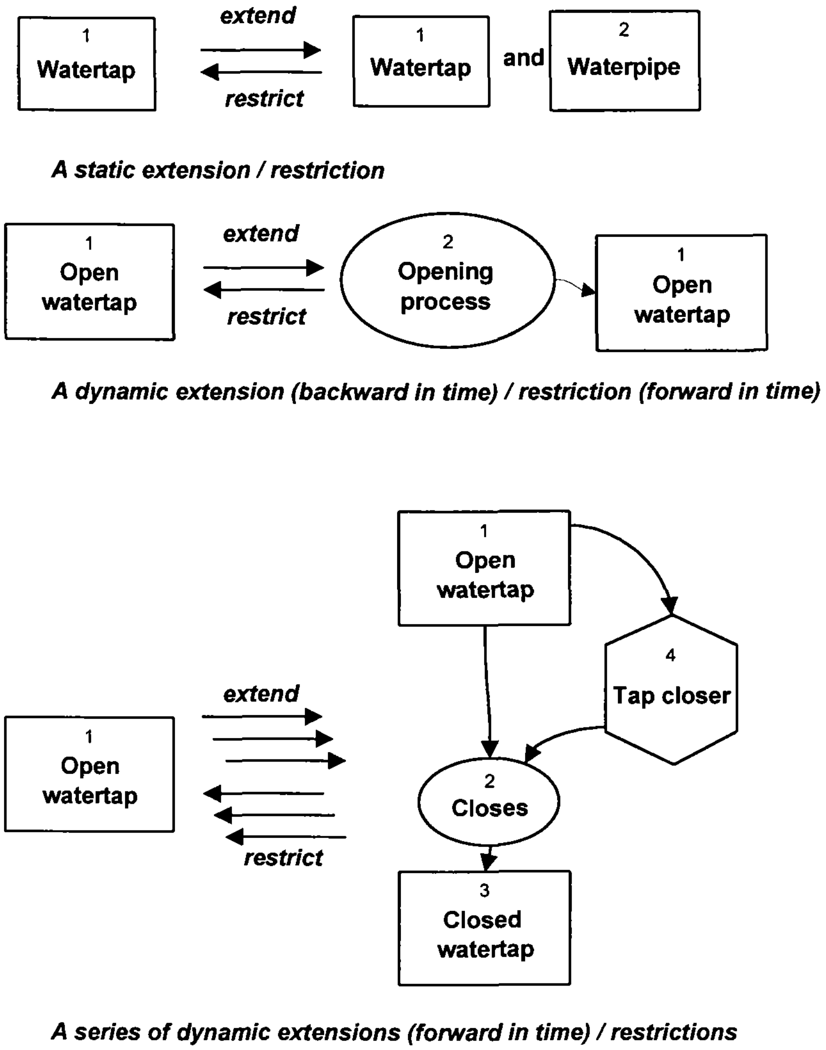


Figure 10: Examples of parallel extensions and restrictions.

A complete library (in syntactical terms) of the parallel extensions and restrictions of the *Trinity* language is presented in figures 11a-f. Note that figure 11f makes clear the way in which the knowledge domain differs from the other two domains: it is possible *to possess knowledge about knowledge*. This object-meta relation (the capability of reflecting and "processing" perspectives intentionally) does not exist within the other two domains (one can think about knowledge, but one cannot "physic" about matter). *Trinity* is special in this respect in that it explicitly enables one to model changes in the knowledge domain (see also the knowledge domain strategy in the battery example before).

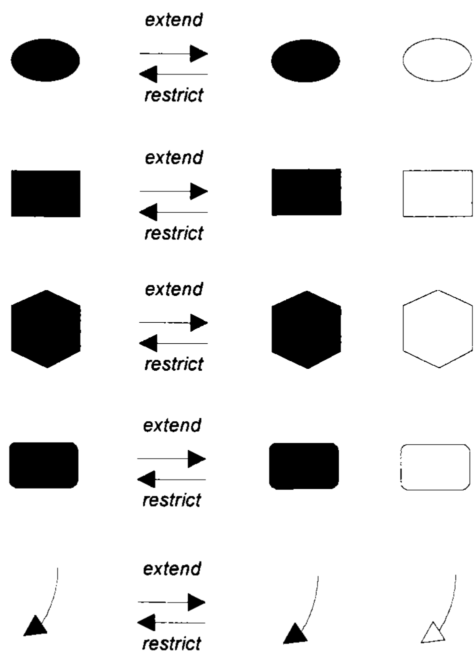


Figure 11a: Static parallel extensions and restrictions (the shading denotes that something is added/deleted).

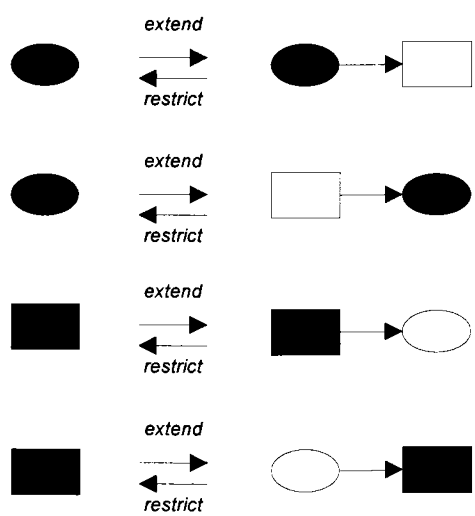


Figure 11b: Dynamic parallel extensions and restrictions in the physical domain.

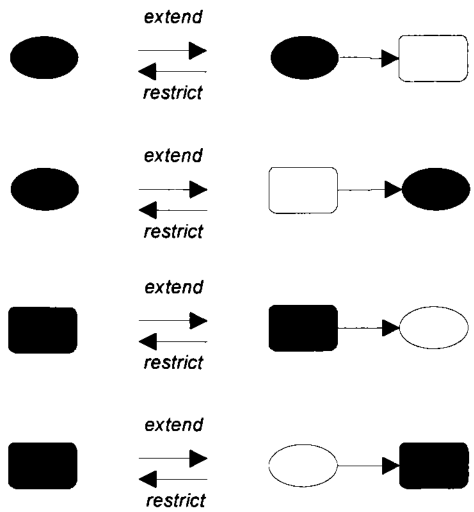


Figure 11c: Dynamic parallel extensions and restrictions in the communication domain.

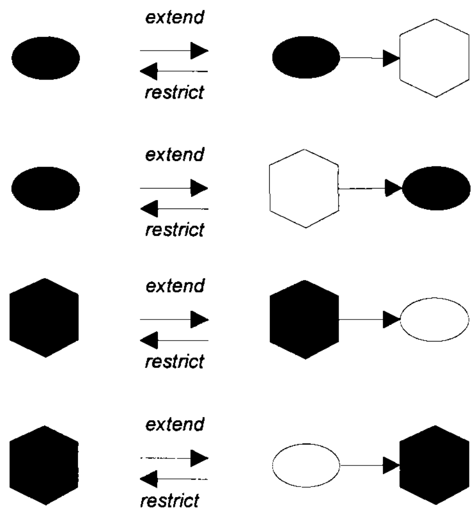


Figure 11d: Dynamic parallel extensions and restrictions in the knowledge domain.

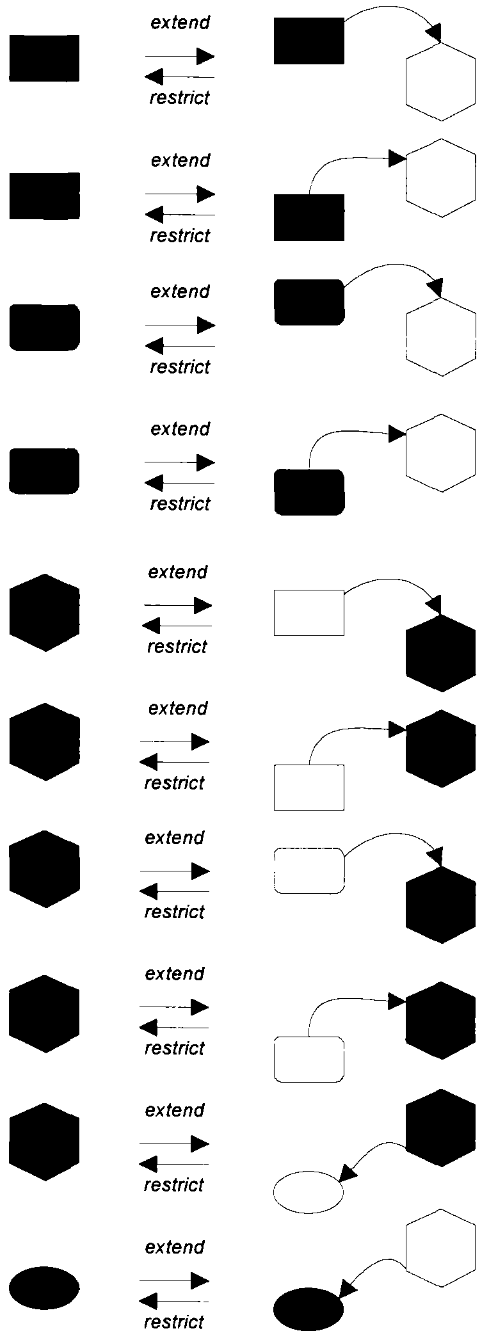


Figure 11e: Inter-domain parallel extensions and restrictions.

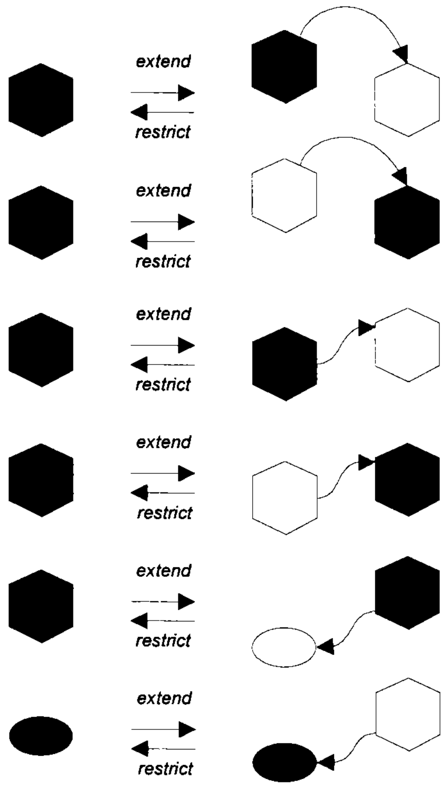


Figure 11f: "Knowledge about knowledge" extensions and restrictions in the knowledge domain.

Parallel specifications and abstractions

In contrast with extensions and restrictions, transformations (specifications and abstractions) obey a conservation principle: the *same* referent is modelled, but the level of detail has changed. A parallel specification turns a model primitive into a model of the same referent, but now it consists of several model primitives. A parallel abstraction turns a model consisting of several model primitives into a model of the same referent, but now it consists of fewer model primitives. Like any modelling step, parallel transformations can also be either empirical or theoretical. For example, when I see an orange, I may *assume* that it consists of a peel and parts, and that it has a specific taste. However, the presence of all these partial qualities are hypotheses: the orange might prove to be a very good plastic imitation, or taste rather unusual. I can also *experience* that this specific orange consists of parts, and has a specific taste[48].

[48] It is a remarkable paradox that, in a very strict interpretation, actually verifying all the qualities that make up the concept "orange" for a specific orange is likely to destroy this orange. This paradox is well in line with a pragmatic philosophical stance: something *is* what was or can be *done* with it.

In elaborating upon parallel transformations, first the general principle underlying *Trinity* transformations will be presented in terms of transformation rules. After that, some simple examples of transformations will be given. Following this, a syntactical library of primitive *Trinity* transformation steps will be presented.

When transforming a *Trinity* model, two[49] syntactical transformation rules must be obeyed; a specification rule and an abstraction rule:

When specifying a model primitive, the boundaries of the resulting complex model must be of the same type (domain) as the boundaries of the original simple model.

When abstracting a complex model, the boundaries of the resulting model primitive should be of the same type (domain) as the boundaries of the original complex model.

In addition, a semantic rule must be obeyed:

Both the more abstract and the more specific model should refer to the same referent[50].

Below, some examples of parallel specifications/abstractions are presented that illustrate these rules (figure 12). The first three examples are static: static specifications result in the distinction of concurrent parts that make up the original model; static abstractions do the inverse. Examples 4 and 5 are dynamic extensions/restrictions. Dynamic specifications increase the time resolution by introducing intermediate states and processes (the behaviour of the referent is modelled in more detail; concurrent model primitives are not introduced). Dynamic abstractions do the inverse. Examples 6 and 7 are dynamic, and operate on causations (arrows) instead of reference model primitives[51]. The last example shows the way in which a physical state is dynamically specified into a state transition that, subsequently, is made intentional.

---

[49] Actually it is one rule, worked out in two different directions: parallel abstraction and parallel specification are inverse operations.

[50] To be precise, the two models form a multi-representation model relation with the referent. It is, however, a specific multi-representation model relation, in that the more abstract model can be specified to the more specific model, and vice versa.

[51] Specifications that operate on arrows are consistent with the fact that a) states and processes are relative concepts and b) in many cases a causation is supported by a theory about the driving forces; the existence of such a theory implies that the arrow itself refers to a system. It is, however, only seldom seen in other qualitative modelling languages that relations between concepts can be understood as systems on a lower systemic level. In the last example of figure 12, the causation arrow that is specified refers to both a physical and an intentional causation (otherwise it would have been an extension).

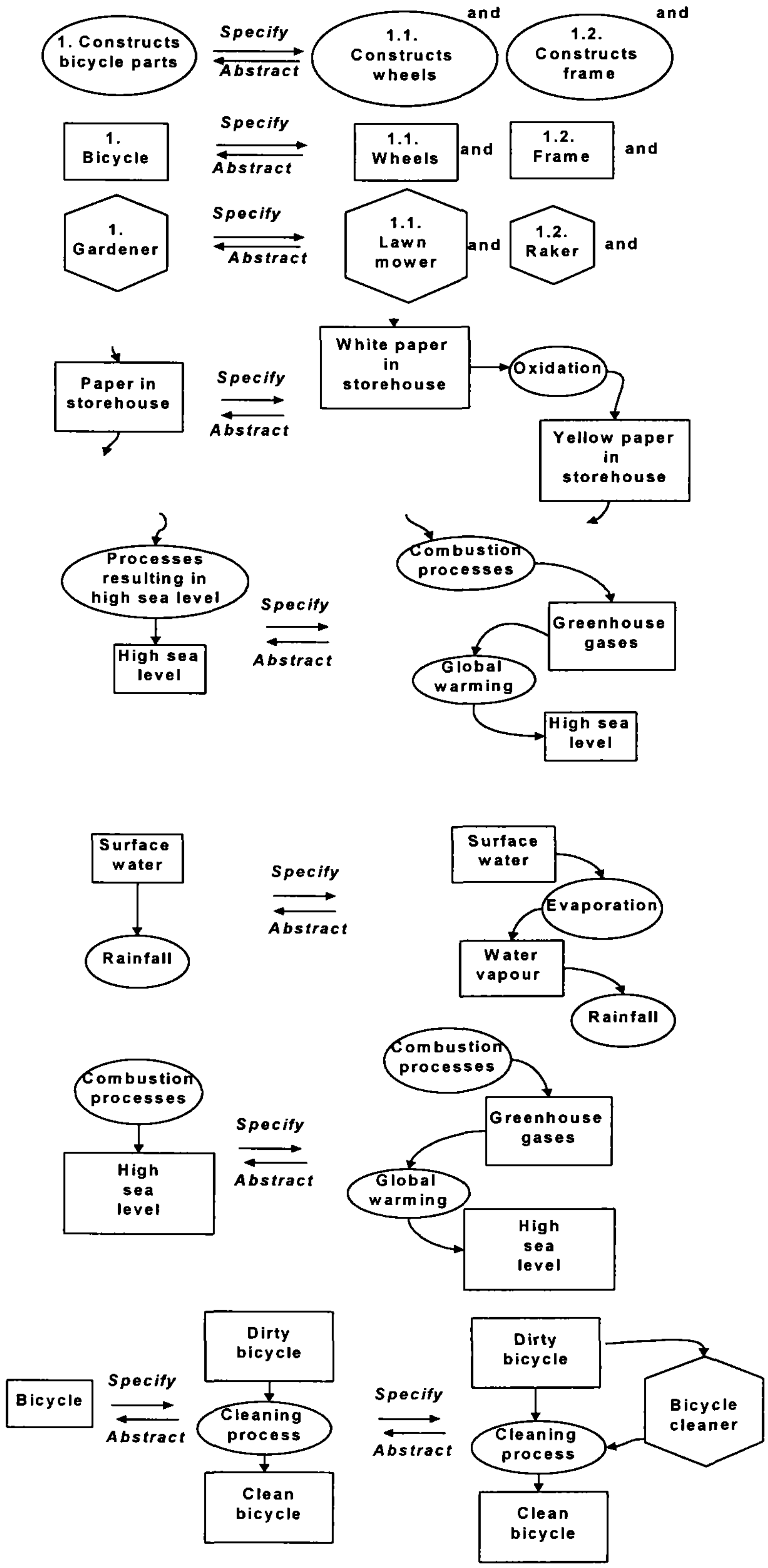


Figure 12: Examples of parallel transformations (both static and dynamic).

Below, a syntactical library of primitive parallel abstraction and specification steps is presented. The library is divided into static and dynamic transformations (figures 13 and 14 respectively). Dynamic transformations of physical domain states, knowledge domain states (perspectives) and communication domain states are presented in figure 14a. Dynamic transformations of processes in the different domains are presented in figure 14b. Two examples of dynamic transformations of arrows are presented in figure 14c. The first example specifies the arrow from the action to the "to be" state. A semantic example of this is the introduction of the car mechanic in figure 17. The second example in figure 14c specifies the evaluation arrow. For example, someone else observes the state directly resulting from the action, writes (the ellipse) some report (the rounded box, a communication domain state), which is evaluated by the original actor. A real-world example is when you are engaged in a difficult parking manoeuvre, and a bystander tells you whether your position is okay.

Note that all the second specification steps in figures 14a and 14b operate on causations (arrows) between a state and a process: this type of specification enables one to *re-interpret autonomous processes as intentional actions* (as now the ellipse is caused by a hexagon by means of a "start implementation" arrow). The inverse abstraction step enables one to *re-interpret intentional actions as autonomous processes*. For example, this type of abstraction is called for when the actor of concern does not take part in both the problem-solving process and the implementation process. He/she/they perhaps cannot be influenced (are out of reach), and therefore this specific action can be interpreted as an autonomous process: specifying the perspective is of no use.

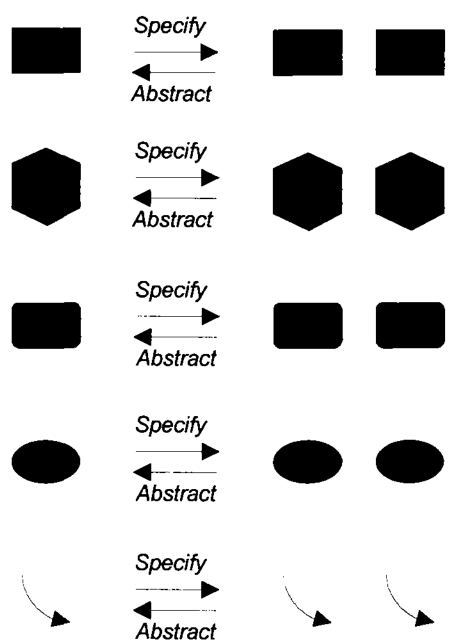


Figure 13: Static parallel transformations.

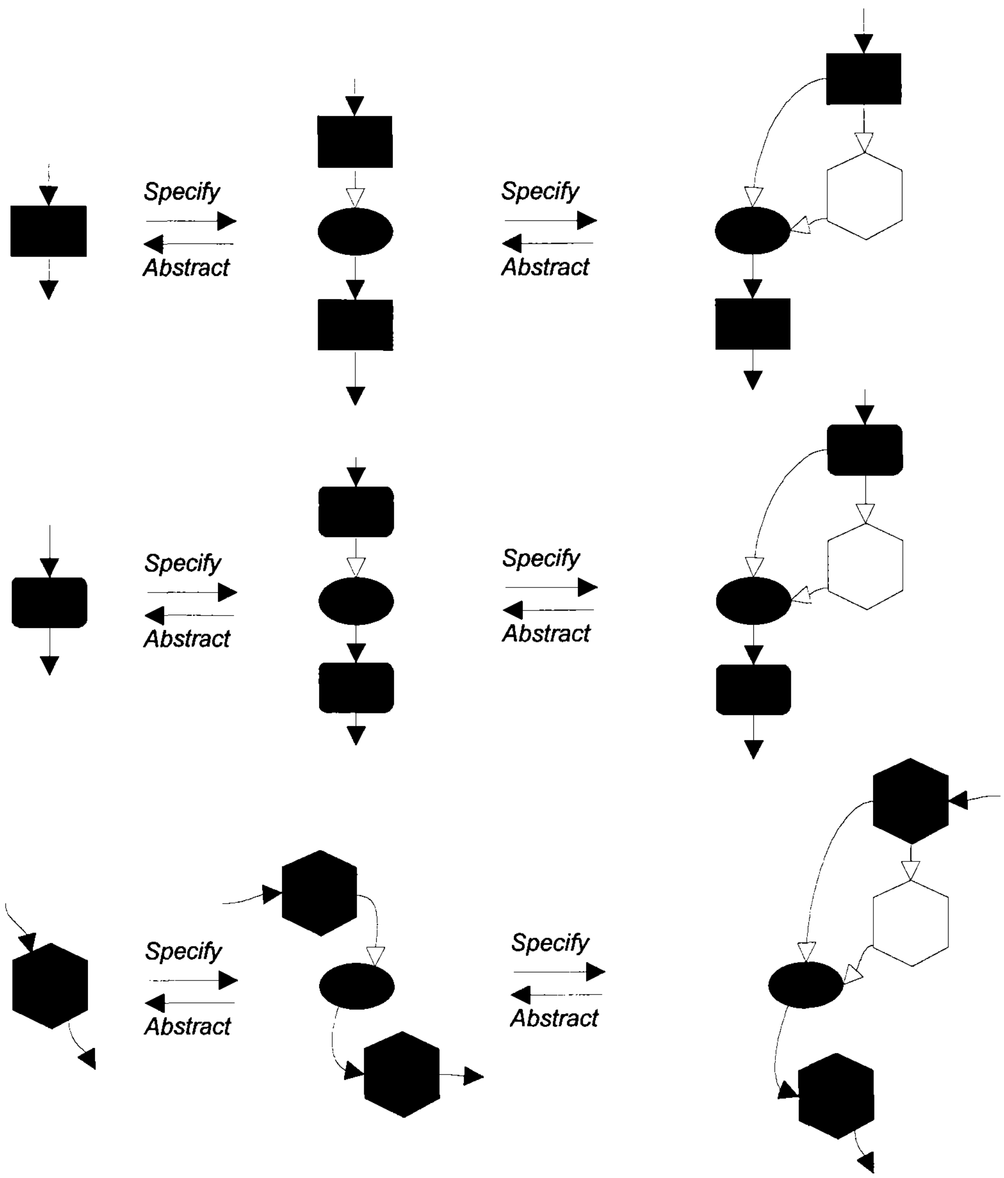


Figure 14a: Dynamic parallel transformations of domain states.

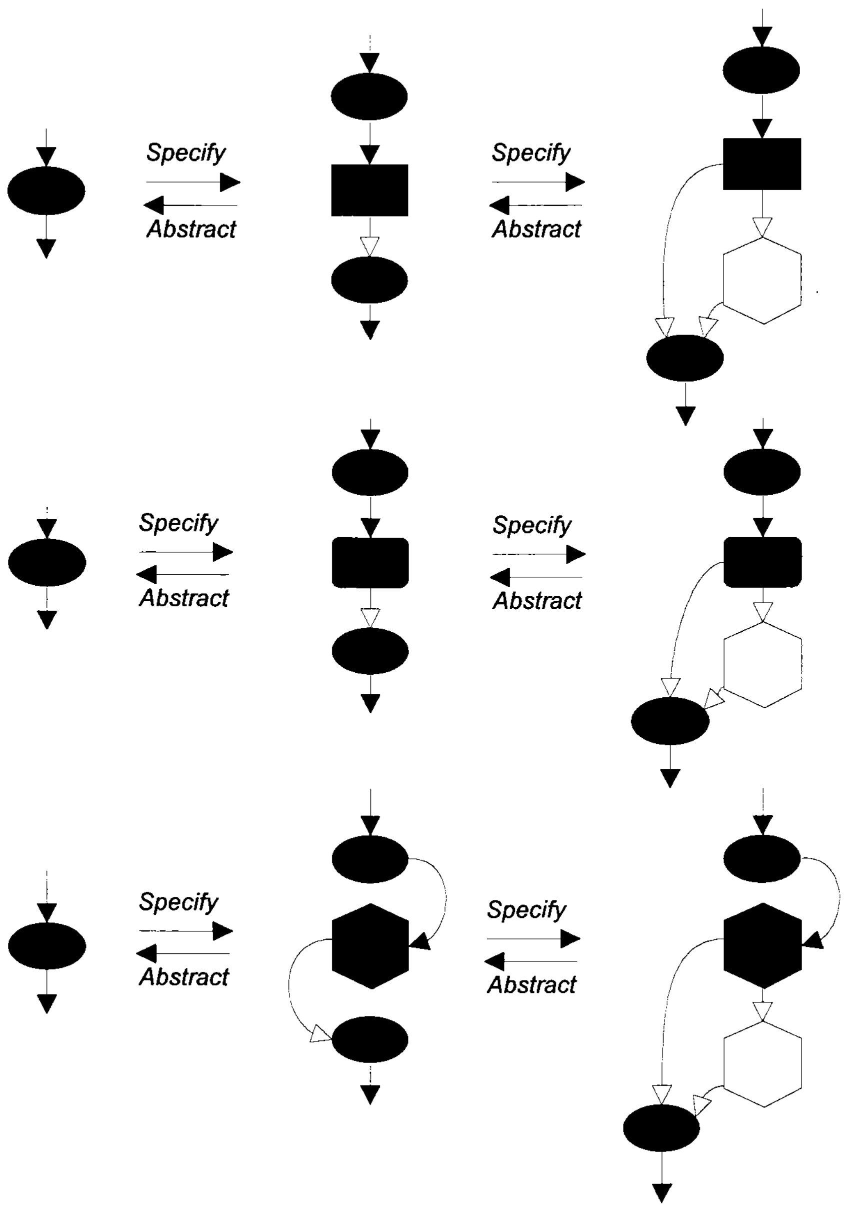


Figure 14b: Dynamic parallel transformations of processes.

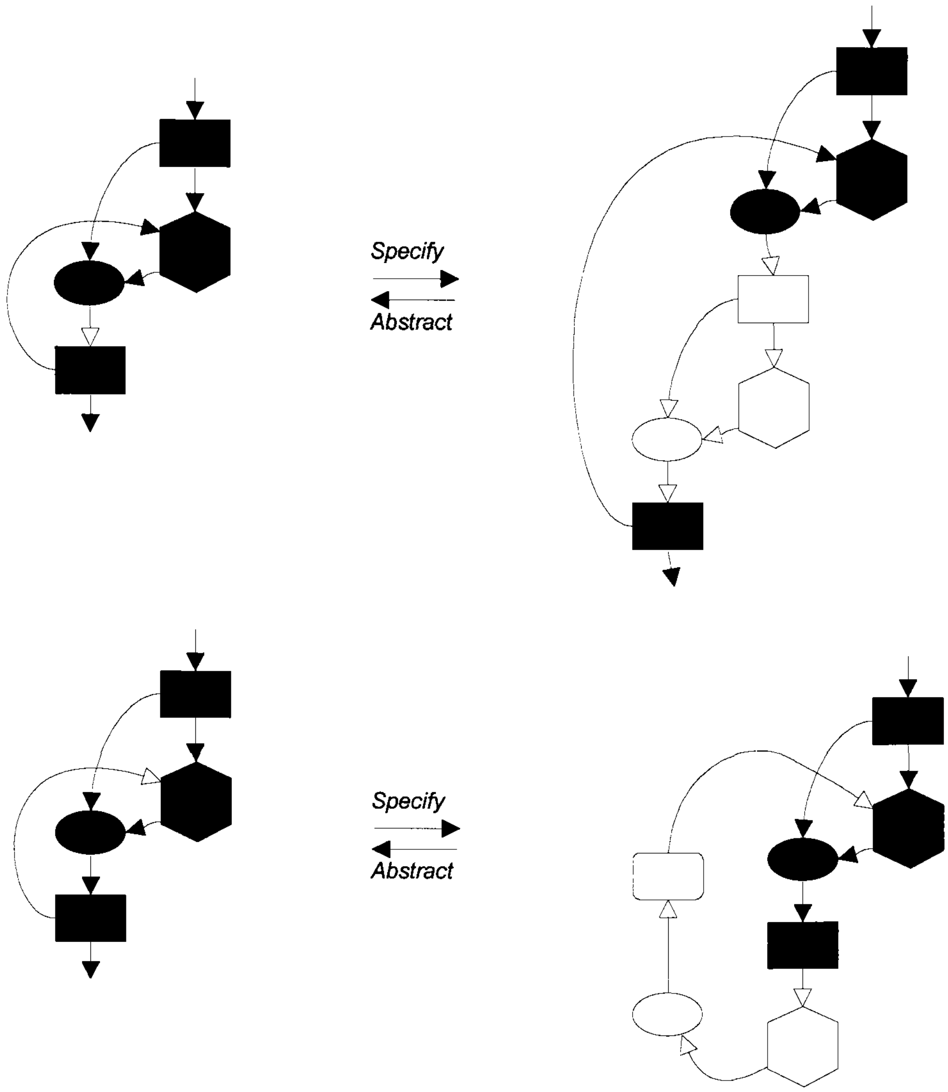


Figure 14c: Two examples of dynamic parallel transformations of causations (arrows).

Multi-level models

When abstracting or specifying a *Trinity* model, a model at a different systemic level is the result. An interesting distinction is whether the original systemic level is discarded, or maintains to be an integral part of a larger model. In the first case, the level of detail simply is *changed*. In the latter case, a systemic level is *added*: a model at two different systemic levels is the result (i.e. a **multi-level model**). In this case, both abstraction and specification increase the overall complexity (see also the discussion about balancing strategies in Chapter 4 of this dissertation).

5.4.4.2 Referent modelling steps

As was described in the theory of qualitative modelling processes, the results of *referent modelling steps*, normally, are invisible in the model (as they change the number of referents of a model, rather than the model representation). In the case that the model refers to more than one referent, this model is called *generic*[52]. We introduced the convention to represent partial generic models that are integrated in a larger model, in bold face (remember the quality control example). In addition, the title of a generic model should explicit the genericity.

Referent extensions and restrictions simply add and delete referents. Referent abstractions and specifications reduce and increase respectively, the number of referents that are modelled by the model, while covering the same scope as before (the set of referents is subclassed or superclassed, rather that extended or restricted).

5.4.4.3 Representation modelling steps

Representation modelling steps increase or decrease the number of alternative models of the same referent. In other words, a *Trinity* model is added or deleted (a piece of paper is added or deleted). We introduced the convention to represent partial alternative models in italics (remember the travel plan example). In addition, the title of an alternative model should reflect that it is only one of the available viewpoints.

In case of representation extensions and restrictions, an alternative point of view at the same systemic level is simply added or discarded. Representation abstractions and specifications introduce a new systemic level, covering the same scope as before. The original systemic level may either be maintained or discarded. In the first case, a multi-level model is obtained (i.e. a model that describes a referent at several, hierarchically related systemic levels. See also the discussion about overall complexity in Chapter 4 of this dissertation). In the second case, the model simply becomes more abstract or specific.

By means of using the *Trinity* modelling steps in combination, model relations of arbitrary complexity, number of viewpoints and genericity can be constructed[53].

---

[52] A very strict interpretation of a generic model is a model that models all the members of a "genus". However, when interpreting the referents of a model as a genus, the term generic model seems to be appropriate.

[53] Note that this, once more, is an echo of KDS (Chapter 3 of this dissertation). Complexity, number of viewpoints and genericity as mentioned above correspond with complexity, diversity and adherence, respectively, which constitute the three dimensions of KDS. See also the footnote in Chapter 4 with respect to the relation between the three types of complex model relations and the axes of KDS.

5.5 *TRINITY* MODELLING STRATEGIES

In this section, several *modelling strategies* will be presented. Modelling strategies are typical, recurrent sequences of modelling steps that, in combination, enable one to reach an overall goal (for example, to trade complexity for breadth of scope). They facilitate a method of thinking and talking about modelling processes (i.e. a conceptual vocabulary) that is less fragmentary and more in terms of modelling goals than modelling steps in isolation do. Modelling strategies were introduced and discussed in Chapter 4 of the theory part of this dissertation. At that point, they were part of a generic theory. In this section, parts of this theoretical framework will be applied to the *Trinity* qualitative modelling language (this is possible as *Trinity* is designed in compliance with the theory of Chapter 4).

Four[54] classes of strategies will be discussed: 1) parallel strategies; 2) multi-referent strategies; 3) multi-representation strategies; and 4) compound strategies.

5.5.1 Parallel strategies

Parallel strategies are sequences of parallel modelling steps, i.e. parallel extensions, restrictions, abstractions and/or specifications. Parallel strategies are characterised by the fact that both the number of *viewpoints*, with respect to the referent and the level of *genericity* of the model remain the same. The only things that change in a parallel strategy are the *systemic level* of the model (the granularity, the level of detail) and the *scope* (the coverage) of the model. In terms of the library of modelling steps presented in section 5.4: the *systemic level* is changed by means of *transformations* (abstractions and/or specifications); the *scope* is changed by means of *extensions* and/or *restrictions*. In figure 15, several examples of parallel strategies are explained visually (for a theoretical discussion of parallel modelling strategies, see Chapter 4 of this dissertation). Some of them will be explained below. After that, examples of the *Trinity* modelling processes will explain the parallel strategies further.

Explanation of parallel strategies

Subtypes of parallel strategies are *expansion strategies* and *reduction strategies*.

Expansion strategies are strategies in which an initially simple and possibly incomplete model is turned into a more complex model. Typically, they are used if a model lacks

---

[54] As *Trinity* is designed according to the theory as presented in Chapter 4, ***every*** modelling strategy described in Chapter 4, as well as combinations of these strategies, can be used within the *Trinity* modelling paradigm. However, here we will focus on three important *basic* strategies, and suggest a *compound* strategy that may be used in very complex cases. One reason for this limitation is that the overall number of possible strategies is rather large and may be confusing at first. Another reason is that the three strategies to be discussed here are the ones that will re-appear in **Part IV: Experiments** of this dissertation. Experienced users of *Trinity* may want to experiment with other strategies, though.

either scope or detail. Reduction strategies do the inverse: an initially complex model is turned into a simpler model.

Two different varieties of expansion strategies can be distinguished: *extension strategies* and *specification strategies*.
In an extension strategy, model primitives are added to an incomplete initial model by means of parallel extension steps. The scope is increased. The strategy is "inward-out": the extension strategy resembles a growing *Lego* model of a house.
In a specification strategy, a global model is detailed by means of a sequence of parallel specification steps. Detail is added (the same referent is modelled at a lower systemic level). The strategy has a top-down nature.

Likewise, two different reduction strategies can be distinguished: *restriction strategies* and *abstraction strategies*.
In a restriction strategy, parts are deleted from a complex model. The scope becomes smaller. The strategy is outward-in.
In an abstraction strategy, parts of a complex model are abstracted, and this procedure is repeated. The scope remains the same, but detail is covered up (the systemic level becomes higher). The strategy is bottom-up.

Mixtures of abstraction steps and specification steps constitute *a transformation strategy*: during the modelling process the *same* referent (i.e. the same scope) is modelled on *different* systemic levels (levels of detail). Typically, a transformation strategy is used if the referent in principle is known (demarcated), but the adequate systemic level required to model it (i.e. a level that suits some purpose, that allows for intentional acting) is not yet known. Alternating abstractions and specifications allows for finding this adequate level of description.

Mixtures of extension and restriction steps constitute a *building blocks strategy*: model primitives are added and deleted. This strategy resembles playing with *Lego* blocks. A building blocks strategy is typically used if the level of detail of the model is known, but the exact scope of the model is unclear. Extensions and restrictions in combination enable one to explore the scope.

A more subtle strategy is the *(obtain) bird's eye view strategy*: this is a mixture of extension steps and abstraction steps. *Both* the scope *and* the systemic level change: the scope changes because of the extension steps; the systemic level changes because of the abstraction steps. By alternating them, the level of detail is traded off for scope (hence "bird's eye view"). A rationale for using a bird's eye strategy may be that only a certain amount of complexity can be understood within a certain amount of time: a "cognitive threshold" exists. A bird's eye strategy allows for keeping the overall structural complexity of the model the same (i.e. obeying the cognitive threshold), while covering a larger scope. However, the other side of the coin is that detail is discarded. Strategies that keep the overall complexity of a model at the same level by means of such a trade off, we call *balancing strategies*. A complex balancing strategy will be presented in the section dealing with complex modelling strategies (section 5.5.4.1). A *focus strategy* is the inverse of a bird's eye strategy: scope is traded off for level of detail.

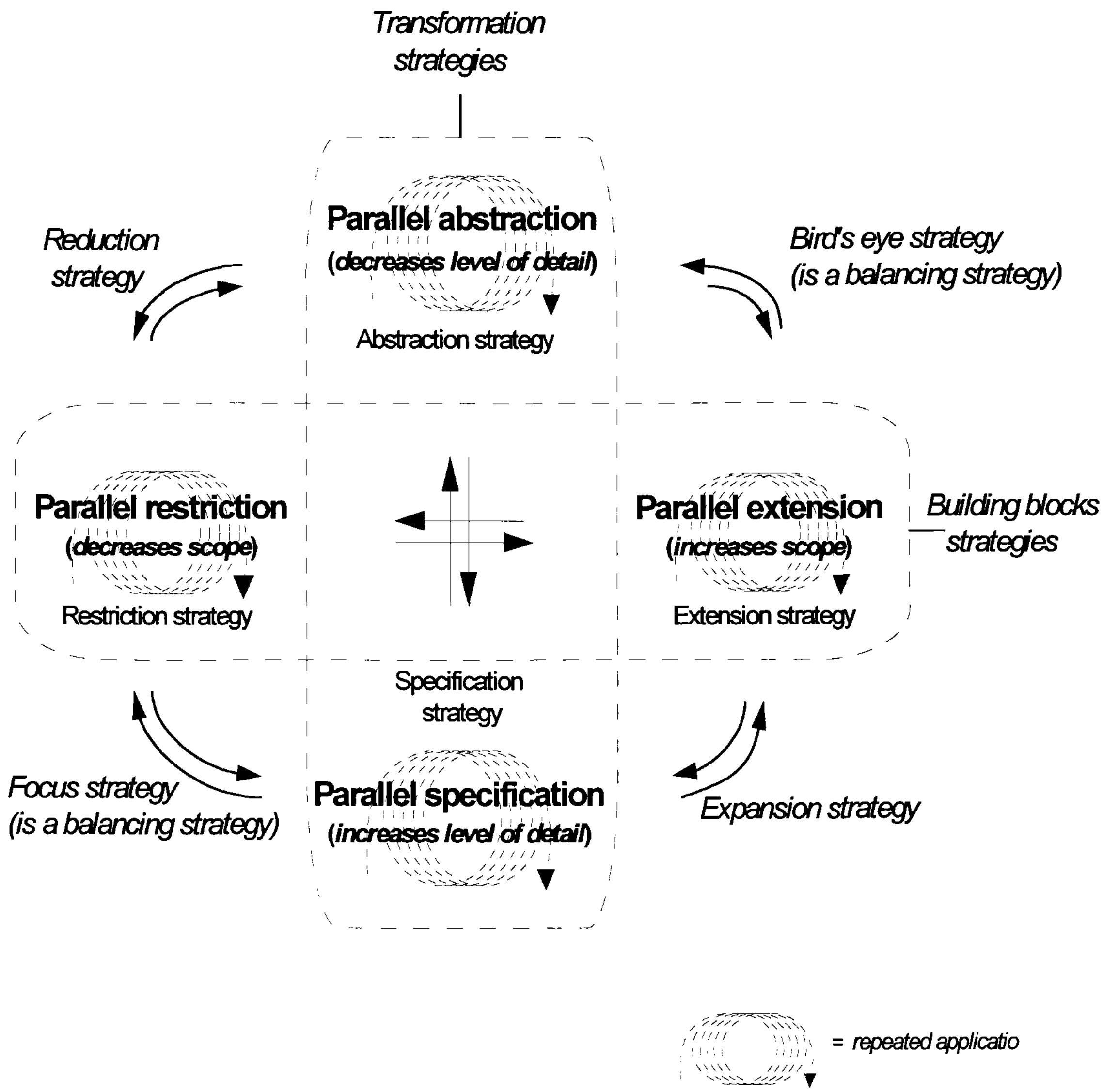


Figure 15: Parallel *Trinity* strategies.

Examples of applying parallel Trinity strategies

The first examples are simple sequences of two specification steps. They, in combination, constitute a simple specification strategy. Note that the upper example is in agreement with the first sequence of figure 14a, and the lower example is in agreement with the second sequence of figure 14b (the syntactical parallel library).

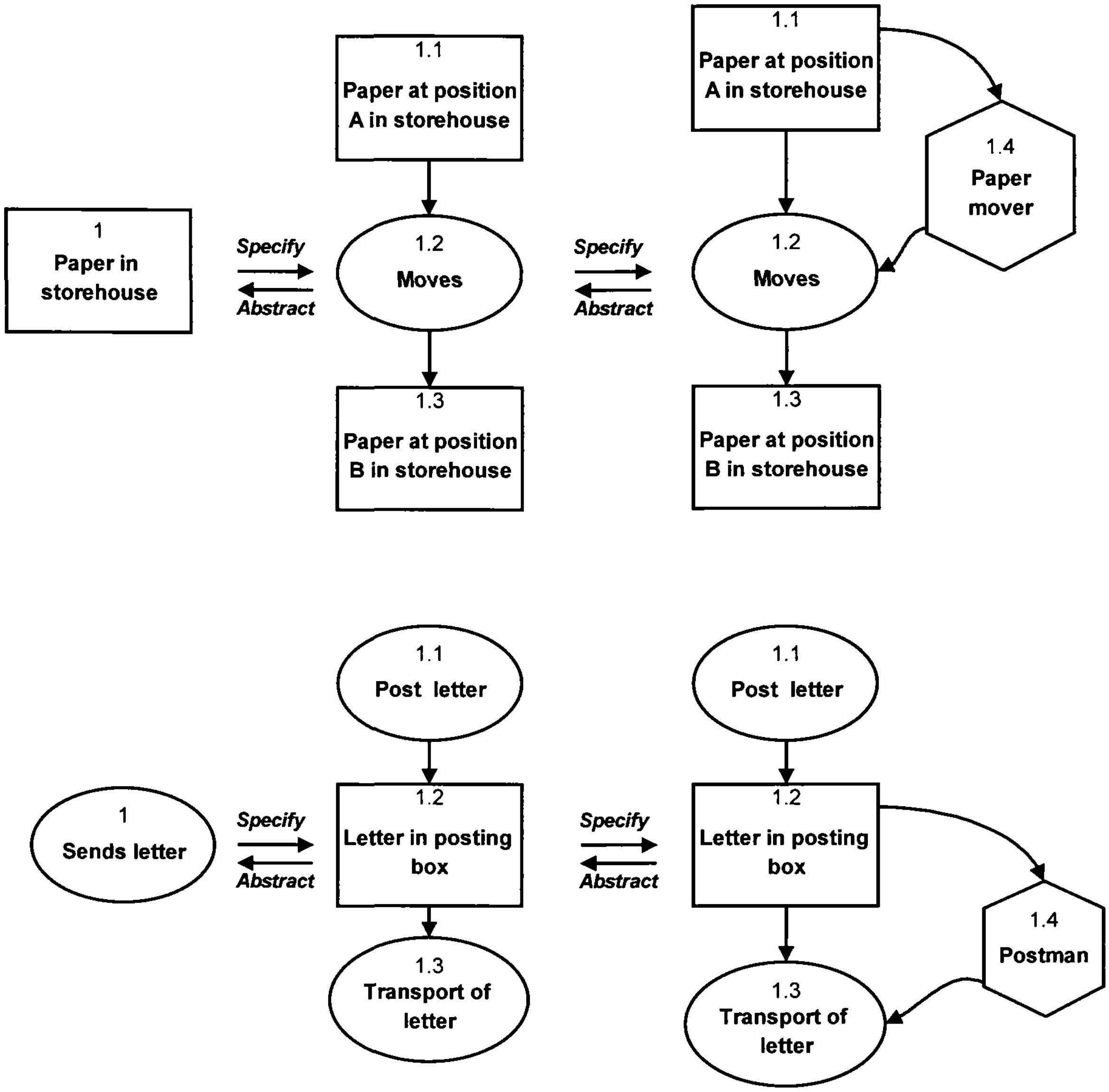


Figure 16: Sequencing parallel specification steps results in a parallel specification strategy. The inverse is a parallel abstraction strategy.

The second example (figure 17) is a more complex parallel specification strategy. The example uses modelling steps of the transformation library (figures 14). Note that the example starts with only one rectangle, and is subsequently specified into a more complex model consisting of several model primitives. However, all intermediate models refer to the same referent (this is a requirement of a specification strategy). In step 3, an arrow (a causation) is specified into an intentional activity. Note that this step might have been worked out in two sub-steps, the second step introducing (specifying) the mechanic hexagon. This shows the way in which modelling steps can be used recursively (i.e. on different levels of detail) in a modelling process. Interpreting figure 17 in reverse order results in an abstraction strategy.

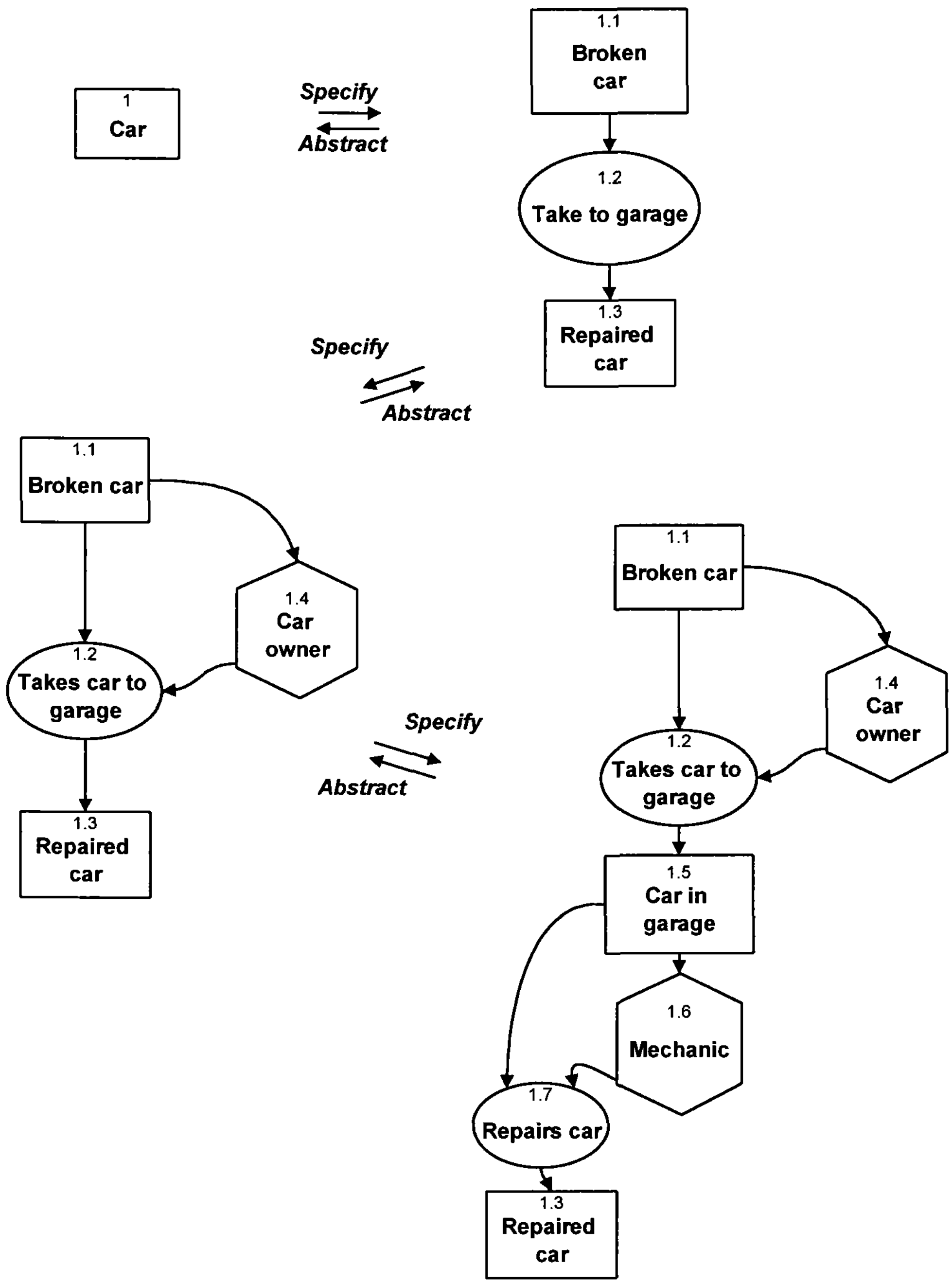


Figure 17: A parallel specification strategy consisting of several steps (the inverse is a parallel abstraction strategy).

The third example presents a transformation strategy: in the middle of a sequence of specifications an abstraction is present as well. This example, concerning a renovation process of a fence, shows the way in which a physical state can be specified into a model referring to a renovation process. As all the steps are syntax-preserving, the strategy as a whole is also syntax-preserving.

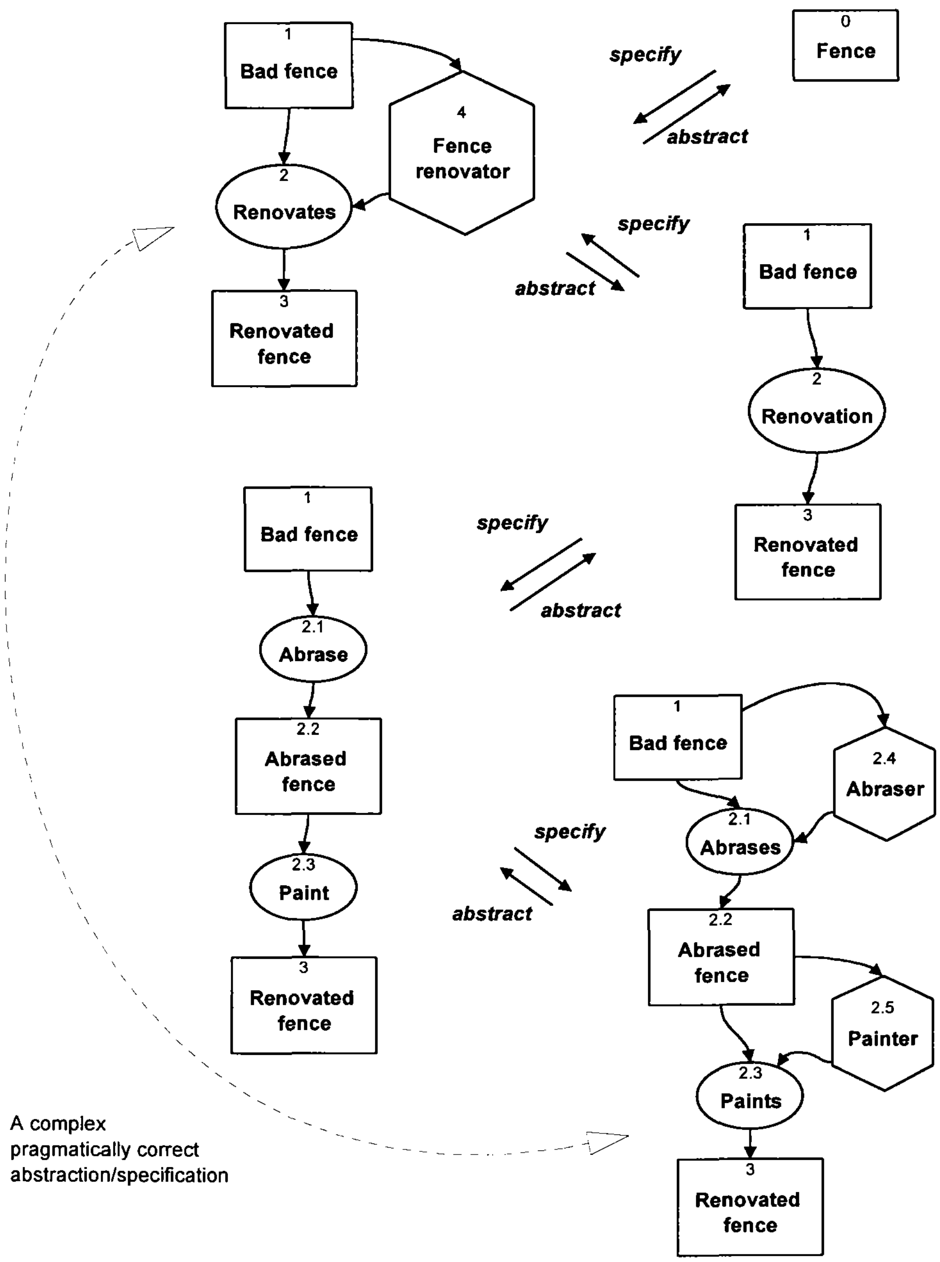


Figure 18: A transformation strategy.

The fourth example (figure 19) presents a modelling process that results in the same model as the result of figure 18, but now an extension strategy[55] is applied during the construction. The modelling strategy uses some of the dynamic modelling steps of the library presented in figures 11.

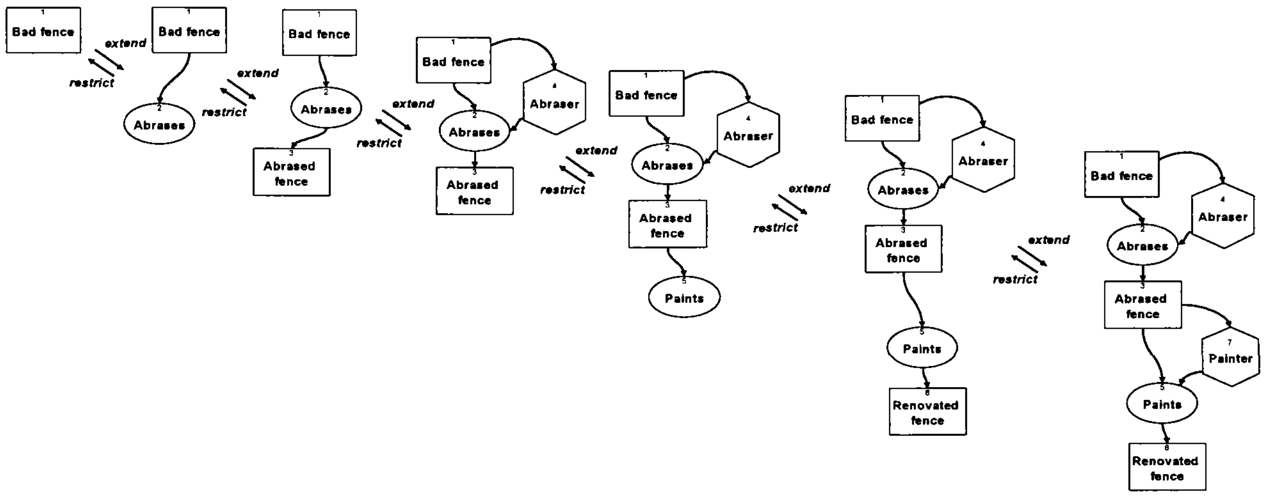


Figure 19: A dynamic extension strategy (inverse: a dynamic restriction strategy).

The fifth and last example (figure 20) shows the application of a simple bird's eye view strategy. A model is abstracted (see the transformation library in figures 14) and extended (see the extension/restriction library in figures 11). Note that the scope of the initial and final models is different (the scope increases), and that the number of model primitives in the initial and the final model is the same (i.e. it is a balancing strategy). This implies that the final model is at a different systemic level.

[55] Steps 3 and 6 in the process resemble a specification strategy. However, it is assumed here that the arrows from rectangle to ellipse refer to a purely physical causation, and that the causation via the perspective is *added*, rather than that the arrows from rectangle to ellipse encompass both the physical and the intentional part of the causation, and that the causation via the perspective is *specified*. However, as arrows derive meaning from the reference model primitives they model, this cannot be derived from the model. Only the modeller can tell.

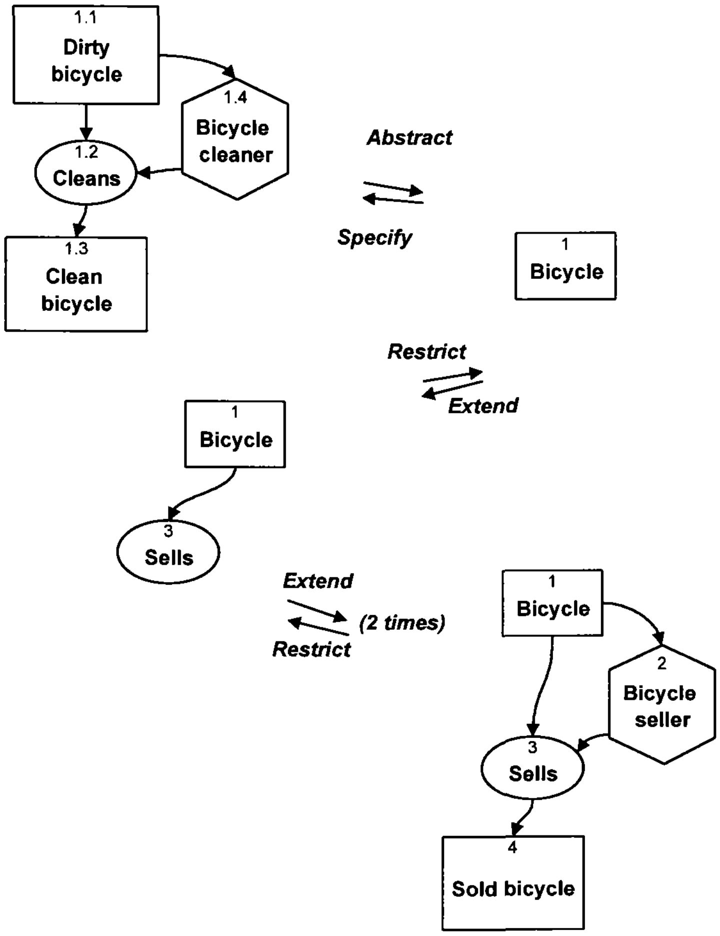


Figure 20: A simple bird's eye strategy.

Figure 21a shows the way in which a process of perspective modelling (i.e. a process that supports a problem-solving process) can be interpreted as a parallel specification process of the three model primitives that refer to the environment of the problem solver. This way of looking at perspective modelling can be recognised as a top-down approach. Figure 21b shows another way of looking at the problem-solving process: here an extension strategy is used (i.e. an inward-out approach). However, this strategy should also result in a model of a perspective, of which the three parts can be abstracted to the environment of the problem solver. This is a universal requirement: in order for a model to be a pragmatically correct model of a perspective, at the highest level of abstraction this model must refer to the situation "as is" (a domain state); the intended action (an ellipse); and the situation "to be" (again a domain state of the same domain) of the problem solver. The script intervenes with ("branches off") the normal flux of events in the "as is" model, and causally results in another flux of events in the "to be" situation (see also the battery example presented earlier).

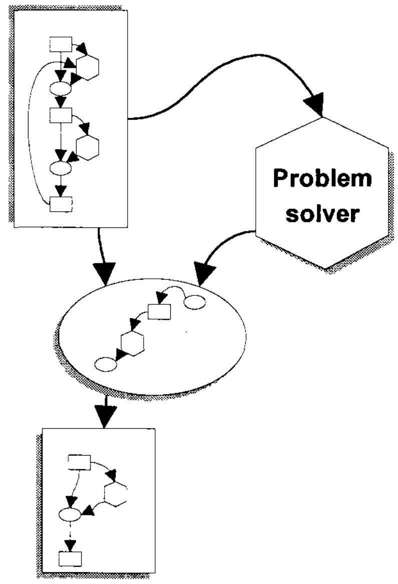


Figure 21a: A specification strategy in modelling a perspective: the perspective results from specification of the three reference model primitives constituting a model of the problem solver's environment.

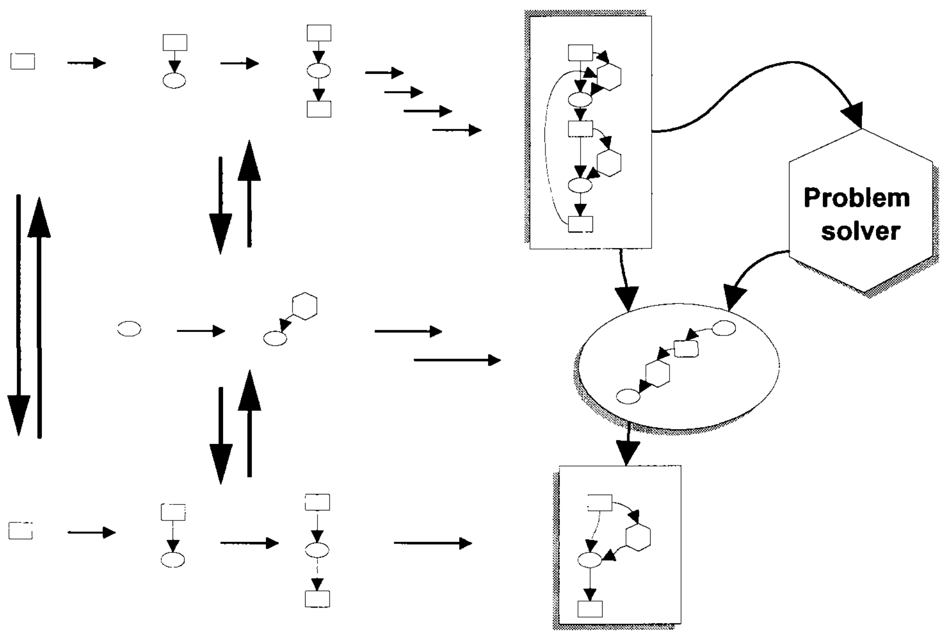


Figure 21b: An extension strategy in modelling a perspective: the resulting model of a perspective at the highest level of abstraction should refer to the environment of the problem solver.

The vertical arrows in figure 21b indicate that analysis, script synthesis and prediction do not necessarily take place in sequential order, but may influence each other, resulting in a possibly complex mixture (see also Chapter 2 of this dissertation).

5.5.2 Multi-referent strategies

In the case of a multi-referent model relation, the model is *generic*. A generic model is a model that represents the members of a genus (a type, a class), rather than one unique referent: the *same* model can be used on *several* occasions. This is beneficial, as it is not required to construct a new model each time from scratch. A generic model can best be understood in terms of a trade off: the model should be sufficiently general to cover a relatively large class of instances, but at the same time be sufficiently specific to be of any support to a modeller (the ultimate generic model would be next to trivial). Examples of the use of generic models can be found in different fields of human endeavour. Examples are: the KADS interpretation language [Breuker and van de Velde (1994)], Mintzberg's [1990] archetypical organisation models, the five generic Systems Dynamics models [Senge et al. (1994)]), and Chandrasekaran's [1988] generic inference models. Also Newton's law (F=m*a) is an example of a generic model: it can be reused in several situations (is multi-referent).

Multi-referent modelling steps change the genericity of a model. As a consequence, a multi-referent strategy is a modelling strategy that utilises changes in the genericity of a model. Therefore, a nickname for a multi-referent strategy is a *generic model strategy*.
During the *construction* of a generic model, several referents are recognised as the members of a more general genus (class). The shared characteristics of members of this class are represented by the generic model. This highly reduces complexity. Genericity increases. During the *use* of a generic model, a referent is recognised (distinguished) in terms of the elements of the generic model: this enables classification of this referent as a (new) member of the genus. Genericity decreases (see figure 22).

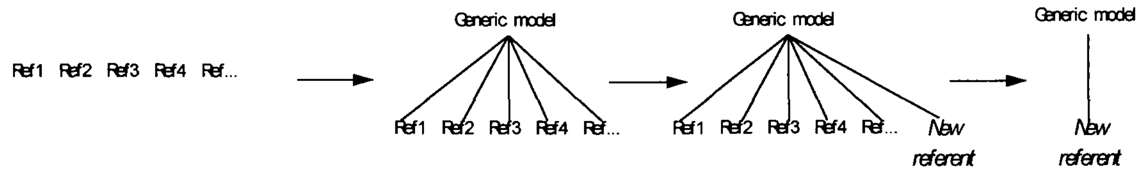


Figure 22: Construction and use of a generic model.

The most important requirement for successful use of generic models is the availability of a sufficiently large library of generic models, a library that covers the whole domain of interest. Typically, the first step of a modeller is to *select* a generic model from this library (or perhaps to *combine* several of them). After this, the generic model is used as a single-referent model of the referent of interest, and is tuned further. A good example (in the field of knowledge engineering) of a library that meets this requirement to a considerable extent is the KADS library of models of human expertise referred to above [Breuker and van de Velde (1994)].

Trinity is especially suited for using multi-referent strategies (i.e. to construct and use generic models). This is so because the construction of models at different levels of abstraction is explicitly supported by means of transformation steps (abstractions and specifications), and the more abstract a model becomes, the more likely it is that its genericity increases. As a matter of fact, several highly generic *Trinity* models have already been presented. For example, the model of an intentional activity in the physical domain is highly generic indeed, as are the right-hand sides of the examples in figure 14c (they model situations of co-operation).
Below, another generic model is presented; a model that is especially useful when operating in *environmental* problem contexts. It is called the **minimal environmental situation of concern** (figure 23).

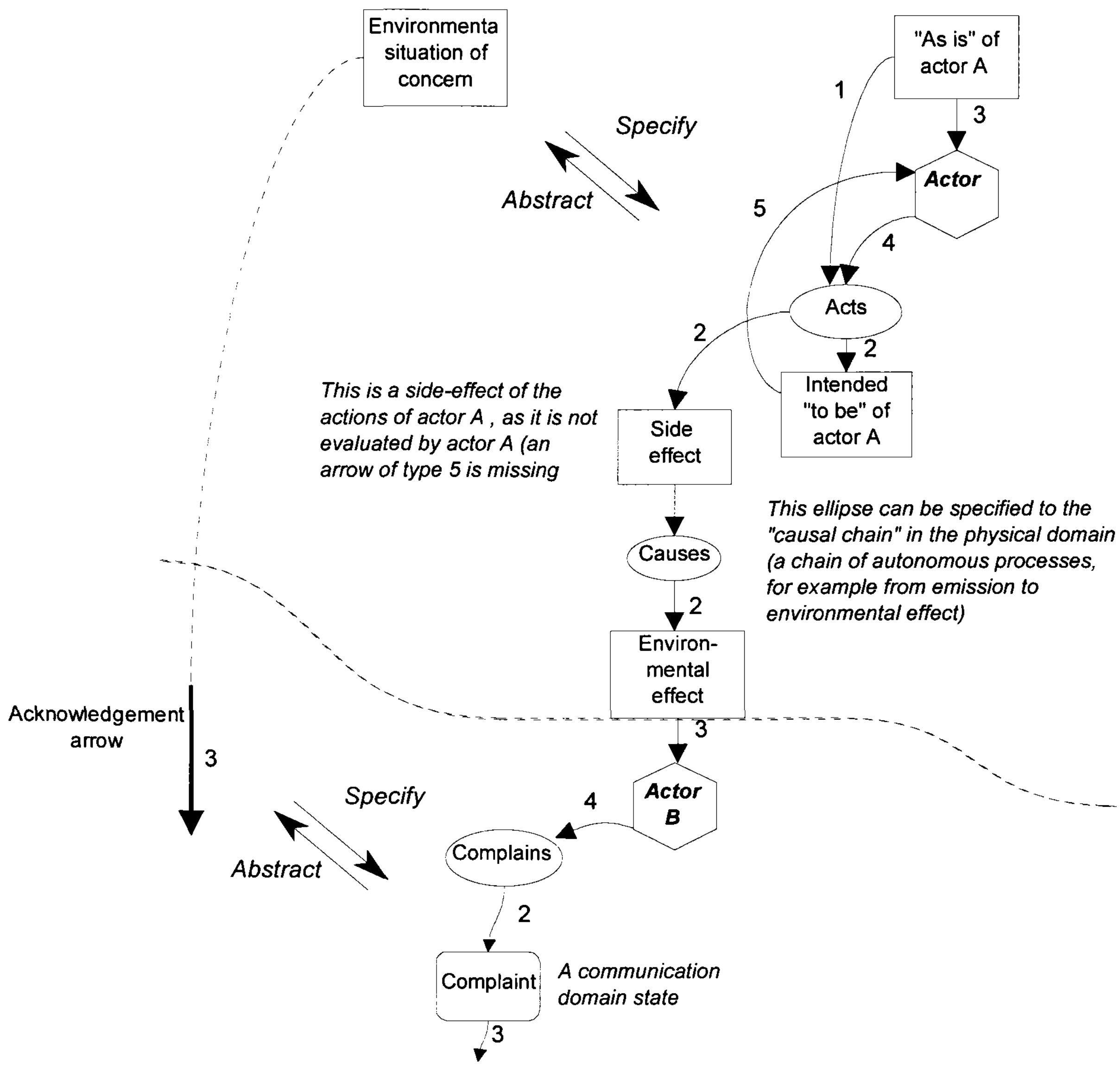


Figure 23: The minimal environmental situation of concern.

The **minimal environmental situation of concern** is a generic model containing the essence of many environmental problem situations: two actors are present, coupled by an autonomous, non-intended process in the physical domain (which may be specified into a chain). The first actor intervenes in the physical environment. As a non-intended side-effect, this physical environment is disturbed. This, in combination, models the situation of concern. The second actor observes the effects of the side-effect, and sends a message (or, more general, communicates his/her concern). The numbers next to the arrows highlight the five basic types of connections between reference model primitives.

Environmental problems concern side-effects arising in the physical environment, and are situated in a social context. All these elements are reflected by the minimal environmental situation of concern. In part V: Experiments, an example will be presented of the use of this generic model as a first step in obtaining a more detailed model of indoor environmental situations of concern.

As mentioned before, a trade off exists between level of reusability and heuristic value (support) of a generic model. As the *Trinity* language enables one to model on several levels of detail (by means of abstractions and specifications), generic models can also exist on several levels of detail. Representation specifications of generic models can still be generic and become part of a library as well. Their heuristic value (their support) increases, but their reusability necessarily decreases. In the Experiments part of this thesis, generic models will be presented that exhibit a larger heuristic value and a smaller level of reusability. For example, generic "as is" models of specific types of environmental problem situations, typically, are specifications of the **minimal environmental situation of concern**. A case in point is the (generic) indoor environmental model referred to above.

Three types of **generic scripts** can be distinguished: scripts directed at changing the *physical domain*, scripts directed at changing the *communication domain* and scripts directed at changing the *knowledge domain.*
Generic models of scripts are presented in figures 24a and 24b (the scripts may be abstracted even further into one ellipse by means of a parallel abstraction). A generic communication domain script is not presented, as it is like a physical domain script in which the rectangles become rounded boxes.

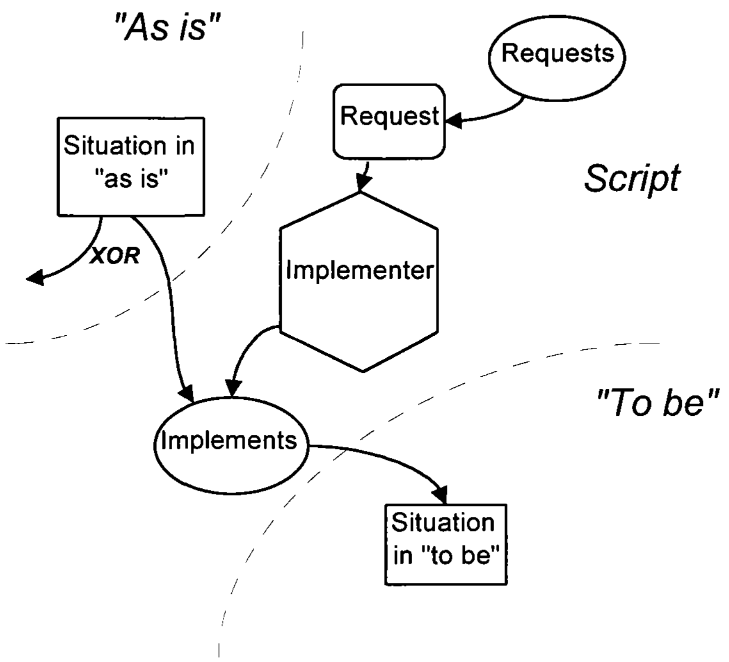


Figure 24a: A generic physical domain script.

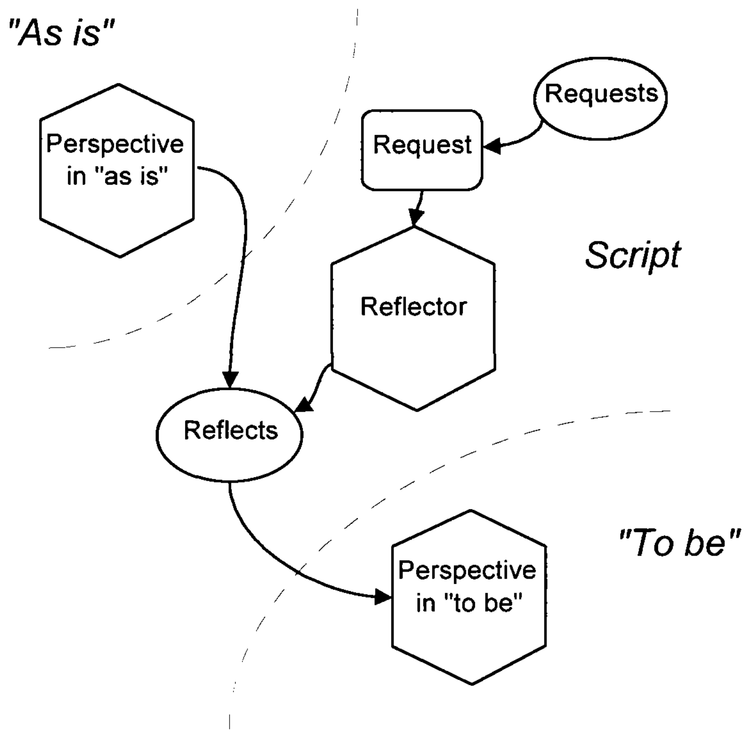


Figure 24b: A generic knowledge domain script.

The generic physical domain script transforms a physical state in the "as is" situation, or introduces a new physical state (not presented). The process actually branches off the "as is" model, as the normal flow of events cannot happen anymore (hence the "XOR" in figure 4a). However, the new situation "to be" may result in a new flux of events.
The generic knowledge domain script introduces an alternative perspective or introduces a new one (not presented). It is the possessor of this new perspective that decides whether or not he/she/they will use it. An example has already been provided by the "Throw away battery" example.

The "library" of generic *Trinity* models presented above is hardly a library, and severely suffers from what was identified to be the most important limiting factor for successful use of generic strategies: it does not cover the domain of multi-actor problem-solving processes[56].
In the Experiments part of this dissertation (part IV), several *Trinity* models will be presented, many of which will appear to be generic. This is a feasible and practical method to extend the library: by means of applying the *Trinity* approach in practice, and adding generic results to it.
It is possible to construct domain-specific sub-libraries (see, for example, the indoor environmental models in the Experiments part of this dissertation). On top of this, a (more abstract) domain-independent library can be constructed, containing generic models that require more specification, but (in line with the trade off) possess a larger genericity. Such a domain-independent library may be divided into complete models of perspectives (i.e. combinations of analysis models, scripts and prediction models), or may be divided into analysis models, scripts and prediction models (which offers the opportunity to construct other combinations as well). The construction of libraries of generic *Trinity* models will be a continuous task for frequent users of the approach.

5.5.3 Multi-representation strategies

A multi-representation modelling step changes the number of alternative viewpoints with respect to a referent. Multi-representation strategies utilise this phenomenon. As *Trinity* is directed at modelling perspectives, and each alternative perspective offers a scenario[57], a nickname for a multi-representation strategy is a *scenario strategy*. A *Trinity* model of a scenario, being constructed out of a set of perspectives, therefore consists of an "as is" model, a script and a "to be" model.

---

56 That is, not at an adequate level of detail: the model of an intentional activity, in principle, covers **all** intentional activities. This implies a maximal genericity. In line with the trade off, its heuristic value is limited, and more specific generic models are required to extend the library.

57 The term "scenario" is sometimes used to refer to several potential developments of one and the same initial situation: it is not known which scenario will develop. This type of use of the concept "scenario" implies a rather **autonomous** development of the flux of events. Within the *Trinity* approach, a problem solver **intends** to **act**, to **intervene** in the flux of events. *Trinity* scenarios model different plausible **action potentials**, rather than autonomous developments.

Scenario strategies in a sense are the opposite of generic model strategies: where one generic model models several referents, a scenario is one of several possible models of one and the same referent. Several alternative scenarios are generated (several plausible perspectives are being modelled), and after a potentially complex assessment procedure, one of them is selected as the one to be acted upon. Scenarios may differ completely (all three partial models, the "as is" model, the script and the "to be" model, are different) or partially (for example, the "as is" models are the same, but the scripts and the "to be" models are different). A scenario strategy furthers *exploration* of several options, and prevents premature closure: a fixation on one alternative, a phenomenon well-known in the literature of decision making and policy analysis (see, for example, [Geurts and Vennix (1989)], [Beach (1990)]) is avoided.

5.5.4 Compound strategies

In the preceding section, some important *Trinity* modelling strategies were explained. Typically, D-type problem solving involves a complex modelling process, with many roundabout ways and dead ends. In terms of modelling steps, the modelling process may be a complex combination of all twelve types of modelling steps. In line with this, several more simple strategies may be sequentially combined into a more complex strategy.
Sequencing strategies simply means that the output of applying strategy 1 becomes the input of applying strategy 2. We want to suggest a sequential combination of some of the aforementioned strategies that help illustrate this point: the following sequential strategy may be used to support the problem-solving process in very complex D-type situations:

Stage 1: Apply generic strategy

After a first orientation in the D-type situation of concern, start with browsing the library of generic models (a *generic model strategy*). At this moment, this will not take a very long time (the library is rather limited). However, the model of an intentional actor always applies (as you are a problem solver). Along with this, in case of an environmental problem the “minimal environmental situation of concern” is likely to provide a starting point for modelling the "as is" situation and (if needed) the acknowledgement process. If it is not possible to find more specific generic models, this will have to do.

Stage 2: Apply scenario strategy

After this, start a *scenario strategy*: try to find alternative instantiations of the actor model, i.e. alternative perspectives. This explorative step is important in order to prevent premature closure: a global overview is obtained of several possibilities to improve the situation of concern. Only when a first idea is present concerning several alternatives (or the absence of alternatives is validated), should more in-depth investigation of alternatives in isolation begin (see stage 3).

Stage 3: Apply expansion strategy to each alternative

This inaugurates the next stage: start with *expansion strategies*. The rather abstract perspectives resulting from the first two stages form a good starting-point for a *specification* strategy.
However, typically at the beginning of a complex problem-solving process, a top-down strategy is rather difficult to apply (the scope of the alternative perspectives is rather vague). Therefore, simultaneously an *extension* strategy (which is a building blocks strategy) can start for each of the alternative perspectives. In contrast with a specification strategy, an extension strategy enables one to "browse" through all the knowledge resources that are available. Building blocks (referring to actors, actions, processes, intentions, actor-specific "as is" and "to be" situations, autonomous initial and final situations, et cetera) can be obtained by means of literature searches, interviews, group sessions, workshops, et cetera. Virtually *any* means to acquire knowledge may be used. This may result in inspiration for the specification strategy as well.
In the end, the results of the top-down specification strategy and the bottom-up extension strategy should coincide: they both result in the same pragmatically correct model of a perspective. The parallel application of the specification strategy and the extension strategy forms a bi-directional approach.

Stage 4: Select perspective

In the likely case that at the end more than one perspective is available (i.e. several scenarios are available), each of them must be assessed with respect to issues like the goals (the value system) of the problem owner; the (financial) effort required to realise the script and to maintain the "to be" situation; the certainty (reliability) of the perspective, et cetera. As was explained in Chapter 3, the action potentials of different scenarios are likely to form an "exclusive or". Multi-criteria methods may support this step (which actually is a representation restriction step, a decision-making process).

Several cycles may be conducted. In terms of the theory of qualitative modelling processes, this compound strategy, basically, is a crude form of a *balancing strategy*. For example, the transition from stage 2 (think about all alternatives in combination at an abstract level) to stage 3 (expand each of the alternatives in separation) is called for in order to remain capable of coping with complexity.

5.6 USING *TRINITY*: PRACTICAL GUIDELINES

In section 5.3.2 we explained that *Trinity* is a backbone methodology, in which virtually any knowledge acquisition method may be inserted. In section 5.5, different modelling strategies (typical sequences of modelling steps) were described. Each of these strategies is directed at a different goal (in terms of scope and/or level of detail of the model). In section 5.7, different modes in using *Trinity* will be distinguished.

From these sections it is clear that *Trinity* is a flexible toolbox, rather than an imperative prescription for operating in D-type problem-solving processes. The obvious reason for this is that, although similar in that they all are intentional activities involving many parties, individual D-type problem-solving processes are quite different in many other respects.

Nonetheless, novice users of *Trinity* at first are typically overwhelmed by the number of different ways in which *Trinity* may be used. This, for an important part, is the result of the difficult research topic that D-type problem solving unavoidably is. In addition, D-type problem solving is a practice, rather than a science. The rules that we use in selecting (and designing) a specific way of using *Trinity* at this moment are not only diverse, implicit and of a heuristic nature, but they are also developing at this very moment. At this moment they are rather crude rules of thumb rather than formal rules. In the Experiments part of this dissertation we will describe some specific ways of using *Trinity* in specific situations. Parallel with using *Trinity* on a larger scale in real-world D-type problem contexts, however, the knowledge base that relates problem characteristics with specific ways of using *Trinity* will develop further. Formalising this knowledge base will be a research topic for the (near) future (first steps will be presented in Chapter 10).

As experienced users, we are rather partial of the flexible nature of *Trinity*. We more and more start to interpret *Trinity* as a pair of spectacles that can be used to order and interpret the world around us: *Trinity* is in our heads rather than on the paper in front of us. Novice users of *Trinity*, on the other hand, tend to find the great number of different ways in which *Trinity* may be used confusing, rather than supporting. An often asked question is: "How should I start?". As explained above, it is difficult to answer this question in a general manner. It is possible, however, to present an overall outline that can be adapted to suit specific situations. This outline is presented below[58].

5.6.1 A global outline of the *Trinity* process

Means to acquire information

In general, a D-type problem context is rather confusing. It is difficult to grasp an integral understanding of such a context. Three major routes can be followed to obtain information that is of help in obtaining such an understanding:

1. direct observation (of the physical domain);
2. communication with actors (in order to obtain an understanding of their knowledge domain); and
3. written sources (the communication domain).

[58] People who understand Dutch may want to read [Diepenmaat (1997)], which provides an extended version of this section of the dissertation.

According to the *Trinity* methodology, a D-type problem context should be understood in terms of perspectives. Perspectives can be modelled as systemic constructions of intentional and autonomous activities, that take place in the physical, knowledge or communication domain.
When acquiring knowledge (by means of observation, communication or reading), it is therefore well advised to structure information in terms of intentional and/or autonomous activities, as these are the "building blocks" of models of perspectives.
In order to support doing so, we introduce the notion of *activity lists*. Activity lists exist in two forms: *intentional activity lists* (short: intention lists) and *autonomous activity lists* (short: process lists).

The intention list

In principle, an intentional activity list (short: *intention list*) is a list of actors (players). For each actor a specific set of features is represented in a structured frame format. A minimal intention list comprises only two features per actor: its name and its role (function). In line with the model of intentional activities, a more complete set of features might be: the name, the role (function), the trigger (the phenomenon that causes acknowledgement), the "as is" situation, the action, the "to be" situation. An even more elaborated list might include features like an explicitly separated representation of the perspective of this actor and the environment (both in terms of "as is", script/action and "to be"), resources and other supplies, costs, time episode, et cetera. Of course, it is possible to elaborate upon each of these features by means of a textual attachment. A minimal requirement of the features list for each actor on the intention list is that it helps in understanding the action of this actor as an intentional activity.

The process list

In principle, an autonomous activity list (short: process list) is a list of all the autonomous activities that can be distinguished in the problem context. A minimal process list is simply a list of processes. A more elaborated process list might separately describe the "as is" situation, the process and the "to be" situation of the autonomous activity. Here also, each feature may be explained further by text in natural language. Table 2 presents some examples of items of an intention list and a process list, respectively.

In many cases at the beginning of a *Trinity* process, documents (reports) and a crude list of actors involved are available. These form an excellent point of departure for developing *Trinity* models.
The *documents* may be scanned and every autonomous process, actor or role (or any feature of the intention or process list for that matter) can be marked with a highlighter. The highlighted parts may be used to construct a first version of the intention and process list.
The *list of actors involved* may be used to start a range of initial interviews. The reports of these interviews may be processed in the same way as the documents, resulting in an extended intention and process list.

Typically, after such a first cycle of information acquisition and interpretation the material is still rather raw in two respects.

Table 2: Example elements of intention and process lists. The first example is a rather concrete element of an intention list; the second example is more abstract. The third element is from a process list.

Actor-name: Peter Carburettor	
Role: Mechanic of Deux-Chevaux	
Trigger: Order from chief of garage	*(a communication domain trigger)*
Action: Clean spark-plugs	
As is: Dirty spark-plugs	
To be: Clean spark-plugs	*(physical domain "as is" and "to be")*
Actor-name: European Union	
Role: Develop European legislation	
Trigger: Environmental effects	*(a physical domain trigger)*
Action: Develop concept-law for emission of Volatile Organic Compounds	
As is: Missing VOC-law	*(missing things are* not *modelled)*
To be: Concept law	*(communication domain)*
Process-name: natural emission	
As is: methane in swamp	
To be: methane in the air	

First, although the lists may be rather long (up to several hundreds of actors and processes), many of the features of the elements of these lists are likely to be empty. Working on intention lists and process lists makes clear that natural language is rather ambiguous, implicit and incomplete. For example, sometimes only an actor's name is mentioned, sometimes only a role is mentioned, and sometimes only an action is mentioned. Not seldom are autonomous processes left out altogether. Completing the intention and process list is a non-trivial enterprise that in many cases requires additional knowledge acquisition processes. Nonetheless, in order to fully understand an intentional activity it is important to have a pragmatically correct picture of this activity. Completing the intention and process list may seem much work, but it is an essential first step in order to be able to grasp a thorough understanding of the D-type problem context as a whole.

Second, on the basis of only the initial intention and process lists it is next to impossible to construct a coherent *Trinity* model. Two important reasons are that crucial actors are simply missing and that the levels of abstraction at which actors are described in many cases match only poorly, if at all. Examples are provided below.

Crucial actors are missing

An example of missing actors is that a new "green" product is produced, that a potential consumer group is identified, but that no one was identified to distribute and (re)sell the product. The actors "producer" and "consumer" cannot be interpreted as a pragmatically correct network without distinguishing the intermediate actor(s).

Mismatch between levels of abstraction

An example of a likely mismatch between levels of abstraction is the situation in which both the European Union and Peter Carburettor are present in one list.

From lists towards models

It is not by accident that missing actors and mismatches in level of abstraction remain rather invisible in natural language, and appear without mercy in attempts to develop *Trinity* models. The *Trinity* modelling language highly supports thinking in terms of networks of intentional actors. *Trinity* models explicit the relationships between actors, and facilitate thinking about networks as coherent (pragmatically correct) wholes, in which the "to be" of activity x functions as the "as is" or the trigger of activity y. Answering questions like "are players missing" and "do levels of abstraction correspond" is much easier when working on models that explicitly rely on the presence of all actors of relevance and matching levels of abstraction. Partial intentions are present in the intention list; *Trinity* models on top of this enable one to think through the consequences of combining these partial intentions in sequential and/or concurrent ways.

Before starting the construction of *Trinity* models, it is important to make two additional distinctions on the basis of the intention list.

The *first* distinction is between *field players* and *advisors*. Field players are part of (are referred to in) the "as is", the script and the "to be" parts of the very perspective of concern. Advisors, on the other hand, do not play a role in the very perspective itself. They rather contribute to the *development* of this perspective. Examples of advisors are researchers and informers.

In practice, it is well advised to thoroughly consult field players before entering the implementation stage. This implies that many of the field players play advisory roles as well. Indeed, field player and advisor are roles rather than persons.

The *second* distinction is a refinement of the group of field players into *"as is", script and "to be" players*. This distinction is relevant as they will appear in different parts of the perspective being modelled (the "as is", script and "to be" parts, respectively).

The central actor, the problem owner (who may be a group of persons or organisations or who may be represented by representatives), is not present in the intention list nor in the model. He/she is the one who intends to alter the D-situation of concern. He/she constructs the very intention and process list. He/she is the driving force behind the problem-solving process as a whole. Table 3 summarises the distinctions made in the intention list.

Table 3: Distinctions on the basis of the intention list.

Actors on intention list
- *consist of*
  - Advisors (do not model them, consult them)
  - *and*
  - Field players (model them as hexagons)
    - *consist of*
      - "As is" players (to "as is" part of model)
      - *and*
      - Script players (to script part of model)
      - *and*
      - "To be" players (to "to be" part of model)

The intentional activities present in the "field players" part of the intention list are used as a basis to construct an initial *Trinity* model. As stated above, this may not be trivial, for example, because of missing actors (as well as the presence of actors without proper reason) and discrepancies in level of abstraction. Going from the list notation towards the network notation (*Trinity* models) again and again proves to be both an essential and complicated step, which stresses the usefulness of using lists and models on top of (rather than as an alternative of) natural language. This issue will be elaborated upon in the Discussion section of this chapter.

The *Trinity* process is now running at full speed. Models are being adjusted and rearranged, using the library of modelling steps and the modelling strategies of Chapter 5; elements of the intention and process list are being adjusted, added, combined, split and deleted; additional knowledge acquisition processes are arranged and executed; and this possibly in many sequential and concurrent cycles. Note that either adapting the original *Trinity* model or altering the lists implies that the intention and process lists do not correspond with the *Trinity* model anymore. Such a mis-correspondence should be avoided, as the elements of these lists explain the elements of the *Trinity* model further.

Correspondence between *Trinity* model on the one hand, and intention and process lists on the other should be maintained[59]. The process stops if action is thought to be both possible and resulting in an improvement (at that moment the model of a perspective is bi-directionally coupled to intention *and* environment, see also Chapter 2). The implementation stage can be entered; the script may be executed. Figure 25 presents a summary of the *Trinity* process as described above.

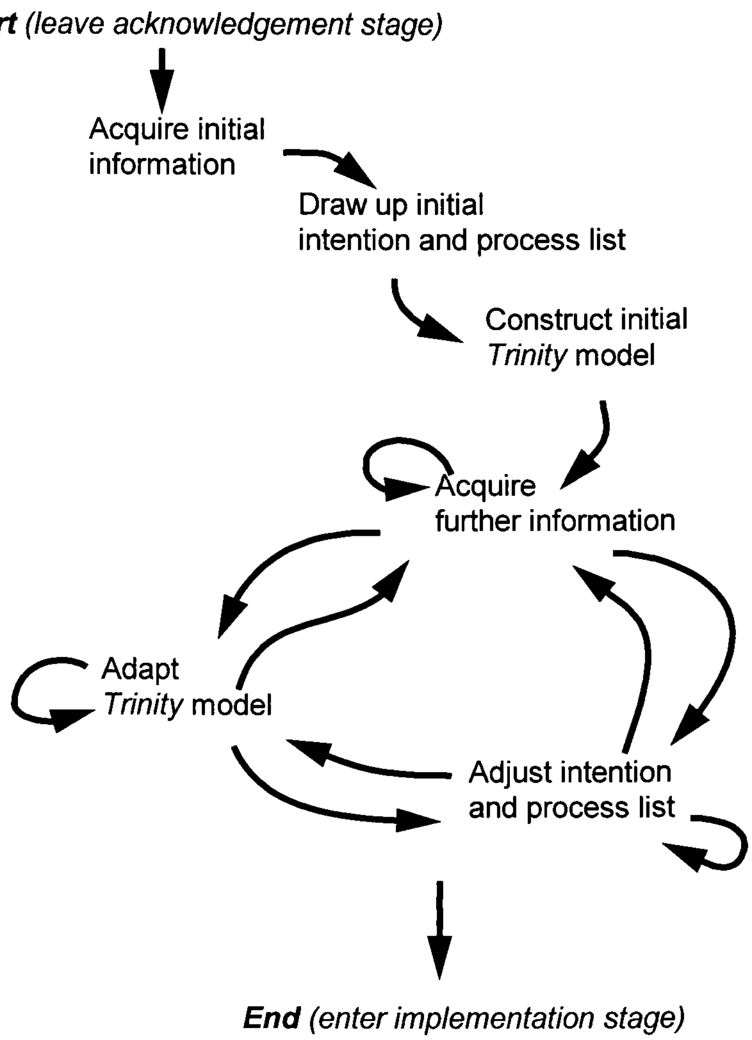


Figure 25: The *Trinity* process from a bird's eye view.

[59] Maintaining this correspondence should be an important design consideration in developing computer-based support for using *Trinity* (a "*Trinity* workbench"). It is for example possible to develop both structured list-editors and model-editors, that use each other and keep each other consistent.

5.6.2 Reflection on the practical guidelines

In this section, a guideline is described that offers support to users in applying *Trinity* in actual D-type problem situations. This guideline emphasises a specific[60] dimension: the global process of going from natural language (documents, interviews, et cetera) via intermediate structured lists (intention and process lists) to *Trinity* models.

This dimension is important, as natural language is by far the most intensively used medium to express concerns and communicate about solutions. This raises the question why departing from natural language as the medium of preference to communicate is worthwhile at all. This question has been answered implicitly in this section already. Here we want to explicitly elaborate upon the argument.

First, we want to stress that using *Trinity* is *not equivalent* to departing from natural language as the communication medium of preference. The design of *Trinity* as a modelling language, as well as the introduction of intention and process lists as an intermediate structured frame language, is rather based upon quite a different observation: natural language, although by far the most flexible, expressive and natural language (sic) to express one's thoughts and feelings[61], in its normal use possesses some flaws that are difficult to resolve. *Trinity* merely fills in these flaws of natural language. Persons engaged in D-type problem solving will use natural language in the same way as before, but their conversations are supported by lists structured in a frame format and a modelling language as well.

In order to further explain this, table 4 summarises the format as well as specific strong points of the three representation media that appeared in this section. Emphasis is on intentional activities (although the same arguments apply to autonomous activities).

Table 4 makes clear that intention lists and *Trinity* models extend the strengths of natural language rather than replace them. As was remarked when deriving intention and process lists from sources in natural language (documents, interview and workshop reports, et cetera), natural language is a rather permissive way of expressing and exchanging thoughts. When trying to draw up an intention list, in many cases essential features like names of actors, roles, specifics of actions, intentions et cetera are either present in a rather vague or implied way, or are even completely missing in these sources. This

[60] In Chapter 5 several other dimensions of using *Trinity* were presented. In section 5.5 *Trinity* modelling strategies were presented: sequences of modelling steps that can be used to adapt models (i.e. they take place in the model dimension). In section 5.7 different modes of using *Trinity* will be described, that are distinguished on the basis of quite different global issues like the order in which the three parts of the perspective are being elaborated (trouble-shooting, trick-exploiting and back-casting variants), the way in which communication about the D-type perspective takes place (isolated, hidden and participative use), and the number of perspectives that are taken into account (cognitive mapping and scenario approaches).

[61] Perhaps a good runner up is body language, but certainly not the *Trinity* modelling language!

notwithstanding the fact that partners in conversation were quite confident that they had both an interesting and meaningful conversation (meaning was shared). Structured lists, by virtue of their pre-formatted frame structure, are far less permissive in this respect. Being engaged in a frame-structured conversation, however, is not likely to be a pleasant enterprise. For this very reason, intention lists are drawn up on the basis of analysing sources in natural language, and completed afterwards[62].

Table 4: Languages, formats and strong points.

Medium	Format	Strong points
Documents and conversations	*natural language*	*flexibility* *low threshold* *expressiveness*
Structured lists (e.g. intention lists)	*frame structures (structured natural language)*	*sharp representation of specific features* ***per actor***
Trinity models	*Trinity modelling language (model primitives and arrows)*	*sharp representation of coherence* ***between actors***

The difference between natural language and structured lists may be large, the difference between structured lists and *Trinity* models is even more astonishing. Admittedly, looking back to the original sources of the structured lists (the documents, interview reports and the likes) is of some help in turning lists into models. However, this does not cover up the large discrepancies in terms of missing actors and abstraction gaps that we have encountered in practice many times.

Apparently, natural language is far too permissive to clearly represent D-type perspectives without any supporting means. Obviously, it is possible, with hindsight, to fully and completely describe a *Trinity* model of a perspective in natural language. However, that is not the point we want to make here. The point is that thoroughly understanding certain aspects of a D-type problem context is not very well supported by natural language.

[62] It should be noted though that using *Trinity* influences one's way of conducting interviews and setting up workshops. Ultimately and preferably, *Trinity* resides between one's ears…

Notably, these aspects are: a) obtaining a good understanding of the pragmatics of intentional actors in isolation, and b) obtaining an understanding of the coherence between these activities as part of an overall system. Aspect a) is supported by both the structured list notation and the *Trinity* interpretation and model representation of intentional activities. Aspect b) is supported by *Trinity* models of networks.

5.7 MODES IN USING *TRINITY*

Different modes in using *Trinity* can be distinguished. We will restrict ourselves to the distinction of some typical modes in which *Trinity* may be used.

5.7.1 Isolated use, hidden use and participative use

When emphasising the persons who are actually using *Trinity*, three different modes can be distinguished. They are:

isolated use;
hidden use;
participative use.

Isolated use means that the problem solver constructs *Trinity* models of multi-actor situations in isolation, on the basis of introspection and thinking. In addition, he may consult literature and other information media. The term "isolated" emphasises that no other actors are involved in the problem-solving process. Although this mode of operation seems to be efficient and attractive at first sight, it often results in perspectives that are not shared by the actors involved in the implementation stage and the desired "to be" situation: the implementation process stagnates or does not start at all. Not withstanding this, some famous examples exist of isolated dealing with multi-actor situations [Machiavelli (1513)].
In case of *hidden use*, this shortcoming is avoided: the problem solver functions as a knowledge broker, and communicates intensively with actors involved in order to thoroughly test and refine his/her interpretation of their points of view. These actors either participate in the D-type situation of concern, or act as consultants/informers in the process of perspective construction. Elicitation sessions can be either bilateral or group processes (e.g. workshops). The modelling process is a background process, performed by the knowledge broker who is an expert *Trinity* user: the term "hidden" emphasises that other actors are not actually involved in the *modelling* process.
Finally, *participative use* means that *Trinity* models are being constructed by several actors in group processes. Participative use facilitates building consensus between different "stakeholders" (see, for example, Geurts and Vennix). The *Trinity* models function as *catalysts* in these processes (see also the overview of functions that models can provide, presented in [Diepenmaat (1993a)]). A potential bottleneck in *participative use* is that all the participants must understand the *Trinity* methodology, and be able to contribute

to the modelling process. In case of professional D-type problem solvers (knowledge brokers, policy makers, managers of large companies) this should not be a problem. In the case of representatives of societal actors, however, a mix of *hidden use* and *participative use* may be considered.

5.7.2 Trouble-shooting, back-casting and trick-exploiting modes

When emphasising *the order* in which the three parts of a perspective are being developed, many different problem-solving processes may be distinguished. We will restrict ourselves to the following trivalent distinction:

trouble-shooting variants;
trick-exploiting variants;
back-casting variants.

In *trouble-shooting* mode, emphasis shifts from the "as is" part via the script part to the "to be" part of a perspective. The reason for this is that in trouble shooting a problem has manifested itself in a rather confronting way in the actual situation "as is". Therefore, a necessarily first step is to diagnose this "as is" situation further. This provides handles for remedial action (scripts), which finally result in a "to be" situation (in which preferably business resembles as much as possible the situation before the trouble).

In *trick-exploiting* mode, the world "as is" is roamed for opportunities to apply a trick: this trick can be understood as a highly generic script. In this situation therefore, initially only the script exists, as well as generic descriptions of "as is" and "to be" situations.

Finally, *back-casting* mode is a mode with a rather reverse nature. After a global diagnosis of what is wrong in the situation "as is", emphasis is quite drastically shifted from this "as is" to possible "to be's", without caring too much about realisability of these futures. Only after several "to be's" have been developed, attention is paid to scripts that might realise these futures. A *back-casting* approach typically resuls in more drastic (less evolutionary) perspectives, as focusing on "to be" in early stages results in a rather explorative way of perspective construction, in which the standard idea killing arguments (like "that cannot be achieved", "that is too expensive", "that is not realistic" et cetera) are suppressed, or at least delayed until later stages in the process. For this reason *back-casting* may be advised in strategic explorations.

Other sequences are also possible. It must be remembered, however, that these distinctions are variations within a theme, the theme being that *any* problem-solving process is a mixture of stages 2a-2c of our model of intentional activities, and each step influences the following. The "extreme" situations presented above should be interpreted as shifts in emphasis, rather than as rigid stage transitions.

5.7.3 Developing several perspectives

So far, we have not emphasised the issue of decision making in this dissertation. *Trinity* in essence supports *construction* of alternative intentional action potentials, rather than *deciding* among them (although clear models of perspectives are likely to enhance decision making as well, and every construction implies a decision). As such, *Trinity* falls outside the "problem solving is decision making" paradigm.
However, it is very well possible, and in complex situations even well advised, to develop *several Trinity* models of perspectives. This results in a more "open" problem-solving process, "open" in the sense that several alternatives are thoroughly thought through, rather than that all the bets are on only one horse. Developing one, two or several alternatives does not change the way of using *Trinity*. It merely implies that in the > 1 case a decision-making step should be expected (that may, for example, be supported by a multi-criteria method, or that may encompass complex integrating steps).
We will elaborate two different modes of using several perspectives. They are:

a cognitive mapping approach;
a scenario-like approach.

In a *cognitive mapping* approach, the perspective(s) of *several* participants in a complex problem context is modelled independently. In a second phase, these different perspectives may constitute the raw material to build consensus, or at least to develop more integral overviews (see, for example, [Weick and Bougon (1986)]).
In *scenario-like* approaches several alternative candidate perspectives are being developed. Developing alternative perspectives resembles a scenario approach, in that several alternatives are taken into account. An important difference between perspectives and scenario, however, is that scenario typically are plausible *autonomous* developments, resulting in plausible future situations, rather than alternative action potentials. However, the *Trinity* modelling language may very well be used to model autonomous multi-actor developments and futures as well. This merely requires that the notion of intervention by a problem owner, implied by the presence of a script, be left out. A scenario may be interpreted as a model of a situation "as is", covering a long episode of time.
We are not sure, however, whether using the concept of scenario (autonomous developments) in favour of the concept of perspectives (action potentials) is an improvement. After all, why bother exploring the future, if not ultimately for reasons of intervention or otherwise intentional action (which boils down once again to the notion of perspectives)? We want to recall here the rather Heideggerian philosophical stance that was mentioned in Chapter 2: something *is* what can be or was *done* to or with it. To put it even more strictly: *meaning cannot exist without action potential*[63]. This is the very reason why we built the *Trinity* methodology around the concept of perspectives: they embody action potentials in a very essential way.

[63] And, *in extremo*: meaning *is* action potential!!!.

The potential use of the *Trinity* modelling language in scenario development will not be elaborated here any further. The relation between D-type perspectives and D-type scenario, however, is interesting enough to put on the research agenda for the near future.

Trinity is a flexible framework rather than an imperative prescription of how to operate in D-type situations. As for now, we failed to present clear rules of thumb that are of help in selecting a proper way of using *Trinity* in more specific situations. For example, in which cases should a trouble-shooting approach be used, and in which cases a back-casting approach? Although some common sense answers can be presented, rigidly answering these questions will require more extensive experimenting with the *Trinity* methodology, and therefore will be a research topic for the near future. The results of these experiments will have to be used as a basis for a handbook. In the next part of this dissertation (part V, experiments) we will present some first results of experiments in using *Trinity*.

5.8 DISCUSSION AND CONCLUSIONS

In this chapter, the methods layer of the *Trinity* methodology was described in detail. At the start of this chapter (section 5.2), elements of the philosophical background (part II) and the theories (part III) of this dissertation were recalled. In section 5.3, these theoretical elements were integrated into the *Trinity* modelling approach. In section 5.4, the (design of the) *Trinity* language was described. Section 5.5 discussed *Trinity* modelling strategies. Section 5.6 presented some practical guidelines, and in section 5.7, different modes of using *Trinity* were presented.

The description of the *Trinity* modelling methods makes clear that the relation with the Philosophical background and Theory parts of this dissertation is thorough indeed. During the description of these methods the notions of perspectives; D-type problem contexts; intentional activities; problem solving as perspective construction and model-based support for problem solving again and again break through the surface. In addition, the *Trinity* language is designed completely in accordance with the theory of qualitative modelling presented in Chapter 4. The notions of complex (parallel, multi-referent and multi-representation) *Trinity* model relations; *Trinity* modelling steps and *Trinity* modelling strategies are a consequence of this theory: they are implied by Chapter 4.

With hindsight, the *Trinity* modelling language extends the theory of Chapter 4 with:

- the *Trinity* principle, that specifies the three domains which can be distinguished;
- the (relative) notions of states and processes that further refine the *Trinity* principle;
- the notion of causation that enables one to construct complex combinations of these qualities;
- models of intentional activities;
- the distinction between autonomous processes and intentional actions;

- the notion of reflection as a subtype of intentional activities, in which the environment is part of the knowledge domain; and
- the distinction between physical domain strategies (directed at changing the physical domain) and knowledge domain strategies (directed at changing points of view, perspectives of actors involved).

The description of the *Trinity* modelling methods is rather thorough. In addition, the philosophy and theories that they are built on encompass an even larger body of knowledge. One might wonder to which degree this might constitute an obstacle in realising a more widespread understanding and use of *Trinity*. As is the case with many methodologies, however, in order to be able to *use Trinity* as a novice, a complete understanding of all the theories and design principles it is founded on is *not* required. *Trinity* models can be understood and constructed on an intuitive basis as well[64]. The bottom line is that *Trinity*, in principle, consists of only *four* different types of model primitives (rectangles, rounded boxes, hexagons and ellipses) and only *five* different types of arrows. In combination with an understanding of the notion of an intentional activity, this is enough to read and construct simple *Trinity* models. In this respect, *Trinity* is not so different from staff notation in music: for many persons it is possible to compose, read and play simple compositions, but only some of us become real professionals. These professionals start to conceptualise higher level notions of models and modelling processes. They start to build what in *Trinity* is called a library of generic models. They start to develop composing strategies. When reading music, they do this in larger "chunks" than a novice does. *Trinity* encompasses a machinery that a professional in D-type problem solving might use in many different variations. Indeed, it is a methodological framework, rather than an imperative description (a cookbook). Constructing, reading and implementing simple models, however, is possible as well.

The libraries of modelling steps may seem to be rather abstract at first. However, *behind every modelling step a modelling heuristic is hidden.* For example, extending an ellipse backward with a hexagon at position 5 implies that the modeller became aware of the fact that someone is responsible for (causes) the process of concern. Extending a state forward with an ellipse implies that the modeller became aware of the fact that this state changes because of some transition. Extending a rounded box with a hexagon at position 1 implies that some message is acknowledged. Transforming a model of an intentional activity into a state implies that for some reason the processes within this state are of no relevance from the modeller's point of view. Using *Trinity* often results in a thorough intuitive understanding of the heuristics, and at a certain moment it will be the heuristics that guide the modelling process, rather than the modelling steps.

---

[64] As a matter of fact, the first version of *Trinity* was an intuitive version. However, in order to obtain a crisp and clear methodology, a theoretical basis was required. After developing this theoretical basis, *Trinity* was re-designed in compliance with this theoretical basis.

Salient features of the *Trinity* modelling language are the fact that both autonomous and intentional processes are distinguished, and the fact that both the perspective and the environment to which this perspective is applied are referred to in a model.
The rationale for distinguishing both autonomous and intentional processes is simply that we consider both of them to be important features of societal problems. For example, the minimal environmental situation of concern would greatly suffer from discarding the difference between intentional and autonomous processes: the combination of both of them is the very essence of an environmental problem!
The rationale for representing both perspectives and the environments to which they refer, is that *knowledge domain problem-solving strategies* are quite different from *physical domain problem-solving strategies*. Knowledge domain strategies try to change an environment by means of attempts to change the *perspectives* of actors. Typically, this requires communication domain activities: other actors are informed or influenced. Physical domain strategies do not require communication, but are directed at changing the environment in a direct way. In many cases, solving societal problems requires well-balanced combinations. A simple case in point is the "throw away battery" example.

Trinity models are claimed to model D-type situations. However, one might argue that this is not correct: a hexagon hardly makes explicit the internal complexity of a perspective. From this point of view, a *Trinity* model seems to be more suited to refer to B-type referents, rather than to D-type referents. However, three important arguments are neglected in this line of reasoning. The *first* argument is that perspectives can be attributed a textual attachment: the perspective (or any model primitive for that matter) can be further explained in natural language. The *second* argument is that a model can be developed at different levels of abstraction (by means of applying a transformation strategy), which facilitates elaboration of complex perspectives. The *last* argument is that indeed a *Trinity* model does not *visually* represent the complexity of the hexagons that are in it. However, there is a difference between what actually is represented in a model and what effect this representation has on the interpreter of the model. Interpreting the model induces and supports recollection processes. The reference model primitives are understood to be references to more complex referents, and the arrows support understanding these referents in combination. What is supported is more than what is visually represented. In a *Trinity* model, there simply is more than meets the eye.

The central hypothesis of this research is that model-based support is beneficial for D-type problem solving. *Trinity* is a methodology that is intended to support D-type problem solving. It, therefore, provides a means to explore this hypothesis. In Part V: Experiments, several experiments in using *Trinity* will be presented.

PART V: EXPERIMENTS

CHAPTER 6

INTRODUCTION TO THE EXPERIMENTS

6.1 INTRODUCTION

This part of the dissertation will describe three experiments using the *Trinity* methodology in D-type problem contexts. The goal of performing these experiments is to investigate the use and presumed added value of *Trinity* under quite different D-type circumstances. Below, first the problem contexts of the three experiments will be briefly introduced. After that, the differences between these settings will be highlighted. Finally, the different ways in which the *Trinity* methodology was tailored to meet the requirements imposed by these settings will be explained.

In the chapters to follow (Chapters 7, 8, and 9) each of the experiments will be described in depth. In Chapter 10, a general discussion and conclusions with respect to the experiments will be presented.

6.2 THE PROBLEM CONTEXTS

Three different experiments using *Trinity* will be described:

- Indoor environmental problems;
- A national agreement between the Dutch government and the corporate sector to realise a substantial reduction in Volatile Organic Compounds (VOC) emissions by the year 2000 (*VOC2000*);
- A strategic conference to stimulate the development of new and innovative ways to deal with building and demolition waste.

All three problem contexts are suitable for using the *Trinity* methodology:

- A *problem* exists. In all three cases an actor (the problem owner[65]) acknowledges that something should be done: stage 1 (acknowledge situation of concern) of the model of intentional activities is entered. However, a perspective is missing, which prevents one from actually doing something about it. Stage 3 (implement script) cannot be started yet.
- The problem context is *D-type*. Many different actors play a complex role in the problem context. A multi-actor situation "as is" should be turned into a better multi-actor situation "to be".

On the other hand, the three problem contexts are quite different, in more than one respect. In order to be able to explain this, first a short overview of the three problem contexts will be presented.

Indoor environmental problems

People remain indoors a substantial part of their time. In some cases, the indoor environment causes problems because of the presence of specific agents. Occupants may complain about undesirable effects because of the presence of these agents (e.g. they complain about headaches). Much research is directed at situations that cause health problems in specific working situations (research directed at occupational diseases). However, problem-causing agents may also be present in private houses. An example is the presence of house dust mites (a biological agent). Many of the complaints, resulting from the presence of agents in private houses, must be typed sub-clinical (e.g. itchy skin, irritated eyes, headache), as concentrations tend to be lower than in working situations.
The problem situation of concern is a problematic indoor environment. At first sight, this is hardly a D-type problem context. However, an indoor environment is intentionally created and changed ever after by a large number of actors. Examples are an architect, a building constructor, a decorator, but also the occupant that changes his living environment in agreement with his preferences and wishes. Therefore, we interpret an indoor environment as a complex, dynamic D-type problem context rather than a physical system.
The central question in this experiment is: Do *Trinity* models offer an indoor environmental problem solver support in diagnosing a problematic indoor environment and producing a remedy?

VOC2000

Agreements between government and industrial sectors are important instruments in Dutch environmental policies. *VOC2000* is an example of such an agreement: it consists of "contracts" between the Dutch Ministry of Housing, Spatial Planning and the Environment and specific (industrial) sectors (e.g. graphical industry, synthetic materials and rubber

[65] No problem owner, no problem!

industry) to reduce the total emissions of Volatile Organic Compounds by at least 50 per cent by the year 2000 (the reference year is 1981). VOC emissions are reduced in order to minimise damage to buildings and crops, as well as to minimise health problems (respiratory system problems) related with high concentrations of ozone in the living environment. The *VOC2000* programme is in the implementation stage of the policy cycle. The programme is rather successful: for many industrial sectors, progress corresponds with targets set, but some industrial sectors show a stagnation, and some sectors are falling short. Some people claim that the falling short is caused by the fact that all the "easy" measurements have been taken, and the more "complicated" ones remain to be implemented. A study was conducted that encompassed many in-depth interviews with field players (both policy circles and business life) in order to determine the reasons behind this presumed complexity of measurements.
The central question in this experiment is: Does the use of *Trinity* offer help in understanding these sources of complexity in combination?

Building and demolition waste

In building and construction, a lot of primary building materials are used. Consequently, a substantial amount of building, renovation and demolition waste is generated. Therefore, the Dutch Ministry of Economic Affairs and the Dutch Ministry of Housing, Spatial Planning and the Environment initiated a strategic conference. The goal of this conference was to further the development of technologies and approaches that tackle this problem. The idea was to gather the main players in the field to discuss this matter, and to identify problem areas and opportunities. In addition, concerted actions should be started that specifically address these problem areas and opportunities.
The central question in this experiment is: Does the use of *Trinity* support the process of obtaining a clear vision about these problem areas and opportunities, and the actions that deal with them?

6.3 DIFFERENCES BETWEEN THE PROBLEM CONTEXTS

The three problem contexts, typical examples of D-type situations of concern, are different in several other respects. For an important part, these differences were dictated by the problem situation at the moment that we entered it. In this section, we will emphasise these differences, and also explain the consequences for the specific way of using *Trinity* that we adopted in each experiment.

The time scale

The three problem contexts differ considerably with respect to the episode covered by the perspectives of concern.
Indoor environmental problems are usually acute problems. Typically they should be solved within months, preferably within weeks.

VOC2000 is an activity that is in the implementation stage of a policy cycle. The time horizon is approximately 1-5 years.
Finally, the strategic conference *Building and demolition waste* is directed at a time horizon of far beyond the year 2000.
All three of them require intentional acting in a D-type context. Therefore, the *Trinity* methodology, in principle, should be able to support all three of them. Implicitly, the three experiments therefore would show that *Trinity* can be used in quite different time scales.

The mode of communication

In Chapter 5, section 5.7.1, three different modes of communicating with actors were explained: *isolated use*, *hidden use,* and *participative use*. The experiments to be described differed in the following respect.

In the *Indoor environmental problems* experiment, we did not actually communicate much with participants in the problem context. We rather functioned as knowledge engineers who interacted intensively with several experts in indoor environmental problem solving (a $\mathbf{D}_{personal}$ process). The reason for this is that we operated on a rather high generic level, where it is less important to consult specific field players. When focusing on the problem context, therefore, this would be an example of *isolated use*. However, *within* the team of experts, a *participative use* of *Trinity* was dominant. The models were discussed, adapted and revised by several specialists, although the expert in using *Trinity* took the lead in actually changing the model (assuring syntactical and semantic correctness).

In the *VOC2000* experiment, intensive communication took place with actors that participated in the problem context. However, *Trinity* models were not actually shown to them: the mode was *hidden use*. The reason for this is that actors, who are engaged in the process for only a small part of their time, should not be burdened with the *Trinity* modelling conventions. The *Trinity* models were used, however, as a basis for reporting the research results to these participants (this is indeed the essence of *hidden use*): the models were "translated" into natural language. Within the research team, *participative use* was the dominant mode.

Finally, in the strategic conference *Building and demolition waste* we did not actually engage in the problem context at all. The results of the strategic conference were rather analysed by us *ex post*, wearing "*Trinity* glasses". In addition, communication took place with the policy maker who was responsible for the strategic conference. Therefore, although we used the documentation material of the conference intensively, the mode of using *Trinity* should be called *isolated use*[66].

---

[66] It was, however, one of our main recommendations for future strategic conferences to support the actual preparations, the conference itself as well as the actions to follow by means of *Trinity* in a mixed hidden and participative mode, because of the additional insight offered by *Trinity* models.

At the moment of writing this, we are conducting several more experiments. There is a strong tendency to use *participative use* within research teams, and *hidden use* with respect to occasional contributors to the problem-solving process. In this way, the benefits of using *Trinity* are maximised, while the number of persons that should master *Trinity* is minimised.

The order in which the perspective develops

A perspective consists of an "as is" part, a script and a "to be" part. As was explained in section 5.7.2, the order in which the three parts of a perspective are being developed may differ considerably. We distinguished a *trouble-shooting* approach, a *back-casting* approach and a *trick-exploiting* approach. The three experiments that will be described in the next three chapters are also different in this respect.

Indoor environmental problems are rather confronting problems: an occupant utters a complaint. Therefore, in this case a *trouble-shooting* approach was adopted. In a *trouble-shooting* approach, first the "as is" situation is analysed (diagnosed); after that, a script is synthesised, and finally the "to be" situation is predicted. In indoor environmental problem solving, diagnosis is the most difficult task: obtaining an "as is" model that explains the complaints is the effort-consuming step. Once available, obtaining a script that will take the complaints away is not too difficult in general (although a serious selection problem may come up, see Chapter 7). The "to be" model is almost trivial: its most important feature is that the complaint (or, more to the point, the effect causing the complaint) should be absent.

VOC2000 is a policy process that is in the implementation stage of the policy cycle. Therefore, we "plunged in" in the middle of the process. As a result, in the *VOC2000* experiment, emphasis is on the description of the perspective underlying a complex and stagnating transition process as a whole.

Finally, the strategic conference on *Building and demolition waste* is a typical example of a *back-casting* approach. The actual situation "as is" develops into a problematic situation, and requires preventive intervention (see also the notion of preventive intentional activities, Chapter 4). Therefore, a plausible future situation (in 2010) is predicted. Emphasis in the first phase of the experiment is therefore on the "to be" model. After that, actions are agreed upon that are intended to realise this future, or at least go into this direction: emphasis shifts from "to be" to the script (by means of a backward reasoning process).

The Trinity modelling strategies that were used

The last discriminating feature that we will mention is the predominant *Trinity* modelling strategies that were used in the experiments (see also Chapter 4 and section 5.5).

Experiments

In the *Indoor environmental problems* experiment, the most dominant modelling strategies are referent restriction strategies and parallel specification strategies. A referent restriction strategy is used to refine generic models. A parallel specification strategy is used to obtain a more detailed model of a referent.
In the *VOC2000* and the *Building and demolition waste* experiments, the main strategy was a parallel building blocks strategy.
These differences will be discussed further in Chapter 10.
In the three chapters to follow, the three experiments will be described in more detail. In the discussion and conclusions chapter of this Experimental part of the dissertation (Chapter 10), the results of the three experiments will be discussed in combination, and general conclusions will be drawn. In addition, the added value of using *Trinity* will be addressed in generic terms at that point (whereas specific added value will be addressed in Chapters 7-9, describing the experiments in separation).

CHAPTER 7

INDOOR ENVIRONMENTAL PROBLEMS

Henk B. Diepenmaat and Maarten J. Leupen

7.1 INTRODUCTION

People remain indoors for a substantial part of their time. In some cases, these indoor environments cause problems, due to the (unknown) presence of specific agents. Occupants may complain about undesirable effects of the presence of these agents (e.g. they complain about headaches). Much research is directed at situations that cause health problems in specific working situations (research directed at occupational diseases). However, also in private houses, problem-causing agents may be present. An example is the presence of house dust mite allergen (a biological agent) or formaldehyde (a chemical agent). Many of the complaints, resulting from the presence of agents in private houses, are sub-clinical (e.g. itching skin, irritated eyes, headache), as concentrations tend to be lower than in working situations.

The problem situation of concern in this experiment is a problematic indoor environment: the occupant has complaints. An indoor environment can be interpreted as a static physical environment, consisting of building materials, decoration materials, and finishing materials. This interpretation is hardly in agreement with a D-type problem context. We do not consider it a very suitable interpretation though, for the following two reasons. The first reason is that it neglects the issue that an indoor environment constantly changes due to autonomous processes. Autonomous processes are, for example, the ageing of constructions and materials, or heating by the sun. The second reason is that an indoor environment is intentionally created and constantly adapted. Important actors are an architect, a building constructor, a decorator, but also the occupant that changes his living environment in agreement with his preferences and wishes. Therefore, rather than as a static physical system we interpret an indoor environment as a complex, dynamic system consisting of both autonomous and intentional activities. The indoor environment is a D-type context, rather than a physical state. Furniture is changed, windows are opened and closed, the sun heats the indoor environment, chipboard dissociates due to a large humidity and heat, and an order is sent to a furniture shop: these all are elements that may be of importance in indoor environmental problem solving. It is a D-type problem context indeed. This is in full compliance with the *Trinity* principle (see Chapter 5).

The central question in this experiment is:

> *"Is it possible to support indoor environmental problem solving by means of Trinity models?"*

The problem owner we are supporting in this experiment is a person, responsible for diagnosing a problematic indoor environment (and coming up with a remedy).
Figure 1 represents the indoor environmental problem solver in his/her problem context.

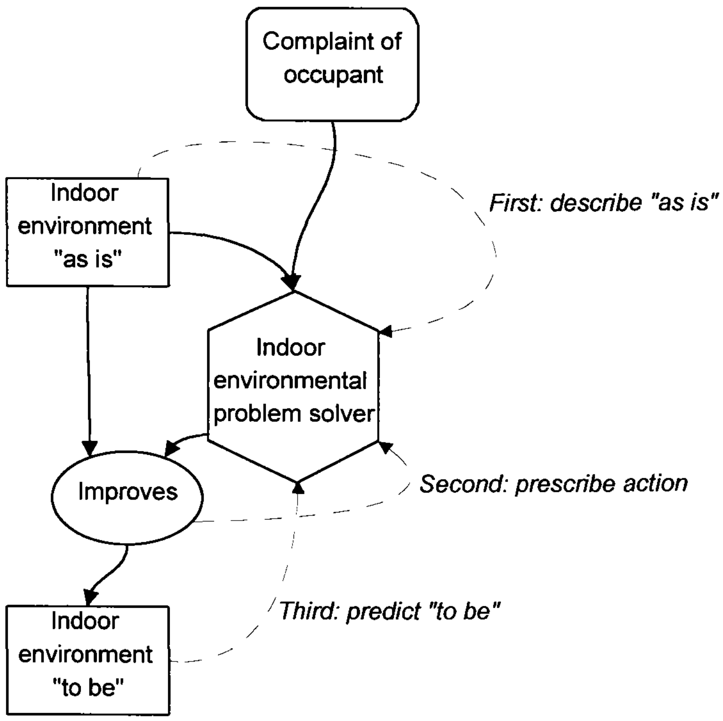


Figure 1: The indoor environmental problem solver in his problem context, following a trouble shooting approach (the sequence in problem solving (perspective construction) is: analyse "as is", prescribe script, predict "to be").

Figure 1 shows that the problem solver follows a *trouble-shooting* approach (Chapter 5): first the situation "as is" is described (diagnosed), second a script is created (a remedy is found), and, finally, the result is predicted. Implementation can begin.

It would be of great help to such a problem solver, if he would have at his/her disposal generic models of indoor environments that might function as a starting-point in further diagnosis and finding remedies. However, indoor environments are a rather large and heterogeneous group. In line with the trade off between genericity (re-usability) and heuristic value of generic models, discussed in Chapter 5, such generic models must be expected to be rather abstract.
In order to answer the above question, we will present (the construction process of) a generic "as is" model of a problematic indoor environment in section 7.2. Subsequently, in section 7.3, this model will be used to derive more specific generic models that support diagnosis of complaints resulting from specific sources and agents. This implies a hierarchy of generic models. In section 7.4, generic scripts will be presented to remedy the complaints. Several ways of using the models in indoor environmental problem solving will be described in section 7.5.

7.2 A GENERIC "AS IS" MODEL OF PROBLEMATIC INDOOR ENVIRONMENTS

As a starting point for a generic "as is" model, we will use the "minimal environmental situation of concern" model from the (at this moment still rather premature) library of generic models presented in Chapter 5. This model is repeated below for convenience (figure 2a). Figure 2b presents its indoor environmental specialisation. Both the model of figure 2a and its more specialised version of figure 2b are generic. In line with the trade-off between re-usability and heuristic value, figure 2b is less generic than figure 2a; its heuristic value is slightly larger, though.

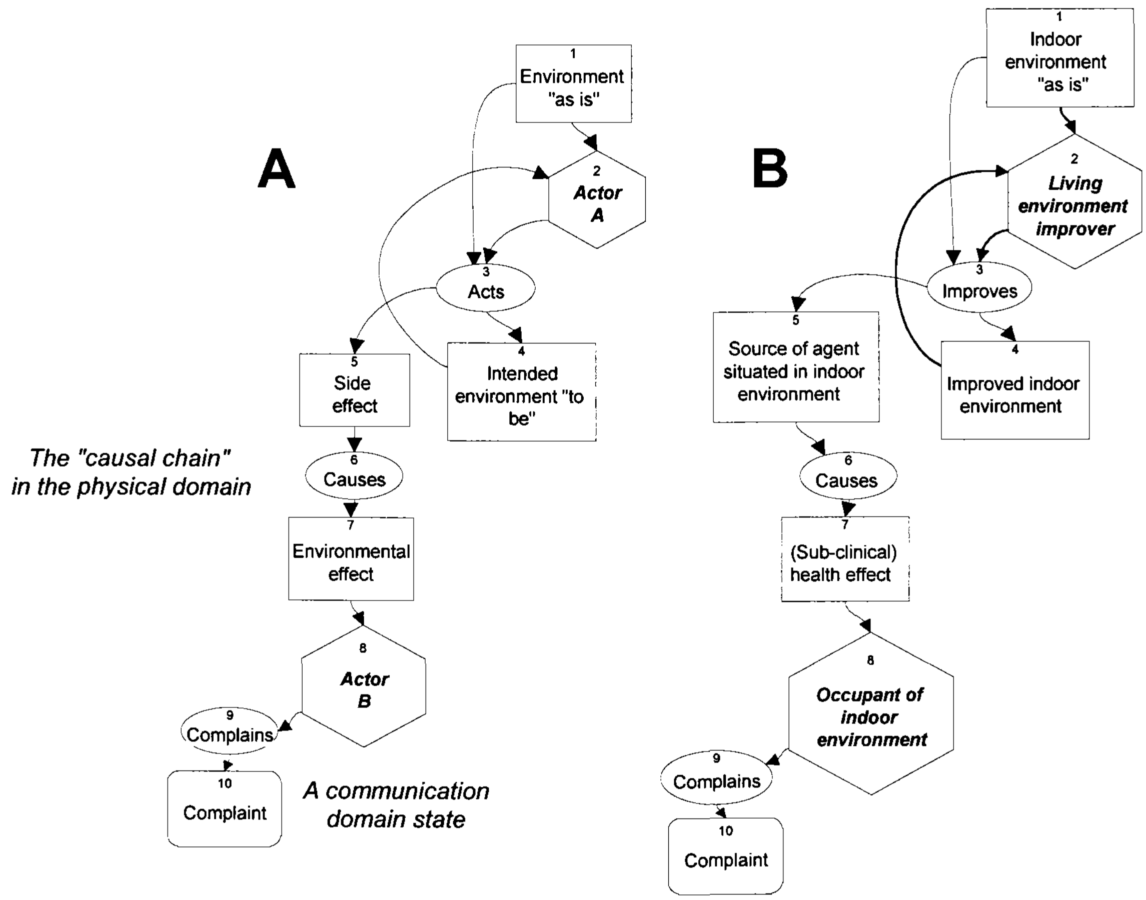


Figure 2a/2b: Generic "as is" models. Figure 2a presents the "minimal environmental situation of concern". Figure 2b presents the "minimal **indoor** environmental situation of concern".

Figure 2b is still rather abstract. However, by means of applying a transformation strategy (to be more specific: a parallel specification strategy) to parts of it the model can be made more specific. A series of parallel specification steps (not represented here) transforms the model of figure 2b into the model presented in figure 3: a more detailed generic "as is" model for indoor environmental problem solving. (For an earlier version of this model, see [Diepenmaat and Leupen (1991)], [Kok (1992)].)

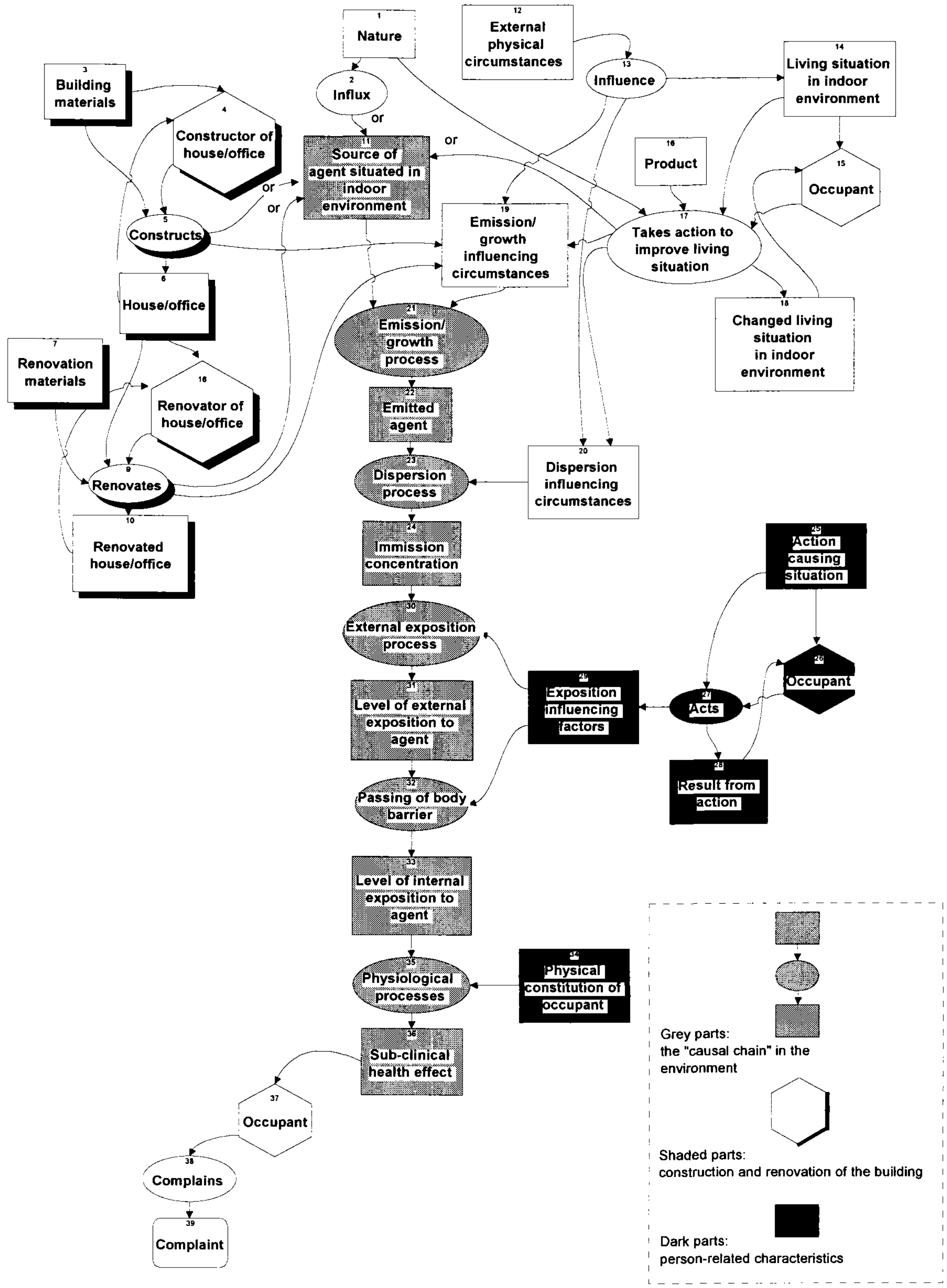


Figure 3: A more detailed generic "as is" model of indoor environmental situations of concern.

The model expresses that, in principle, four routes exist by which a source of an agent may be introduced into an indoor environment (see the four arrows entering rectangle 11): during the building and/or renovation process as a side-effect of an intentional activity (upper left), by means of an autonomous process from nature (e.g. pollen are blowing in), and as a side-effect of an activity of the occupant to improve his/her living situation (e.g. a cat is introduced into the indoor environment).
As a result of autonomous processes (ellipse 13, e.g. sun heating, rain, wind, leaking of a roof) or intentional actions (ellipse 17[67], e.g. closing the windows, introducing heat isolation, turning up the temperature, do-it-yourself activities), potential sources of agents may start to emit or (in case of a biological agent like house dust mites) may start to grow and emit (ellipse 21). The dispersion of this agent (ellipse 23) may be influenced as well, for example, by the ventilation behaviour or the moving around of the occupant. The respondent is being exposed to the agent externally, and (after passing the body barrier) internally (ellipses 30 and 32). The level of exposition depends on the specific person, rather than on the indoor environment: how long is he/she present, what kind of activity does he/she perform? The last stage is that the agent causes a sub-clinical health effect (for example, an itching skin, a head-ache, burning eyes, or a combination) or a clinical health effect (for example, allergic asthma, or cancer resulting from asbestos), which is influenced by her/his physical constitution (e.g. age, sensitivity, atopy, see ellipse 35).

When being confronted with an indoor environmental problem, in line with the *trouble-shooting* approach, the first task is to identify the causal sequence that resulted in the complaints. The "as is" model of figure 3 provides a generic scheme that may support this diagnostic task. The result of diagnosis will be a more specific version of figure 3 that tells the "causal history" of the complaint. (The difference between figure 3 and the result of diagnosis resembles the difference between models 2a and 2b.)

7.3 SPECIALISING THE GENERIC "AS IS" MODEL TO SPECIFIC SOURCES AND AGENTS

The generic "as is" model presented in figure 3 is still abstract. Therefore, in order to support diagnosis, more specialised "as is" models might be of help. Below, three examples will be presented and partly explained:

1. a model concerning house dust mites that may cause allergic asthma;
2. a model concerning chipboard emitting formaldehyde, which may cause irritated eyes, headache and itching skin; and
3. a model concerning photocopiers emitting ozone, which may cause a headache.

[67] Hexagon 15 represents the perspective of the occupant engaged in several roles, all of them being related with improving his living situation ("seeking pleasure" or "avoiding pain").

House dust mites and allergic asthma

Figure 4 presents a specialisation of figure 3 for house dust mites and allergic asthma. House dust mites typically enter the indoor environment by being brought in by human or animal transportation. As mites are a biological source, rather than a chemical or physical one, the "emission" process is preceded by a growth process in this case (the ellipse "growth/emission" might be specified into an ellipse "growth", followed by a rectangle "increased amount of mites", followed by an ellipse "emission of allergen": a dynamic parallel specification in terms of Chapters 4 and 5). Growth-influencing circumstances are humidity, temperature, the presence of a "habitat" (e.g. mattresses, upholstered furniture, carpets); the presence of food (crumbles, et cetera). For example, an occupant introducing carpets favours growing conditions, as the habitat improves; an occupant who vacuums often disfavours growing conditions, as food is taken away. Children playing on the floor are exposed additionally: playing on the floor is an exposure-influencing circumstance.

Chipboard causing irritated eyes, itching skin and headache

Figure 5 presents a specialisation of figure 3 for chipboard. If the chips of chipboard are glued with ureum-formaldehyde, this may dissociate resulting in formaldehyde. Formaldehyde is known to cause irritated eyes, itching skin and headache.

The dissociation process is a chemical process. Important dissociation-influencing circumstances are a high temperature and the presence of H_2O (the dissociation is a hydrolysis). An important emission-influencing circumstance is the precise composition of the chipboard (is the glue used ureum-formaldehyde, melamine-formaldehyde or another one), as well as the amount (surface) that is present.

Photocopiers and printers causing headache

Figure 6 presents a specialisation of figure 3 for photocopiers emitting ozone, which may cause a headache. Here an important emission-influencing circumstance is the intensity of use, which results from the intentional behaviour of the occupant. Factors that influence ozone dispersion and exposure are similar to the factors of formaldehyde, as both of them are gases.

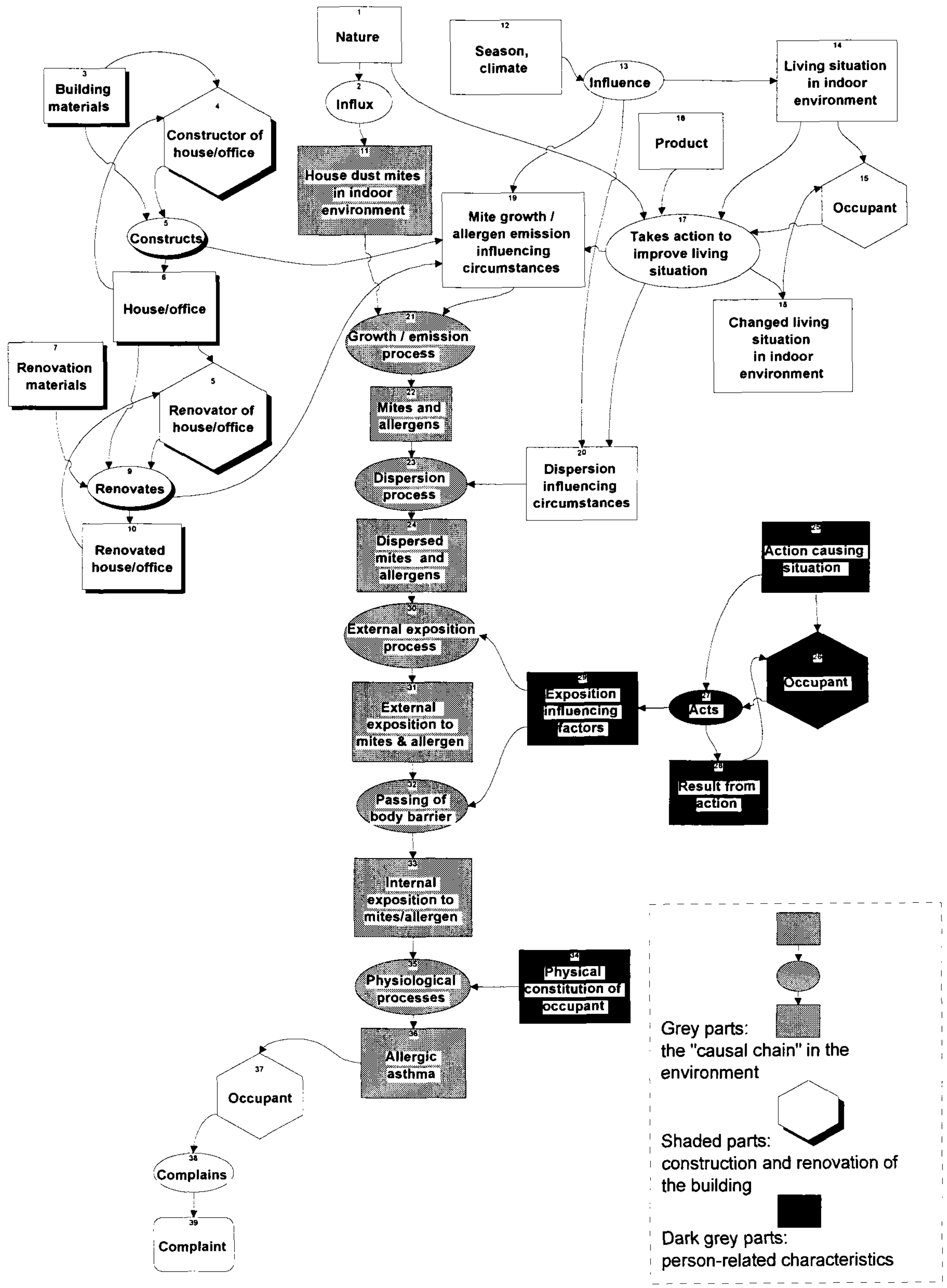


Figure 4: A generic "as is" model for house dust mites causing allergic asthma.

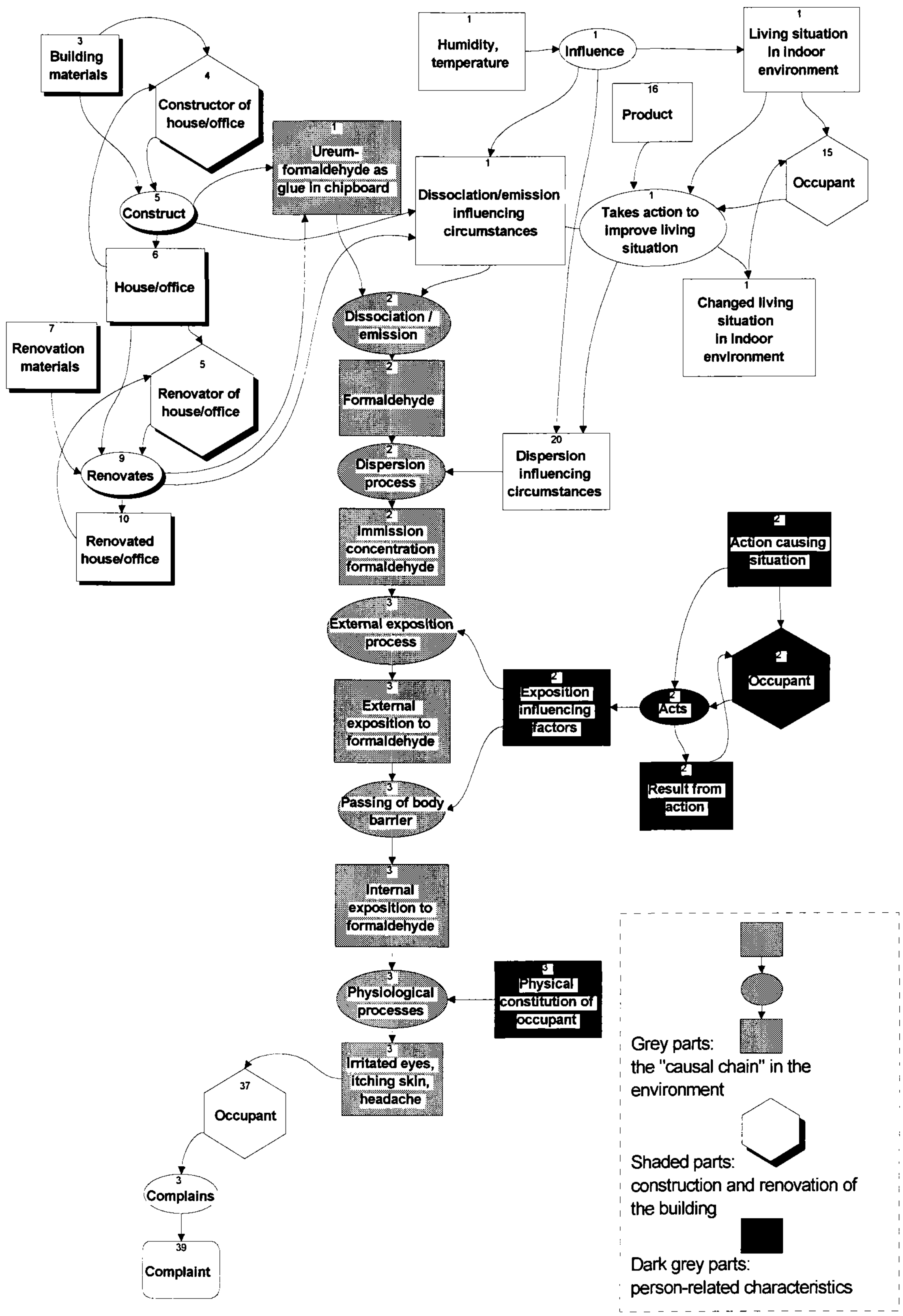


Figure 5: A generic model for chipboard causing several complaints.

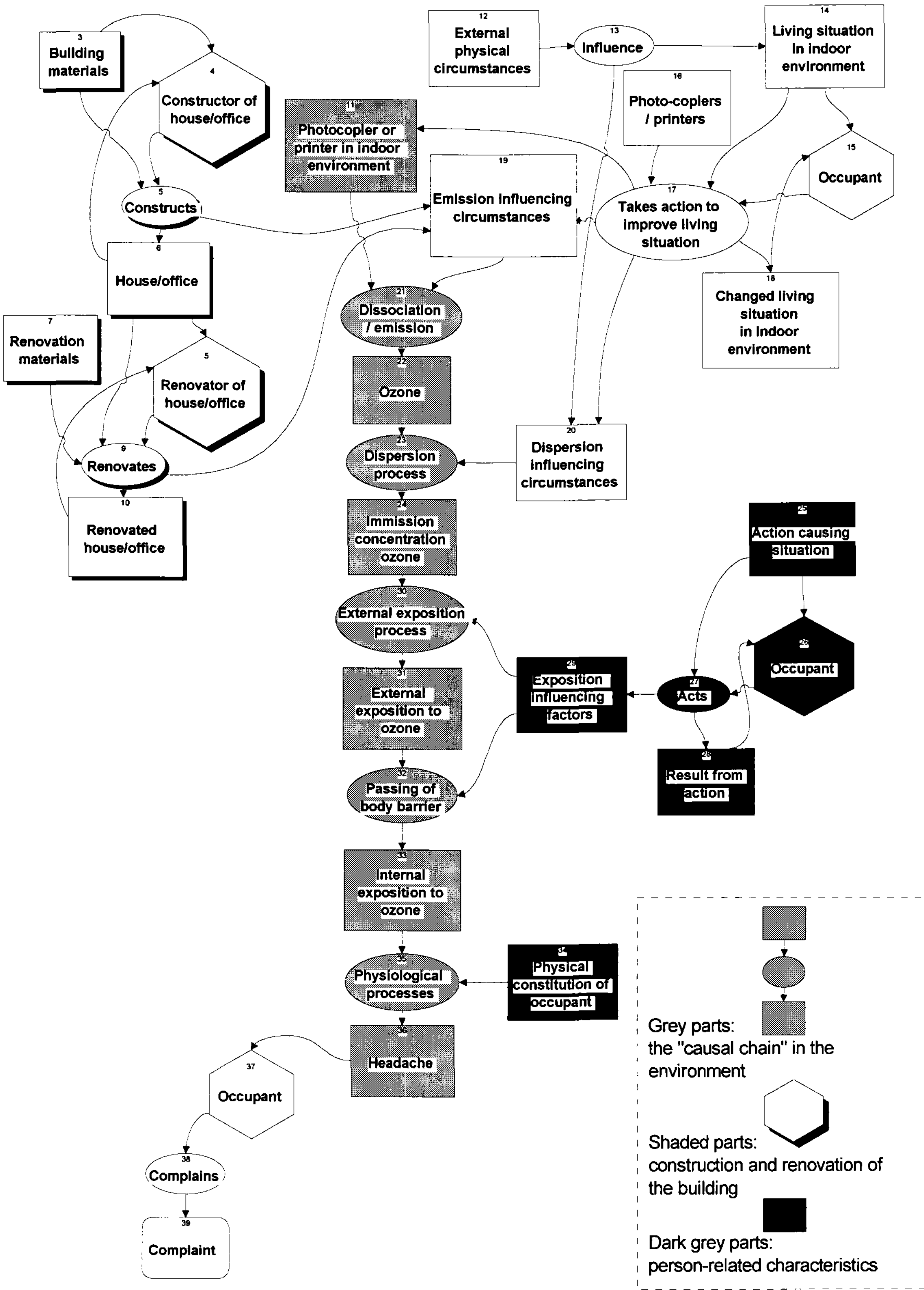


Figure 6: A generic model for photocopiers causing headache.

The three more specialised "as is" generic models presented in figures 4-6 are rather similar to the generic model presented in figure 3. Small structural differences can be noticed though.
For example, house dust mites enter a house predominantly by means of an unintended side-effect of animal or human transportation (in and on clothes and fur). Chipboard predominantly is introduced during construction and/or renovation activities. Finally, a photocopier or a printer is introduced by the occupant to improve his/her indoor environment (hexagon 15). This is shown by the arrows entering rectangle 11.
A biological growth process (mites), a chemical dissociation process (chipboard) and intensity of use (photocopiers and printers) are three quite different manifestations of one and the same ellipse (no. 21 in figure 3). Each of them is influenced by different factors (rectangle 19 in figure 3). This example shows that the generic model of figure 3 is generic indeed: it covers quite different problematic indoor environments.

7.4 GENERIC SCRIPTS TO TAKE AWAY (SUB-CLINICAL) HEALTH EFFECTS

An interesting feature of figures 3-6 is that the models are *not* pragmatically correct. Many of the rectangles modelled in the causal environmental chain are not associated with an intentional activity. However, figures 3-6 only represent the "as is" part of a perspective. According to the *Trinity* methodology, the second step in a trouble shooting approach is that a *script* should be modelled that enables one to "branch off" the "as is" model[68] somewhere upstream of the undesired health effect[69]. Implementing this script should result in a different "to be" situation in which this health effect does not manifest itself anymore. In this section, we will present some generic scripts that provide an overview of these possible interventions.

Generic scripts for changing the physical and the knowledge domain in a problematic indoor environment are presented in figures 7a and 7b. Like "as is" (and "to be") models, these generic scripts can be changed by means of the full repertoire of *Trinity* modelling steps and modelling strategies as presented in Chapter 5: they can be extended and restricted, abstracted and specified.

---

[68] As was explained in Chapter 2, branching off an "as is" model takes place by means of introducing an alternative causal route by means of an arrow that is part of the script. The "normal" route and the alternative route form an "exclusive or": doing nothing implies the normal route, intervention by means of implementing the script implies the alternative route.
[69] Also persuading the complainer to stop complaining, or neglecting the complaints altogether are well-known strategies to deal with problems. However, we will neglect these possibilities here.

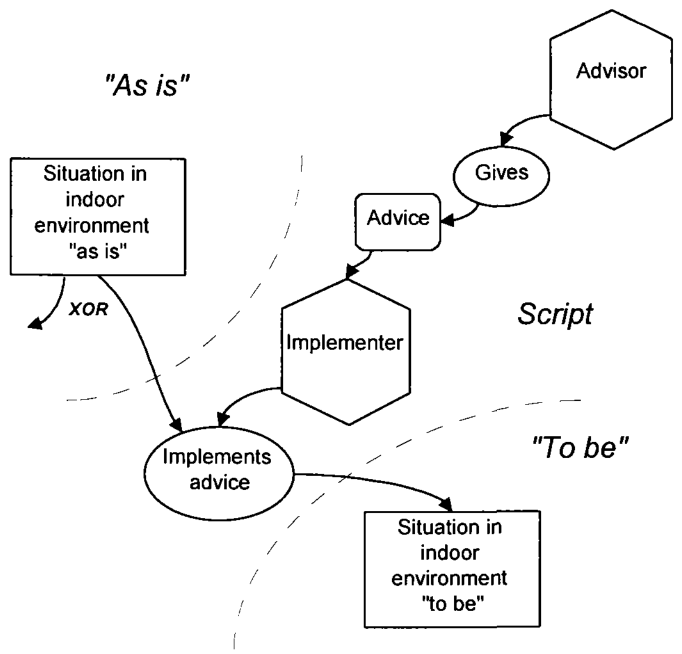


Figure 7a: A generic script that alters the physical domain.

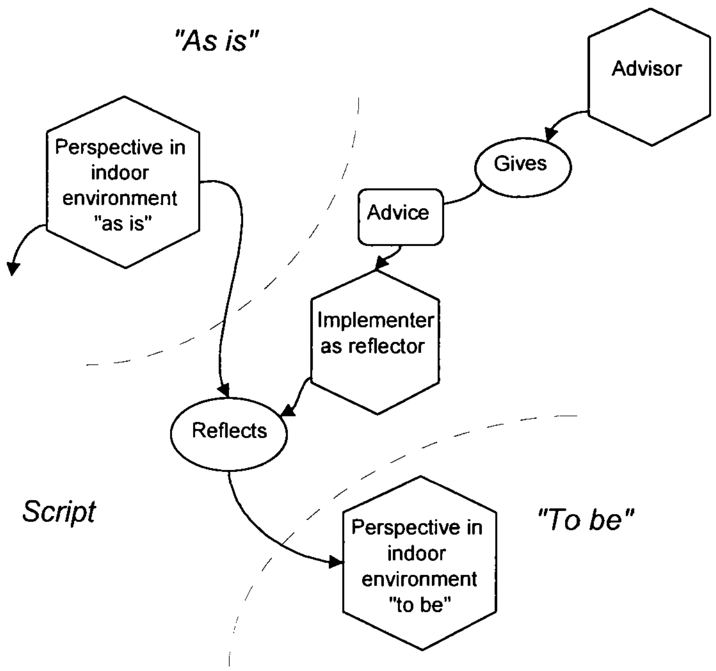


Figure 7b: A generic script that alters the knowledge domain of an indoor environment.

In the same way, parts of the "as is" model presented in figure 3 (or the specialisations presented in figures 4-6) can be branched off. In principle, **all** the rectangles "upstream" of the effect (rectangle 36 in figures 3-6) can be the point of application of a script that changes the physical domain (i.e. like figure 7a); and **all** the hexagons "upstream" of this rectangle can be the point of application of a script that changes the knowledge domain (i.e. like figure 7b). These positions in the model are points where the flux of events may be changed, and thus the undesired sub-clinical health effects may be avoided. Figure 3 (and 4-6) makes one thing clear: many potential branching points are present! Some important scripts will be explained.

Generic scripts preventing the source of the agent to enter the indoor environment

Sources of agents can be introduced in the indoor environment by any of the following four ways (see also figure 3 rectangle 11):

1. by means of the building construction process;
2. by means of a renovation process;
3. by means of an autonomous influx from nature; or
4. as a side-effect of an intentional action of the occupant.

Examples of type 1 and 2 are the use of formaldehyde-emitting chipboard. An example of type 3 is the influx of pollen from outside. An example of type 4 is someone who buys a photocopier that emits ozone.
Types 1, 2 and 4 can either be remedied by a knowledge domain approach or a physical domain approach. The knowledge domain approach would be to change the perspective of the actor in such a way that he/she/they would stop introducing the trouble-causing material. The physical domain approach[70] would be to prevent trouble-causing materials to be available, typically by means of some regulations. A case in point is the introduction of "KOMO" certified chipboard in the Netherlands: certified chipboard is low-emission with respect to formaldehyde.
Type 3 necessarily requires a physical domain intervention approach, as the influx process is autonomous. An example is preventing pollen from entering an indoor environment by means of applying a specific filter to the air-conditioning.

Generic scripts changing growth- and emission-influencing circumstances

Growth- and emission-influencing circumstances correspond with rectangle 19 in figure 3. Circumstances that influence growth and/or emission are quite diverse, and largely depend on the type of source. For example, in the case of a photocopier it is the intensity of use (an intentional activity) and the technical copying process (an autonomous activity) that

[70] "Physical domain approach" means that the goal is to alter the physical domain as represented in the "as is" model. This may require activities though in all three domains, as typically is the case in regulations.

determine emission; in the case of chipboard prepared with ureum-formaldehyde the temperature and the humidity affect the dissociation process that results in formaldehyde; in the case of house dust mites, activities like often vacuuming the house decrease the presence of house mites; and introducing carpets favours the presence of house mites. Therefore, quite different scripts, some directed at the knowledge domain (e.g. tell the occupant to often vacuum) and some directed at the physical domain (e.g. change the copying process) can be applied.

Generic scripts changing the dispersion process

The dispersion process, typically, is either gaseous diffusion (for agents that are gases or small particles) or mechanical diffusion (for example, walking through dust or some other agent on the floor). Again some of the scripts are knowledge domain approaches (tell the occupant to stop spreading the agent by walking around; or to open the windows more often) and some of them are physical domain approaches (alter the mechanical ventilation system; introduce ventilation devices in the gables of the building).

Generic scripts changing the external exposure process

External exposition is influenced, for example, by the length of time that one is exposed, or by the distance that one is away from the agent. Therefore, knowledge domain approaches would be to tell the occupant to remain shorter in the rooms or buildings of concern; or to keep as far away from the agent as possible.

Generic scripts changing the process of passing the body barrier

Passing the body barrier typically takes place either by the respiratory system (inhalation) or by the skin. An example of an internal exposition influencing factor is physical exercise: this increases the volume of air inhaled per unit of time, and therefore the uptake of gases like formaldehyde or small particles like pollen. A knowledge domain script therefore might be, for example, to tell the occupant to minimise physical exercise. A physical domain script would be to introduce a respiration filter (for example, in the case of do-it-yourself activities).

Generic scripts changing the physiological processes causing the health effect

A common denominator for the scripts that apply here is: give medication. Medication either prevents the effect to be perceived or prevents the effect to happen. An example is hypo-sensibilisation of an allergic asthma patient.

Combination scripts

Sometimes, a singular approach (utilising only one point of application in the "as is" model) is not enough. For example, often vacuuming a house is likely to reduce the presence of house dust mites, but is not enough to let them disappear altogether. In these cases a multi-threaded approach might work: in addition to keeping the house cleaner, a programme to reduce humid walls and floors might be implemented (house dust mites prefer a humid habitat), and smooth, seamless flooring and furniture might be installed in order to minimise the habitat of mites. A combination script might succeed where a singular approach would fail.

An example script in *Trinity* convention, directed at preventing health effects caused by photocopiers, is presented in figure 8. In specific situations these scripts are likely to be more specific: the states, processes and perspectives are further described, and likely several more intentional activities are distinguished in the script.

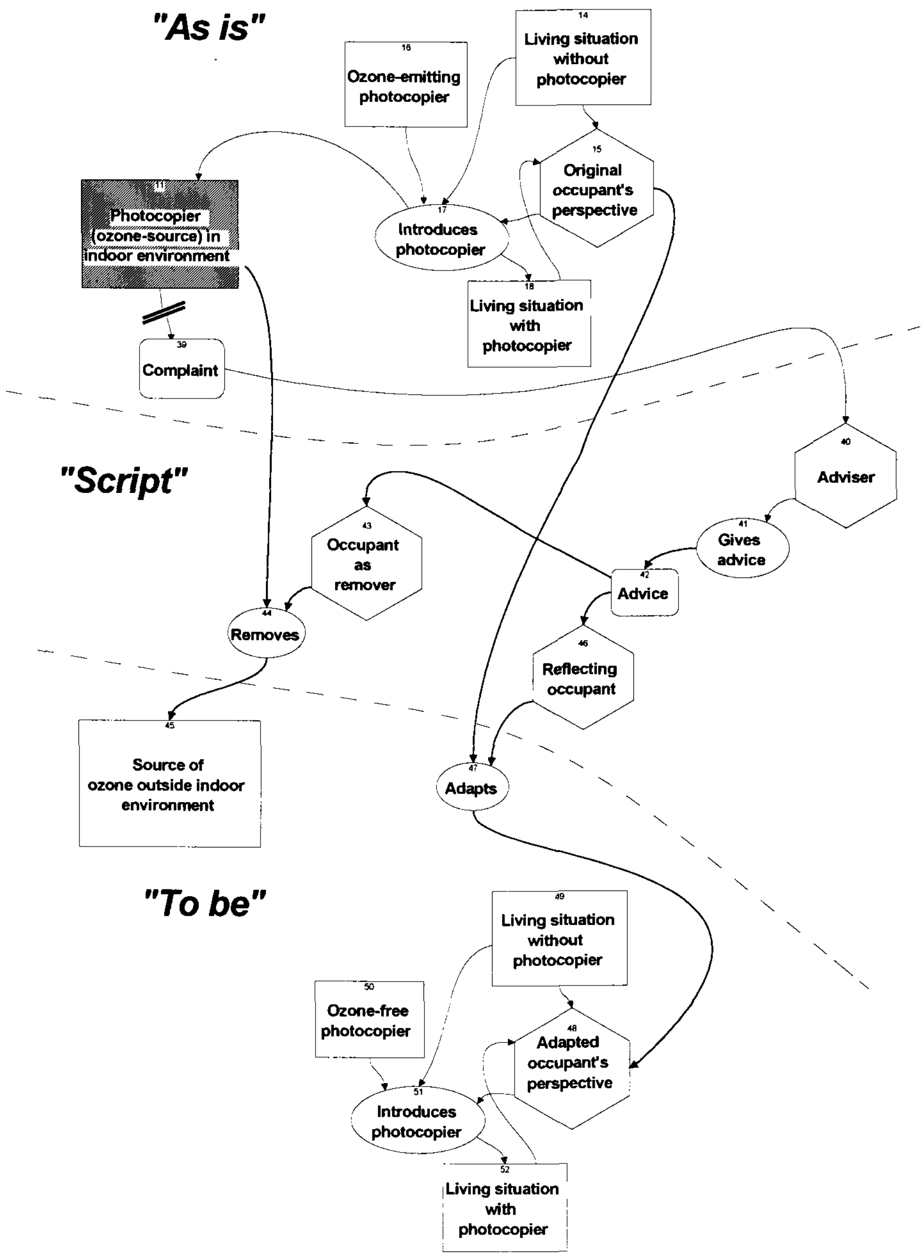


Figure 8: Remedying the ozone-related complaints caused by a photocopier.

Only the relevant part of the complete "as is" model (figure 3) and "to be" model is shown. The script combines a physical domain strategy (left, resulting in another physical state in the "to be" situation), and a knowledge domain strategy (right, resulting in a new (different) perspective in the "to be" situation).

7.5 USING THE MODELS IN INDOOR ENVIRONMENTAL PROBLEM SOLVING: A SKETCH OF A MODEL-SUPPORTED INFRASTRUCTURE

In this section, a knowledge infrastructure will be sketched that is expected[71] to result in a more effective and efficient way of solving indoor environmental problems. The infrastructure intensively utilises models like the ones presented before.
The infrastructure is a D-type context, and therefore can be sketched by means of a "to be" *Trinity* model. Note, however, that the problem-solving process in this case is not "how to solve a specific indoor environmental problem", but rather "what is a feasible route to obtain an infrastructure in which indoor environmental problems in general are solved effectively and efficiently".

First we will describe two important problems that prevent indoor environmental problem solving to be effective and efficient at this very moment. After that, the main players in the proposed "to be" situation will be presented. Finally, the proposed situation will be sketched by means of a *Trinity* model (which is a small experiment in testing the use of *Trinity* in itself).

Two important problems

In indoor environmental problem solving, two of the three stages are difficult. The first stage is **diagnosis**, i.e. to obtain a descriptive "as is" model of the problem situation. The second is **finding a remedy**, i.e. to obtain a script that prescribes actions. Predicting "to be" in general is not so problematic, as the most important feature of the "to be" model is that the health effect should be absent.
The first problem is to **diagnose** the cause of the complaint. The "as is" models presented earlier in this chapter provided generic descriptions of problematic environments. These models made clear that a problematic indoor environmental problem is a complex system indeed. At this moment, the number of (non-psychosomatic) agents that may cause complaints in indoor environments is estimated to be several hundreds. This sheds light on the overwhelming magnitude of questions and observations that may be required to perform a thorough diagnosis. Figures 3-6 are valuable supportive tools in diagnosis: they guide and support the diagnostic process, and prevent a shallow diagnosis. A *first* step in diagnosis might be to use the complaints (and recent changes in the indoor environment)

[71] Indeed, it is a "to be" model!

to make a first selection from all these potential causes. A *second* step in diagnosis might be to verify whether source-agent-specific generic models of suspected agents enable one to describe the problematic situation "as is". This further reduces the number of plausible causes. Finally, a measurement to verify the actual presence of highly suspected agents might conclude and substantiate the diagnosis.
The second problem is to find an appropriate **remedy**. As was mentioned before, once having performed a thorough diagnosis, finding an *effective* remedy (a script that will work) is not so difficult (the most extreme option would be to remove the occupant from the building). However, finding a remedy that is *both effective and efficient* (efficient in the sense of acceptable, affordable, realisable, the best candidate, et cetera) is difficult indeed.

At this moment, the situation is that the specific type of remedy that is proposed largely depends on the person that is asked to find one, rather than on a thorough insight into the indoor environment of concern. Two examples will illustrate this problem.
The first example is concerned with a case of allergic asthma caused by house dust mite allergen.

A **physician** will come up with the remedy: take a medicine or a hypo-sensibilisation cure. The script branches off at rectangle 33 of figure 4.
A **ventilation expert** will claim that improving the ventilation system will reduce the humidity in the building, which will result in less mite growth.
A **vacuum cleaner representative** will maintain that a vacuum cleaner with an end filter (to prevent re-circulation of house dust mites allergen as part of respirable dust particles) will result in a significant improvement.
Someone **selling mattresses** will say that anti-mite mattress covers will be of great help, as they prevent the allergen from leaving the mattress. (Mattresses are an important source of house dust mite allergen: they are warm and humid, offer a habitat and contain skin parts. House dust mites love mattresses.)
An **expert in building constructions** will claim that changing the fundaments of the building will prevent water from entering the walls because of capillary suction (this causes additional growth of house dust mites). In addition, he might suggest a change in building regulations (which would prevent bad fundaments to be constructed in the future: a solution that will not help this specific complaining occupant).
A **firm in filters** might give the advice to use a respiration inhalation filter. In addition, this firm will back up the vacuum cleaner representative.
A **chemist** might claim that using an acaricide (acarus = mite) will stop the complaints.
A **neighbour** might give the advice to move to another part of the city.

The second example is a formaldehyde problem resulting from chipboard.

The same **building construction expert** from the first example might argue that the chipboard should not be present in the building at all. It should be removed.

A **roof specialist** might observe that the roof is leaking, and therefore the chipboard is humid. Repair the roof, and the dissociation process of ureum-formaldehyde will stop.
The **ventilation expert** in this case is not paid per ventilation system, but per satisfied customer. Therefore, she gives the advice to open the windows more often.
A **painter** might suggest to paint the chipboard with a sealing paint.
A **physician** might suggest to take an aspirin.

The problem is: to some extent *they all are right*. All the suggestions may contribute to minimising the complaints. The difficulty is that the perspectives and interests of the different "problem solvers" are too shallow to come up with a remedy that is efficient as well as effective.

Here also, the "as is" models (figures 3-6) and the generic scripts (section 8.4) are of great help. A diagnosis according to the generic model will prevent a shallow description of "as is". In addition, the presence of a library of scripts would help a problem solver in assessing the bones and merits of several options, and perhaps to construct a script that foresees drastic interventions only after less drastic interventions have failed. In the case of chipboard, for example, a first step might be to paint the chipboard (a physical domain approach) and to simultaneously adjust the ventilation behaviour of the occupant (tell him/her to open more windows more frequently: a knowledge domain approach). If this is not successful, the chipboard might be removed (first analysing an air sample would be advisable, though). In designing the script, important considerations would be costs, time, annoyance of the occupant and sustainability of the measures.

In summary: providing an indoor environmental problem solver with a library of generic "as is" models and scripts would address both the diagnostic and the remediation problem described above. The availability of such models facilitates both a thorough assessment of different candidate causes and a rational design of (combinatorial) scripts. More specific generic "as is" models and scripts should be available for each important source-agent combination (e.g. chipboard-formaldehyde).

The players

In principle, the proposed infrastructure concentrates on four major actors: occupants, problem solvers, information technologists and researchers.

Occupants are persons who live in the buildings of concern, and experience the health effects.
Problem solvers are persons who, as a result of their professional occupation, are the receptor of complaints. They should come up with a solution. Typical examples are physicians and employees of public health agencies. Problem solvers intensively use a knowledge-based system that is built on the basis of the generic models and that supports diagnosis (construction of an "as is" model) and on finding a remedy (construction of a script).

Researchers are responsible for construction and maintenance of generic models that describe problematic situations and prescribe remedies (i.e. scripts). They do so in close co-operation with indoor environmental problem solvers, as they are the ones that are confronted with actual problems and problem situations.
Information technologists (knowledge technologists) build and maintain the knowledge-based system mentioned above.

The model

A model of the proposed infrastructure is presented in figure 9. In *Trinity* terms, the model shows the global outline of a "to be" model. The model can be easily detailed and/or extended by means of *Trinity* modelling steps and modelling strategies as presented in Chapter 5. We did not do this, as figure 9 is only meant to be a sketch that illustrates the construction and use of the models presented in sections 8.2-8.4 in a problem-solving context.

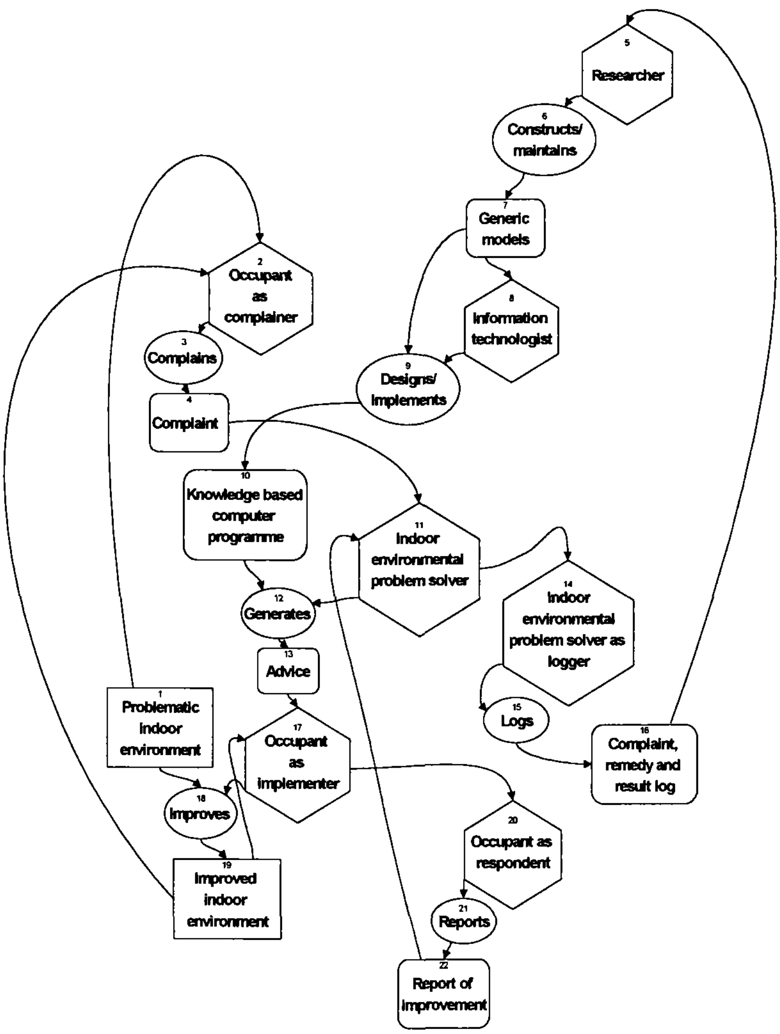


Figure 9: A proposal for an infrastructure that is based on model-based support for indoor environmental problem solving.

The knowledge-based computer programme is a tool that supports and enlightens the use of the generic models. In terms of functionality, the programme can be thought of as a library of generic "as is" models and scripts, a browser that supports surveying these models, tools that support the specialisation and combination of generic models (i.e. diagnostic and remedial aids), and an expert system that highlights plausible generic models on the basis of global information (like the type of complaints, a global description of the building, et cetera) and likely sources, agents and remedies. The intended user of this tool, called the indoor environmental problem solver in figure 9, typically is a general (medical) practitioner or an employee of a health organisation. The generic models provide a basis for such a tool, but may, for example, provide a basis for an educational programme or a public information campaign as well.

7.6 DISCUSSION AND CONCLUSIONS

In this chapter, we presented the use of *Trinity* models in indoor environmental problem solving. The construction process of a generic "as is" model was globally explained, and the use of generic "as is" models and generic scripts was illustrated. In addition, an infrastructure (represented by means of a *Trinity* "to be" model) was proposed, that utilises the models. This provides an example of the way in which *Trinity* can be used on different levels, addressing different problems.

The models were claimed and shown to be generic. But how generic is generic? The number of different agents that may cause complaints in indoor environments is estimated to be several hundreds (this includes biological, physical, chemical and particle agents). According to the experts that co-operated in this research, many (if not most) problematic situations are described in figure 3. We tested this in discussions for eight source-agent combinations. It may be expected that rather "unusual" sources and agents require small adaptations to be made to the model. This is exactly the reason why *Trinity* provides a library of modelling steps and a toolbox of modelling strategies. Generic models are a flexible aid in problem solving, rather than a straight waistcoat. They offer a conceptual vocabulary that is less fragmentary than problem situations in isolation, and at the same time more helpful than the general term "an indoor environmental problem situation". The "right" level of genericity or complexity can only be established in real-world indoor environmental problem solving. The generic models presented in this chapter are based upon the knowledge of experienced indoor environmental problem solvers. This may not be a 100 per cent guarantee for their applicability; it is, however, the best basis we have.

Actually, we presented a *hierarchy* of generic "as is" models. Figure 10 presents this hierarchy. At the highest level of abstraction, the generic "minimal environmental situation of concern" is situated. A less generic model is the "minimal indoor environmental situation of concern". This model was transformed into the generic "as is" model of figure 3: a model that is more detailed, but covers the same number of referents. "Source-agent" specific "as is" models (for example figures 4-6) are situated at a lower level. Situation-specific "as is" models are situated at the lowest level: they model a specific problematic indoor environment in a precise way. Such a hierarchy specialises models downwards, and generalises models upward. A step downward (i.e. a *specialisation*; a *refinement*) requires two modelling steps: first, the model is specified by means of a representation specification, and, second, the resulting set of model relations is restricted by means of a parallel restriction. A step upward (i.e. a *generalisation*) also requires two modelling steps: a parallel extension followed by a representation abstraction.

In addition, the generic "minimal indoor environmental situation of concern" was specified into the generic "as is" model of figure 3 by means of a series of parallel transformations. This shows the way in which different sequences of *Trinity* modelling steps may result in different models with different levels of genericity and/or detail. More generic models (higher models in figure 10) cover more referents at the expense of specificity. More detailed models (models at the right of a horizontal arrow, for example, figure 3) show more features of the referent, at the expense of simplicity.

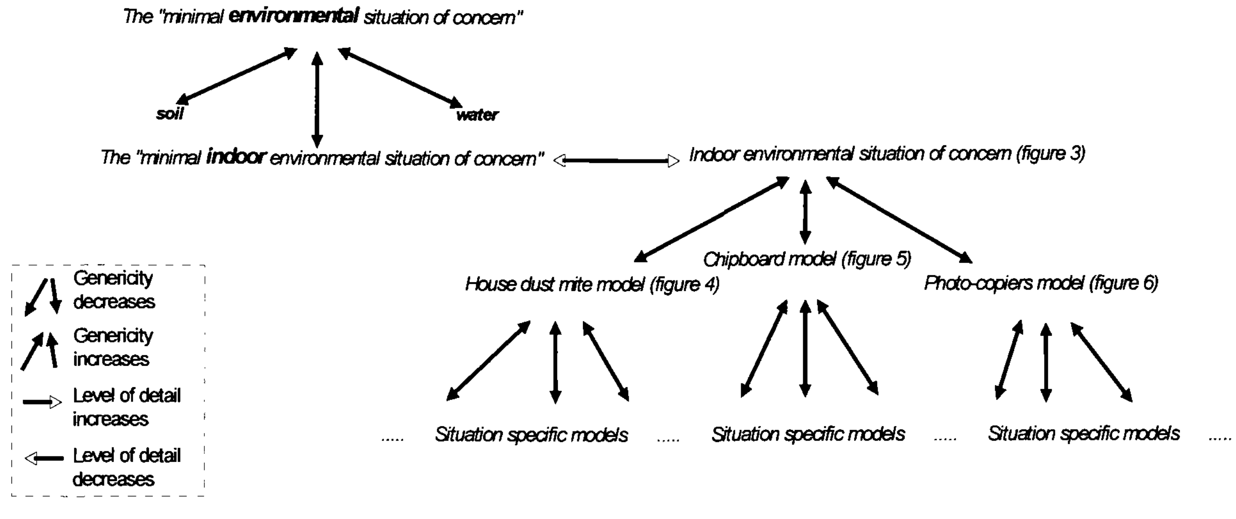


Figure 10: A hierarchy of generic "as is" models.

An important difficulty in indoor environmental problem solving is the fact that both a thorough diagnosis and an efficient remedy require an interdisciplinary approach. In too many cases, the diagnosis as well as the remedy depend on the problem solver, rather than on the problem situation of concern. The models presented in this experiment address exactly this difficulty: the availability of generic models prevents too shallow a diagnosis, and contributes to the design of both effective and efficient scripts. The models induce a bird's eye view, and enable a model-driven way of diagnosis and remediation. Rather than starting each time from scratch, the generic "as is" models provide a starting point for diagnosis, and the generic scripts are of help in actually removing the complaints. This facilitates a top-down strategy in indoor environmental problem solving: in this experiment, we intensively used *Trinity* representation modelling strategies (next to a parallel transformation strategy).

The proposed infrastructure addresses several other difficulties as well: it ensures that both state-of-the-art scientific knowledge and practical knowledge (feedback from the field) concerning indoor environmental problems is managed and maintained properly, and it foresees a knowledge-based system that helps a problem solver navigate through and use the generic models. This implies a knowledge management approach (see, for example, [Spijkervet and van der Spek (1996)]). Preliminary exercises in designing and implementing an indoor environmental problem solvers workbench ("BAS", not reported here) indicated that such a workbench is a feasible concept indeed [Diepenmaat and Leupen (1991)].

The answer to the central question of this chapter:

> *"Is it possible to support indoor environmental problem solving by means of Trinity models?"*

is: this is indeed the case. The presented models support someone responsible for diagnosing a problematic indoor environment. The indoor environmental expert most intensively involved in this experiment stated that the *Trinity* models highly structured his (existing) understanding of indoor environmental problem situations (as well as remedies) in important ways. As a matter of fact, he stated that the models provided him with a new way to think about the indoor environment. Notably, he mentioned explicitly the following ways of support offered by the models:

- During problem solving the models provide many "eye-openers" with respect to both cause(s) of an indoor environmental effect and remedies;
- In addition to this, the models emphasise a relative order in importance (upstream processes in the "as is" models are considered to be more fundamental causes, and therefore more fundamental branching points for the script);
- The models encompass different knowledge domains (traditionally belonging to different disciplines, and known by different persons), and as such contribute to preventing a too shallow diagnosis and remediation process;
- On top of material (physical) aspects of indoor environmental problem contexts, the model makes explicit the many different actors that (may) participate in both causing and remedying indoor environmental health effects. He considers this to be *very* important, as these persons in many cases are envisaged actors in executing the script;

- The models in combination support "zooming in and out", i.e. to decrease the scope of a model and to increase the level of detail, vice versa. The hierarchy of generic models, with at the "leaves" situation-specific models, is the background structure on which this zooming process takes place. He considers this feature to be very supportive in the convergent problem-solving process (he used the concept of a zoom lens to illustrate this).

Trinity enables one to construct clear descriptions of indoor environmental situations of concern that emphasise *alternative* points of applications of potential scripts (which can be modelled by means of *Trinity* as well). In addition, the role of the actors involved is highlighted. The possibility to distinguish both knowledge domain and physical domain approaches in remediation is an important strong point, as they require quite a different methodological approach: knowledge domain approaches typically are concerned with instruction, advice and information of the occupant; physical domain approaches typically require trained professionals to change the physical indoor environment. The difference between autonomous processes and intentional actions in the "as is" model clearly manifests itself in the type of approach that can be applied: an autonomous process typically is prevented by means of a physical domain script; an intentional activity typically is prevented by means of a knowledge domain script. A clear insight into several possibilities of intervention facilitates the design of scripts that are efficient as well as effective: the situation that the type of intervention is dictated by the type of discipline of the problem solver can be minimised.

CHAPTER 8

VOLATILE ORGANIC COMPOUNDS 2000

Henk Diepenmaat, Lars van Lierop and Chris Bruijnes

8.1 INTRODUCTION

In the period of May 1986 to February 1989, a policy programme was formulated to effect a reduction in emissions of Non Methane Volatile Organic Compounds (NMVOCs) into the air by industry, small businesses and households in the Netherlands. This programme is based on an agreement between the Dutch Ministry of Housing, Spatial Planning and the Environment, provincial and municipal authorities, and the corporate sector to effect a reduction in Volatile Organic Compound emissions into the air by at least 50 per cent in the year 2000 (compared to 1981). The main goal of the programme is to reduce the effects of ozone in the lower layer of the atmosphere (this tropospheric ozone causes damage to materials, plants and crops, and negatively affects public health).
The current strategy consists of reduction plans per sector. Examples of sectors are the graphical industry and (production and use of) paint. Such a reduction plan typically indicates the types of measures that should be implemented in (sub)sectors in order to reduce the VOC emission. The overall reduction potential based on all these measures is about 65 per cent
The implementation of the strategy is in an advanced stage. Many sectors are on schedule (the overall programme is rather successful). However, several sectors are behind schedule (in some of them even an absolute *increase* in emissions is noticeable).

We were commissioned by the Dutch government to conduct an analysis within four sectors, in order to obtain a clear picture of the difficulties and problem areas that these sectors experience in their attempts to reduce VOC emissions. In addition, difficulties and problem areas as experienced by civil servants (national, provincial and municipal) and official representatives of the lines of business of concern were to be considered. On the basis of the results of this analysis, additional policies might be developed, directed at eliminating these difficulties and problem areas, if possible.

We decided to support this real-world multi-actor problem-solving process, *in-line*, using *Trinity*. In line with this, the central research question of this experiment is:

> *"Does Trinity offer support in analysing this policy process, and in finding ways for improvement?"*

The problem owner we are supporting in this experiment is the ministry responsible for the VOC2000 national environmental policy process (and, more in specific, the person(s) responsible for managing the VOC2000 process).
In this chapter, we will present an overview of the way in which *Trinity* has been used. First we will give a *Trinity* interpretation of the problem context and the problem-solving process. Following this, the knowledge acquisition process, enabling us to construct a *Trinity* analysis model, will be outlined. This will show that knowledge acquisition processes, embedded in a *Trinity* approach, can be quite complex. After that, we will describe the way in which we turned the knowledge acquisition results into a *Trinity* model. Subsequently, we will show the way in which we used this model as a basis for deriving the contours of remedial actions (the contours of a script). Finally, we will discuss the ways in which *Trinity* offered support during these processes (which provides the answer to the research question), and end with a summary of the impact of our activities on the future of the *VOC2000* programme.

8.2 A *TRINITY* INTERPRETATION OF THE PROBLEM CONTEXT

In terms of *Trinity*, the (sector-specific) processes induced by the VOC agreement can be classified easily as D-type processes[72]. Many different actors play different and mostly complex roles. However, in a very strict sense, we were not asked to develop perspectives to guide these sector-specific innovation processes. Rather, we were asked to help in *improving several stagnating processes*. This stagnation manifests itself as an inability to reduce VOC emissions in accordance with the agreements.
This implies that both the "as is" and the "to be" model of the improvement perspective to be developed refer to a *complete* transition process and, therefore, each of them encompasses an "as is" model, a script and a "to be" model in turn. A reflective shift in problem context is at stake (see also section 2.5). Figure 1 visualises this shift in terms of a *Trinity* model.
To be even more precise: we were asked to conduct the first step of a *problem-shooting* approach, where emphasis is on an analysis of the stagnating situation "as is". In addition, the contours of the script were to be explored.

[72] Actually, the problem owner claimed that in order to classify his problem, it would be necessary to extend the "ABCD" typology with "E-type" problems: Extremely difficult D-type problems.

The "role switching" from a complete intentional activity towards the referent of an "as is" model (see figure 1) may seem confusing at first, but is rather typical for attempts to intentionally improve D-type processes. A reflective shift in problem context implies a different intention. This shift in intention in this case is from "realising innovations" towards "improving a stagnating innovation process". This shows that models of situations "as is", scripts and models of situations "to be" are demarcations of parts of the real world that serve a purpose (the problem owner's purpose). They are mental artefacts in a very literate meaning.

The experiment to be described encompasses predominantly an *analysis of the stagnating transition process*. Scripts and "to be" models will be constructed in the (near) future (first steps will be presented in section 8.5).

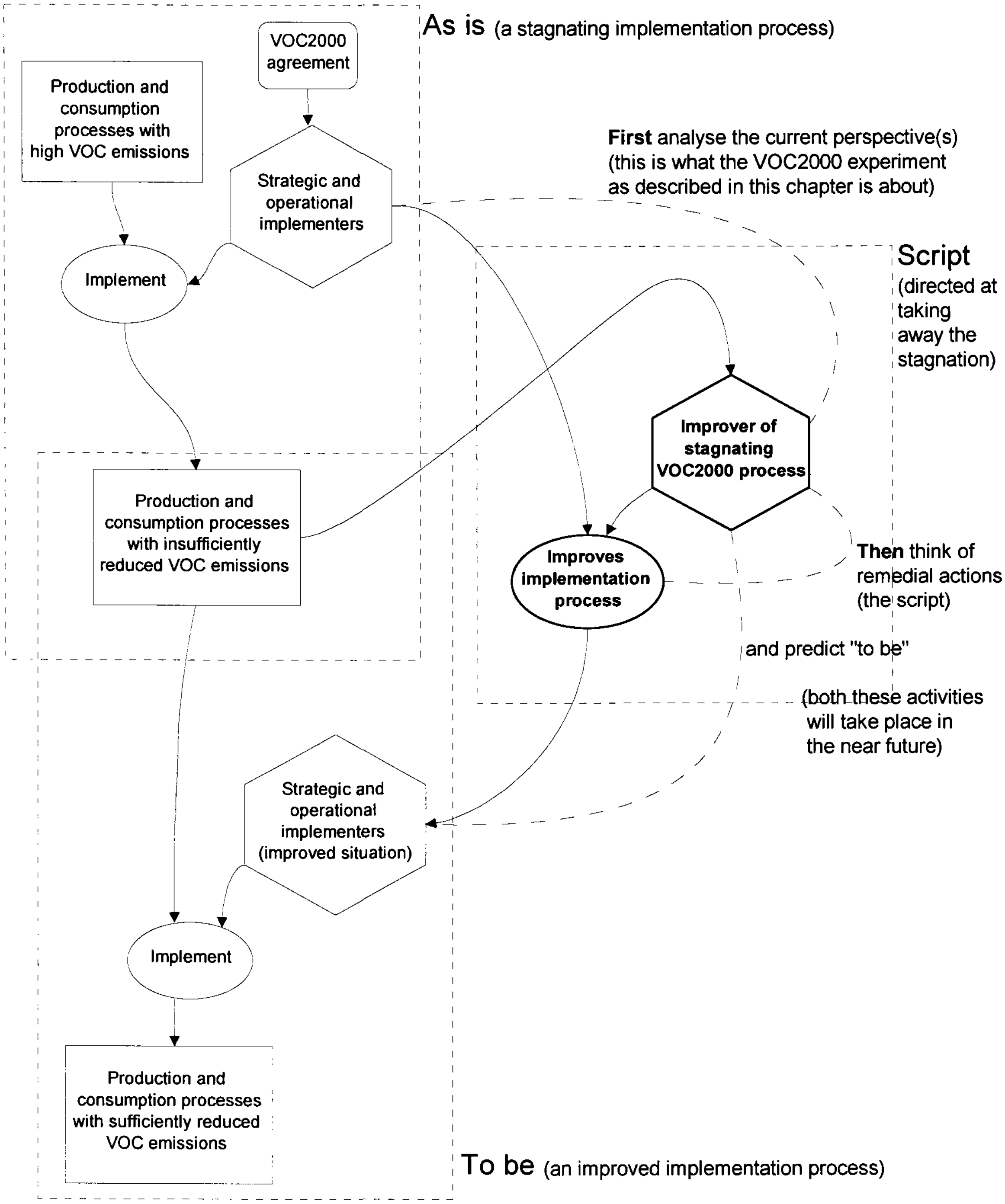


Figure 1: An overview of the *VOC2000* experiment in *Trinity* notation.

8.3 THE KNOWLEDGE ACQUISITION PROCESS

Analysing the stagnating "as is" situation is not easy. The VOC2000 programme has a voluntary basis, rather than a "command and control" basis. Industries, to a large extent, are flexible in their approach to meet the agreements. In addition to this, the scope of (even sectoral parts of) the programme is large. This makes it difficult to obtain a clear and coherent overview. Especially if the goal of the agreement is endangered and improvements are required, this lack of overview turns out to be problematic.

As a knowledge acquisition strategy, we decided to interact intensively with *field players*, as they are the actors who should actually change their behaviour, and therefore the very actors who actually experience the reasons for stagnation (the problem areas). In a sense, we tried to "reverse-engineer" the perspective underlying the stagnating transition process *on the basis of problem areas, as mentioned and prioritised by field players*.

The knowledge acquisition process will be described in more detail below,but prior to this we want to discuss the reverse-engineering process. We observed the stagnating transition process through the eyes of field players. It should be mentioned that this "lens" is not without aberration.
First, this implies that our analysis is based on *opinions* (of field players). These opinions in turn are based on a mixture of experiences and expectations. It is impossible to completely prevent worst-case expectations of individuals, that never have occurred in practice, to slip into the analysis (although intensive cross-validation of problem areas, see section 8.3, is likely to minimise this phenomenon). Opinions may be questioned. However, the problem areas as experienced by partners in complex transition processes should better be taken seriously, rather than be question-marked in advance. After all, opinions motivate and guide actions.
Second, our knowledge acquisition process is directed at establishing problem areas. Therefore, non-problematic parts of the overall process, although important in obtaining a coherent overview, will not be emphasised.
Third, the problem areas will have to be clustered and abstracted several times during the process, as we are aiming at an overall picture (see also section 8.4). This implies that the resulting models will be abstract too. This, however, is useful, rather than problematic, as the resulting "as is" model will be used as a basis for additional policies and remedial actions. Such a first basis should not be cluttered with detail. Specification will be appropriate at an "if-needed" basis at a later stage.

The knowledge acquisition process that we designed and used, as well as its detailed intermediate and final results, are described in depth in [Diepenmaat (ed) et al. (1997)]. Here we will restrict ourselves to an outline of the knowledge acquisition plan and a description of its results.

Experiments

A knowledge acquisition plan was formulated that consists of six stages[73]. They are:

Stage 1: Select sectors of interest. This reduction was considered to be necessary because of the large scope of the VOC programme. Four sectors were retained; selection criteria were (a lack of) compliance with target reductions in VOC and relative contribution to overall emission of VOC in the Netherlands.

For each sector:

Stage 2: Make an inventory of experienced problem areas. The goal of this stage was to obtain a raw inventory of problem areas in realising VOC reduction, as experienced or expected by field players belonging to the sectors. This step consisted of a series of about fifty bilateral interviews. Interviewees were representatives of the Dutch corporate sector, branch representatives, company representatives, and national civil servants participating in the VOC2000 programme. In addition, we interviewed environmental permit licensers (local civil servants), as they play an important role.

Stage 3: Structure raw inventory of problem areas per sector. On the basis of the interview reports of stage 2, problem areas were extracted from the interview reports per sector, clustered if necessary, and each of them was represented in a structured frame format. The frame format highlights specific attributes of a problem area, such as a further explanation, its cause, the process in which the problem area manifests itself (for example, production, sales), the actors involved in this process, and potential solutions. (Note that this frame format, in essence, is a rudimentary precursor of the intention list as presented in section 5.6.)

Stage 4: Allow for feedback and prioritise per sector. The formatted lists of problem areas per sector were mailed to each of the interviewees of this sector, who were allowed to comment on them and refine them, and were offered the opportunity to add problem areas still missing. In addition, the interviewees were asked to rate the problem areas on a scale from 1 (hardly important) to 5 (very important) in terms of their negative influence on implementing VOC reduction measurements. Note that this implies that *all* the problem areas mentioned *within* a sector were presented to *all* the interviewees *of this sector* (i.e. a

---

[73] Actually, this was the first time we used *Trinity* as a means to support the *design of (part of) a problem-solving process*. The research approach was a result of a modelling process (and the debate paralleling this process) conducted by two members of the research team (a small-scale participative mode), the process was fuelled by discussions with other research team members (hidden mode) as well. The six stages, actually, are an abstraction of a throw-away *Trinity* model (that indeed has been thrown away). At the moment of finishing this dissertation, we consider having thrown the model away to be a big mistake: designing problem-solving processes supported by *Trinity* is both a useful and an interesting enterprise, worth experimenting, and a dissertation in itself. At that moment, however, we did not yet consider this to be an interesting line of research, and, in addition, we had a strict deadline to meet.

cross-validation and prioritisation within each sector). As a result, we obtained an impression of the importance of problem areas experienced per sector.

For all sectors:

Stage 5: Plenary prioritisation. On the basis of the sector-specific, prioritised formatted lists a gross (overall) list of some 25 problem areas was constructed by means of a hierarchical clustering procedure. During a plenary workshop, to which all the interviewees of the sectors were invited, these problem areas were discussed and a prioritisation procedure was conducted once again.

Stage 6: Report research activity. The results of this stage are reported in [Diepenmaat (ed) et al. (1997)].

The knowledge acquisition process is visualised in figure 2. Three stages encompass intensive interaction with field players (these stages are shaded).

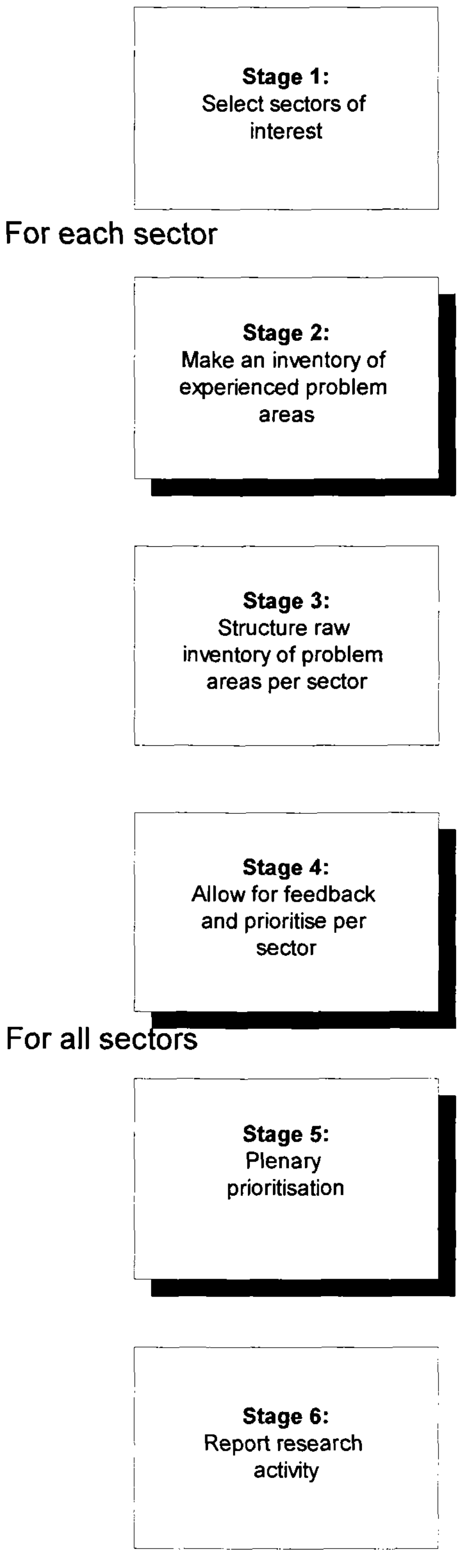


Figure 2: Stages in the knowledge acquisition process. Stages that encompassed intensive interaction and communication with field players are shadowed.

The main result of carrying out this knowledge acquisition procedure is a broadly accepted, thoroughly reviewed, prioritised *list of 25 generic problem areas* that prevent, or at least hinder, achieving the goal of the *VOC2000* agreement.

8.4 FROM PROBLEM AREAS TOWARDS A *TRINITY* MODEL

We will not go deeply into the overall results of the knowledge acquisition process, but will restrict ourselves to some examples of the way in which we operated in constructing the *Trinity* model. The general procedure was as follows[74]:

For each problem area on the list:
  translate problem area to actors
  For each actor:
    describe roles and related actors
    On the basis of all actors and roles:
      construct *Trinity* model

The actors and roles were derived from the problem areas. This is an interpretation step that took place on the basis of the understanding of the overall process that we acquired during the knowledge acquisition process. Remember that originally the interviewees were selected *because* they play (a) role(s) in the transition process (they are field players). This interpretation step in a sense highlights their specific roles[75]. For persons unfamiliar with the *VOC2000* field this process may not be straightforward in all cases.

The procedure will be illustrated by means of two examples from the list of 25 problem areas. Table 1 shows (a concise version of) the two example problem areas, as well as the actors and roles that we derived from them.

---

[74] As was explained in section 5.6.1, some iteration followed these initial steps.

[75] In the near future we will construct more detailed (sub)sector-specific *Trinity* models. During the construction of these models, we intend to make a more intensive use of "participative mode", in order to obtain more feedback.

Table 1: Examples of *VOC2000* problem areas and the actors that were derived from them.

Problem area 1: Acceptance of changes in working methods and/or product quality. Changes (not necessarily a deterioration) in working methods for product and/or product quality are not accepted by retailers and (intermediate and end) users of these products.

Actor: **Retailer or upgrader**
Role(s): Upgrades, retails, markets[76] high-VOC product alternative (in situation "as is"); Is reflecting on low-VOC alternatives; Upgrades, retails, markets low-VOC product alternative (in situation "to be")
Related actors: Producers (and end users)

Actor: **Producer**
Roles: Produces high-VOC products ("as is"); Reflects on low-VOC alternatives; Produces low-VOC products
Related actors: Deliverers of machines and other resources; Retailers and Upgraders; Decision maker within company

Actor: **Consumer**
Roles: Consumes high-VOC products ("as is"); Reflects on low-VOC alternatives; Consumes low-VOC products
Related actors: Retailers and Upgraders

Problem area 2: The quality and homogeneity of permitting environmental licences is precarious.

Actor: **Company employee responsible for environmental permits**
Role: Negotiates permits with local government official
Related actor: Local government environmental permit licenser

Actor: **Local government environmental permit licenser**
Role: Negotiates permits
Related actor: Company employee responsible for environmental permits

Actor: **Informer of local government representative**

---

[76] The process of adding value to this product is called *upgrading* a product, for example by using it as a half-product, by packing it, et cetera. The process of transporting and offering products is called *retailing*.

Role: Informs local government representative with respect to process and product alternatives at company level, in order to further their dissemination
Related actor: Local government environmental permit licenser

Actor: **Preserver of homogeneity of environmental permits**
Role: Preserves quality and homogeneity of Dutch environmental permits
Related actor: Local government environmental permit licenser

---

In principle, more specific roles might be derived from these problem areas (which would require additional knowledge acquisition steps). However, we tried to be as generic as possible, and not to include situation-specific considerations.

Figure 3 shows the *Trinity* model that we made on the basis of the list of actors and roles, derived from all 25 problem areas. During the modelling process we predominantly used a parallel building blocks strategy. In addition, we occasionally abstracted some roles (by means of parallel abstraction steps) in order to obtain the minimal model that enables us to situate all the 25 generic problem areas of the list, albeit not in the greatest detail.

We were rather surprised to see that constructing this model did not require any additional knowledge acquisition steps: it was constructed solely on the basis of the information about actors and roles, derived from the list of problem areas. As the resulting model turns out to be a coherent whole, apparently, almost every process in the transition process is more or less problematic (which is not a very reassuring observation).

From a practical point of view, however, the key question is: in what ways is this "as is" model helpful in finding ways to improve the stagnating process? This issue will be addressed in the next section.

8.5 FROM *TRINITY* MODEL TOWARDS RECOMMENDATIONS

In this section we will explore the contours of the script model of the improvement process, and will show the ways in which the *Trinity* model is of help in this respect.
The list of problem areas, resulting from the knowledge acquisition process, covers an extremely wide scope, from technical problems, via global competition of companies settled in different countries, to badly informed and (presumedly) unwilling consumers (and many more). An interesting observation is that our modelling activities, on the basis of these problem areas, had an enormous impact on our level of understanding the relations between the problem areas, as well as on the transition process as a whole. The model makes very clear that problem areas should not be addressed in isolation, as many of them are causally linked. As a general remark this is not a very unexpected one, *but the model shows,*

in terms of actors and actions, the ways in which these linkages manifest themselves. This offers important handles for future improvement. Two examples will substantiate this.
For example, one of the problem areas is that detailed financial information concerning specific measures is found to be lacking. The model shows that such information is to be gathered by a financial specialist (a role, not necessarily a separate person), who is part of a team drawing up an innovation plan. This team predominantly consists of employees of the companies of concern: *they* are responsible for the financial quality of innovation plans, and *not* external actors. Therefore, a feasible route to address this problem area is to *facilitate and support innovation teams in their activities*. This should best take place "in line", as this information is typically very case-specific: generic approaches are likely to provide information that is far too general (this was mentioned indeed several times as a complaint). If such innovation teams prove to be absent, then *that* is the very problem that should be tackled first (the model suggests to focus in this case on decision makers within the companies, as *they* are the requesters for innovation plans).
The second example is that the model is very specific in that producers, retailers/upgraders and consumers *all three* have to change their behaviour *in concert*. This is especially manifest for the producers, retailers/upgraders and consumers. If either one of them falls short, the efforts of other ones are bound to fail as well. This feature of D-type change is not addressed explicitly in the current VOC2000 reduction policies. A well-balanced strategy should address (the problem areas related to) all three of these groups simultaneously, in a coherent fashion.

Generalising these observations (and many more that are not reported here): one thing that especially became clear is the far from optimal correspondence between the policy instruments, traditionally used to stimulate the *VOC2000* processes, and the problem areas as resulting from the knowledge acquisition process. Traditionally, the instruments in use mainly are the negotiation of clear sector-specific reduction targets, the organisation of technical committees to prepare information of a rather general technical nature, and, in only a limited number of cases, the provision of financial support for trial innovations. Such a non-optimal correspondence in principle can be identified on the basis of a direct comparison of the list of problem areas with policy instruments in use: the model is not required *per se*. However, the model, in addition, shows very clearly that policy instruments in use predominantly affect the *first steps* in the overall process (notably, the upper left part of the script). And many, if not most, of the problem areas are located in the other three quarters of the script. The policy instruments in use at this moment give an *initial push* to the overall transition process. Both the *Trinity* model of the process, as well as the stagnation in some of the sectors, show that giving an initial push is simply not enough, as many problem areas reside elsewhere (every process in the model covers one or more problem areas), and their specific nature is both (sub)sector-specific and in addition only poorly understood. The policy instruments push a cloudy, and at times rather refractory, system. Pushing clouds may be difficult, but pushing even harder is certainly not a solution to this.

Two remedies that were mentioned several times by public servants are: a) increase the effort in providing generic information (which comes down to increasing efforts in the (upper) left part of the script even further), and b) regard legislation as an alternative for the agreement-based approach (which is assumed to increase the pressure on corporate circles). However, the *Trinity* model shows an important weak spot in the present approach, and (therefore) suggests quite a different direction for improvement. This weak spot is the lack of a clear view at, and the absence of, policy instruments supporting **the transition processes as a whole** (notably the other three quarters of the model). Figure 3 presents a very general picture of the overall transition process, and provides a first step in this respect. In specific sub-sectors the models can and should, however, be more specific and more refined, in order to support the identification and further elaboration of remedial and supportive actions. Only if the process is clear, well balanced and coherent, remedial and supportive actions with respect to problem areas can be planned. This is the very key towards process improvement.

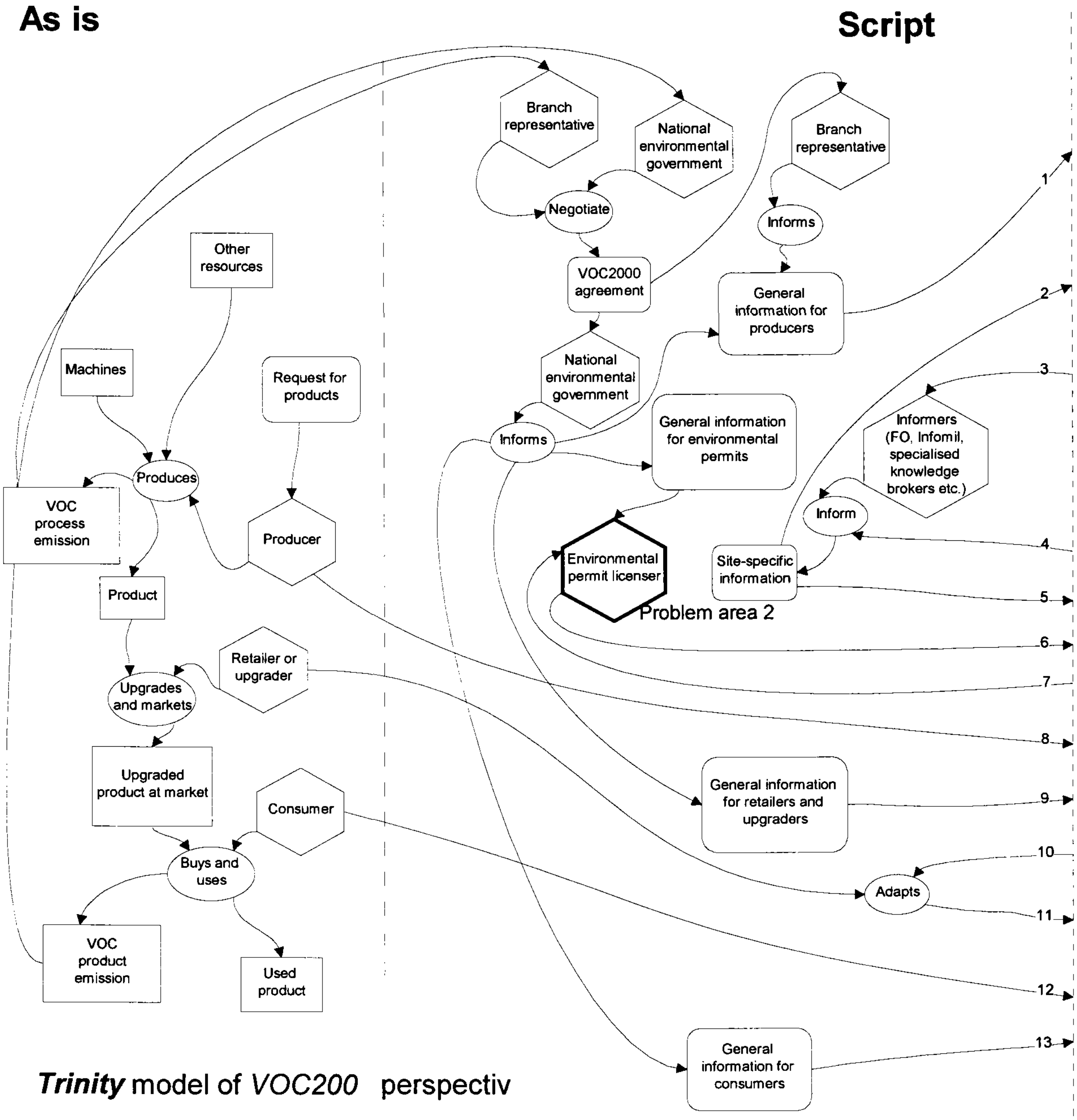
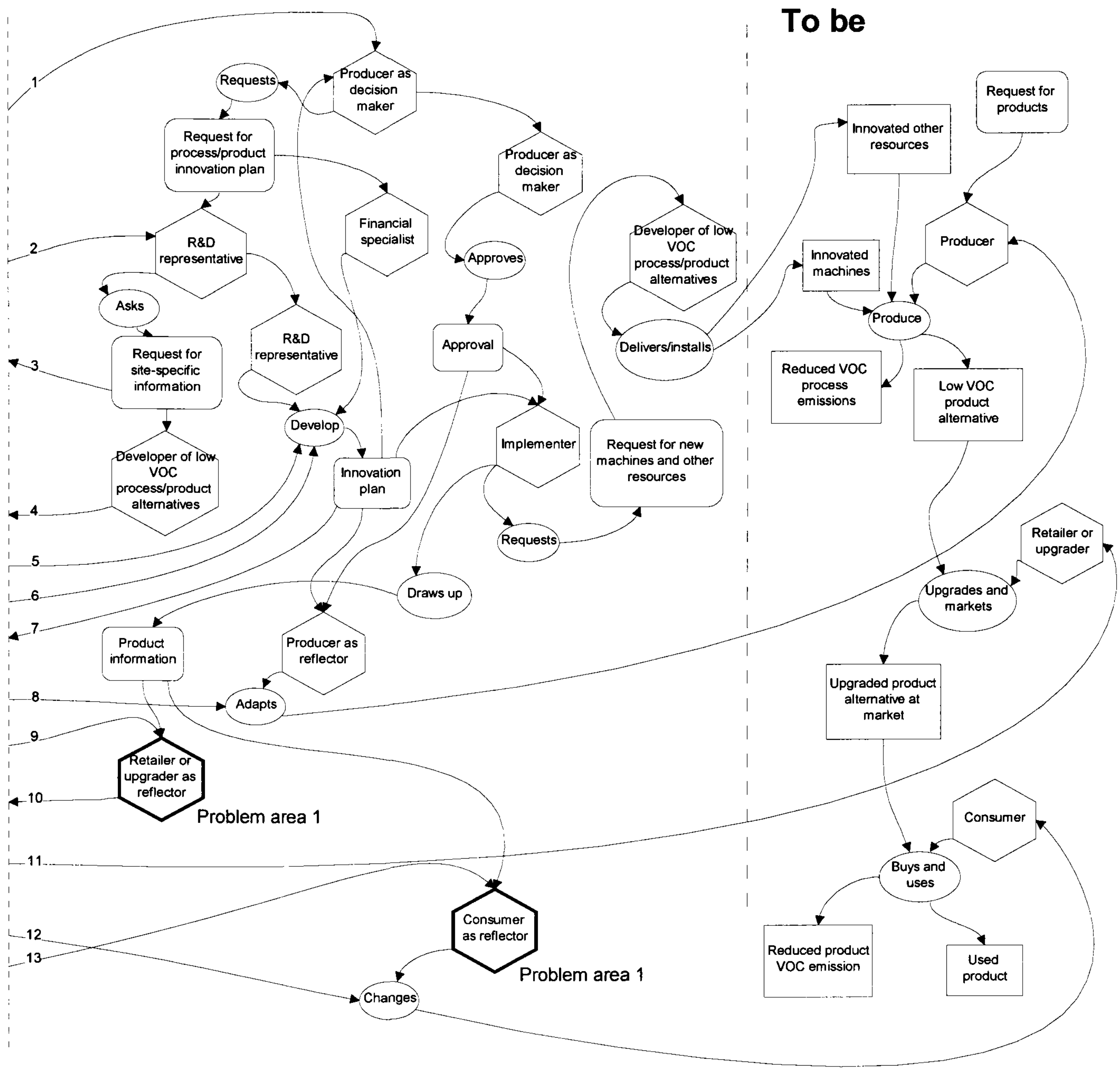


Figure 3: A *Trinity* model of the *VOC2000* process.
The bold lined areas indicate where the example problem areas 1 and 2 are situated. This model is based on the actors and roles, as derived from the list of 25 generic problem areas. Therefore, *all the activities in the script part* are considered to be more or less problematic.


To be
1
2
3
4
5
6
7
8
9
10
11
12
13
Requests
Producer as decision maker
Request for process/product innovation plan
Producer as decision maker
Financial specialist
R&D representative
Asks
Approves
Request for site-specific information
R&D representative
Approval
Develop
Developer of low VOC process/product alternatives
Innovation plan
Implementer
Request for new machines and other resources
Requests
Draws up
Product information
Producer as reflector
Adapts
Retailer or upgrader as reflector
Problem area 1
Consumer as reflector
Problem area 1
Changes
Developer of low VOC process/product alternatives
Delivers/installs
Innovated other resources
Innovated machines
Request for products
Producer
Produce
Reduced VOC process emissions
Low VOC product alternative
Retailer or upgrader
Upgrades and markets
Upgraded product alternative at market
Consumer
Buys and uses
Reduced product VOC emission
Used product

In line with this, as a first step we recommended the selection of specific sub-sectors where substantial VOC emissions are manifest. Directly following this first step, a thorough investigation, for each of these sub-sectors, of the (either forthcoming or ongoing) implementation processes as a whole should be conducted. Completely realising the goals of the *VOC2000* programme requires a close co-operation between public authorities and corporate circles. For the government, this implies that emphasis should shift from generic views toward sub-sector specific views, and from pushing clouds towards participating in, facilitating and supporting well-known processes[77] (see also [Cramer (1991) p. 35]). These recommendations were accepted by the national steering committee of the VOC2000 policy process, and will form the basis for future activities. Figure 3 and the discussion above show that *Trinity* is likely to offer important support in these future activities as well.

8.6 DISCUSSION AND CONCLUSIONS

In this chapter we described the way in which *Trinity* supported (the first steps in) attempts to improve a Dutch plan for reduction of VOC emissions; in some sectors this plan is not working out very well. The problem context of the *VOC2000* programme is enormous indeed: it covers important parts of the Dutch (and even international) production and consumption processes. As a result, the problem areas that we established, as well as the "as is" model (figure 3) that we constructed, are at a rather high level of abstraction. Notwithstanding this, they provide a profound and clear understanding of the overall transition process.

The *VOC2000* experiment shows that use of *Trinity* can be easily integrated with a research activity, that otherwise would have been conducted without model-based support (and in that case would have stopped with the list of generic problem areas as its main result). We simply inserted the "business as usual" approach, as described in section 8.3, as a complex and highly interactive knowledge acquisition step in a *Trinity* scheme of operation. This implies that we did not confront field players with the *Trinity* models (which amounts to *hidden use*).

We used the concept of generic problem areas as a "lens" to observe the process. Considering the fact that we were engaged in the first step (analysis of "as is") of a *trouble-shooting* approach, this turned out to be a very effective lens. For script synthesis and "to be" prediction, which will take place in the near future (see also figure 1), such a lens is expected to be less appropriate, as experience with the future is a rare phenomenon indeed.

As was explained in section 8.5, the (construction of the) *Trinity* model was very supportive in obtaining a coherent overview of the problem areas in combination, and a

[77] The current strategy may be characterised as an "agree, inform, wait and see" approach. The envisaged strategy is an "agree, participate and facilitate, and succeed" approach.

deeper and better understanding of the underlying transition process as a whole. Therefore, our general conclusion of this experiment is that using *Trinity* made a big difference in comparison with the "business as usual" approach (ending with the list of problem areas). It may be argued that other modelling methodologies (for example Systems Dynamics, see also Chapter 11) would also have helped in this respect. A distinguishing feature of *Trinity* models, however, is that they refer to actors and (in this case predominantly intentional) activities. We consider this to be of great importance, as intentionally changing a D-type system requires that many different actors act in concert. Therefore, actors and actions should be referred to as explicitly as possible in the models. And this is precisely what *Trinity* models do.

At a more detailed level, *Trinity* supported our activities in the following manners.

First, it was very clear from the start what should be the goal of our activities (an "as is" model of the transition process) and what intermediate steps should be taken to realise this goal (to draw up lists of actors and roles). This may seem trivial, but we experienced this as an enormous support, considering the overwhelming complexity of the *VOC2000* agreement and the processes in which it results.
Second, the (construction of the) *Trinity* "as is" model proved to be very supportive in obtaining a coherent understanding of the stagnating processes as a whole. At first, we considered the list of generic problem areas to be a valuable representation of the process. However, constructing the model on the basis of this list made clear that this was a serious misconception. Apparently, there is quite a difference between a list representation of problem areas and a coherent model of the underlying intentional processes as a whole (turning the list into the model was not a trivial enterprise!). Lists simply are not very supportive[78] in understanding the network relationships between its elements. Especially in D-type processes, therefore, the step from lists towards process views is of eminent importance (see also section 5.6, table 4). Many problem areas turned out to be causally related, and remedies that appeared to be quite to the point from a superficial point of view (like increasing the effort in providing general information) after a more thorough analysis turned out to be rather circumstantial.
Third, the model proved to be of great help in defining the contours of future remedial actions. Its construction made us realise that increasing the efforts to give an "initial push" to the process is not likely to result in a lot of improvement, as, at a lower systemic level, each of the constituting transition processes is rather unique, exhibits specific manifestations of (several of) the 25 generic problem areas, and therefore requires *specific* support *throughout* the transition process. This last requirement is quite in contrast with the policy instruments currently in use in the *VOC2000* process.

[78] This perhaps is an understatement.

In conclusion, our answer to the central research question of this chapter:

> *"Does Trinity offer support in analysing this policy process, and in finding ways for improvement?"*

is yes. The model offers a coherent overview of the transition process as a whole, and points out a discrepancy between the policy instruments in use at this moment and the position of problem areas in this process, which opens several routes to improve the process. This was confirmed by the problem owner, who in addition mentions the following processes in which *Trinity* offers him support:

- Identification, selection and characterisation of relevant actors;
- Understanding the relationships between these actors more explicitly than before;
- Understanding chain effects when considering interventions in the network;
- Identifying "the crucial" problem areas;
- Legitimisation of (potential) policy measurements and other actions;
- Consolidating one's position in negotiations.

The problem owner intends to use *Trinity* in the near future to support his management activities at a more detailed level (i.e. subsector-specific) as well.

The *VOC2000* experiment made us realise, once again, the sometimes extreme complexity of D-type intentional activities. Nonetheless, by means of *Trinity* we were able to deal with this complexity. We were able to identify several routes towards improvement, that were formulated as recommendations and presented to the *VOC2000* steering committee. These recommendations were accepted and, as a result of this, in the near future we will try to support *VOC2000* processes on the basis of a more thorough understanding of sub-sector specific transition processes *as a whole*. We will use this understanding to act as a participant and process facilitator. On the basis of the experimental results described above, we will continue using *Trinity* as an important means to support these activities.

CHAPTER 9

THE STRATEGIC CONFERENCE "BUILDING AND DEMOLITION WASTE"

9.1 INTRODUCTION

The use of primary resources (primary building and renovation materials) in the building sector is considerable. In addition, a substantial amount of building, renovation and demolition waste is generated. For this reason, the responsible ministries in the Netherlands (i.e. Economic Affairs and Housing, Spatial Planning and the Environment) took the initiative to organise a strategic conference in order to stimulate the development of knowledge and technologies that specifically address these issues. In more specific terms, the goal of this strategic conference was to bring together representatives of the relevant actors, to discuss the problem as a whole, and to identify opportunities and problems so as to improve the situation. In addition, the goal was to start concerted actions that specifically address these opportunities and problems and would lead towards a better future situation. This strategic conference was one of a series addressing different environmental topics (including packaging waste, and recycling of metals), and took place on March 21, 1995.

The corner stone of the strategic conference was a document reflecting the results of a preliminary investigation [BEA (1995)]. Predominantly on the basis of this preliminary study, a list of potential (technological) options to take action was composed that functioned as an input document of the conference. During the conference, the options on the list were discussed, a selection was made, and agreements were concluded about future concerted actions.

At the moment that the experiment described in this chapter started, the conference had already taken place. Rather than participating in it, we analysed parts of the material of the strategic conference *ex post*. In addition, we had interviews with responsible policy makers. This (i.e. ex post) is not the way we feel that *Trinity* should be used. However, at that moment field experiments in using *Trinity* were scanty, and as a result a slight hesitation existed to use *Trinity* directly "in line" in future strategic conferences. Therefore, the experiment was designed as a test case for future use in forthcoming strategic conferences. In line with this, the central question of the experiment is:

"Does the use of Trinity in strategic conferences result in added value?"

The problem owner we are supporting in this experiment is someone responsible for organising (successful) strategic environmental technological conferences. We are aiming in this experiment at establishing whether *Trinity* is a suitable instrument to support such conferences in the future.

We will try to answer the central question by means of reviewing parts of the strategic conference *Building and demolition waste*, and the construction of *Trinity* models.
The goal of this experiment is not to criticise the strategic conference, nor the persons and organisations that organised it and participated in it. The focus is on answering the central question, and we merely use the conference as a means to do so. We highly appreciated the availability of the documents. Especially the description of the building and demolition sector in 2010 in the background document proved to contain very valuable material for our modelling processes. Nonetheless, for reasons of privacy, actors that are represented in the models are anonymous: they are referred to in a functional way (in terms of their role) rather than as an identifiable (group of) organisation(s) or person(s).

Our claim is that the use of *Trinity* supports and improves the preparation as well as the organisation of strategic conferences. We claim this because strategic conferences, in *Trinity* terminology, are directed at obtaining a D-type perspective and starting the implementation of the script part of this perspective (see also the description of the goals of the conference presented above). The background intention of a strategic conference is to obtain a thorough understanding of the situation "as is", to explore potential futures ("to be's"), and to come up with a script that is expected to result in a better future. An additional goal of the conference is to start execution of this script. The fact that the perspective of concern in this case is long term and strategic should not affect the applicability of *Trinity*: the major requirements are that a problem owner and a D-type intentional activity can be distinguished. These requirements are met.

Figure 1 presents the problem owner of this experiment in his/her context. In terms of Chapter 3, the intentional activity is a mixture of a curative and a preventive activity: the situation is problematic, and is bound to become even more problematic in the future. Figure 1 highlights the order in problem solving adopted in this experiment; after a global analysis of "as is", emphasis rather radically should shift towards thinking of better futures ("to be's"). Only after that should attention shift towards the construction of scripts that are expected to realise the transition. This amounts to a *back-casting* approach [Quakernaat (1995)]: an approach in strategic explorations in which the focus is on better futures first, and only after one or several better futures have been "designed", is focus directed at feasible scripts that enable one to attain these futures. A strong point of a back-casting approach is that it favours thinking about non-evolutionary (large-step) transitions: the remark "that cannot be realised" is avoided until later stages in the problem-solving process, which results in a rather explorative way of problem solving[79].

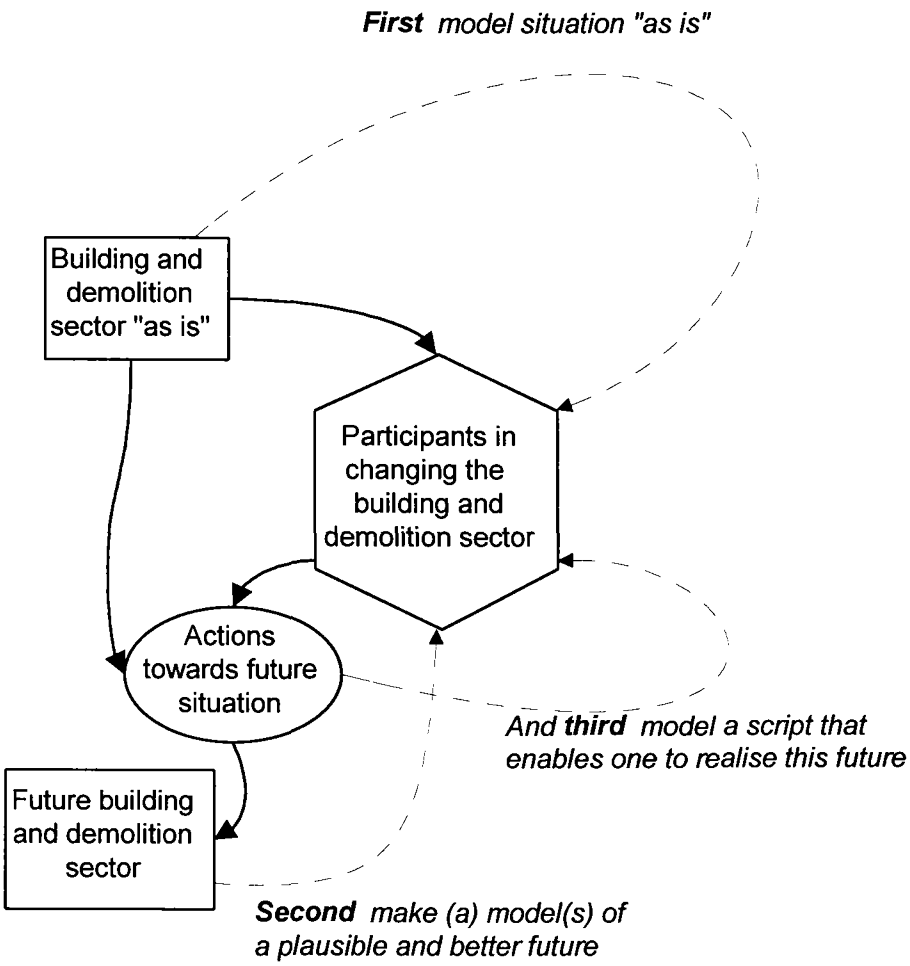


Figure 1: The strategic conference *Building and demolition waste* in its problem context. We applied a *back-casting* approach (the sequence in problem solving is: only globally analyse "as is", then emphasise thinking about "to be's", and finally construct script(s)).

[79] In disfavour of this approach it is mentioned that focusing on "to be" may result in daydreaming. However, development of the corresponding script and "as is" model is not neglected, but merely postponed.

Section 10.2 presents an essential *Trinity* model of the building and demolition sector "as is". In section 10.3, both a global and a more detailed "to be" model of this sector are presented. The year of reference is 2010. In section 10.4, focus is on the transformation process (a script). Section 10.5 generalises the conference-specific results: what are the implications of this experiment for other strategic conferences? In addition, the overall results are discussed and conclusions are drawn.

9.2 A GLOBAL "AS IS" MODEL OF THE BUILDING AND DEMOLITION SECTOR

Figure 2 presents an essential "as is" model of the building and demolition sector. The model is essential in that it focuses on the three core intentional activities: building, renovation and demolition. The grey rectangles highlight primary resource use, the black rectangles highlight generation of waste. The width of arrows suggests a relative importance (this is not an official *Trinity* syntactical convention).

Figure 2 is exaggerated, as at this moment some 60 per cent of the waste generated in building, renovation and demolition is re-used in other processes, albeit not always in an optimal way from an environmental point of view (for example, demolition waste may be used for the foundation of a highway). Nonetheless, it clarifies the essentials of the problem situation: many primary resources are used, and much waste is generated (see also the problem context description in the introductory section of this chapter).

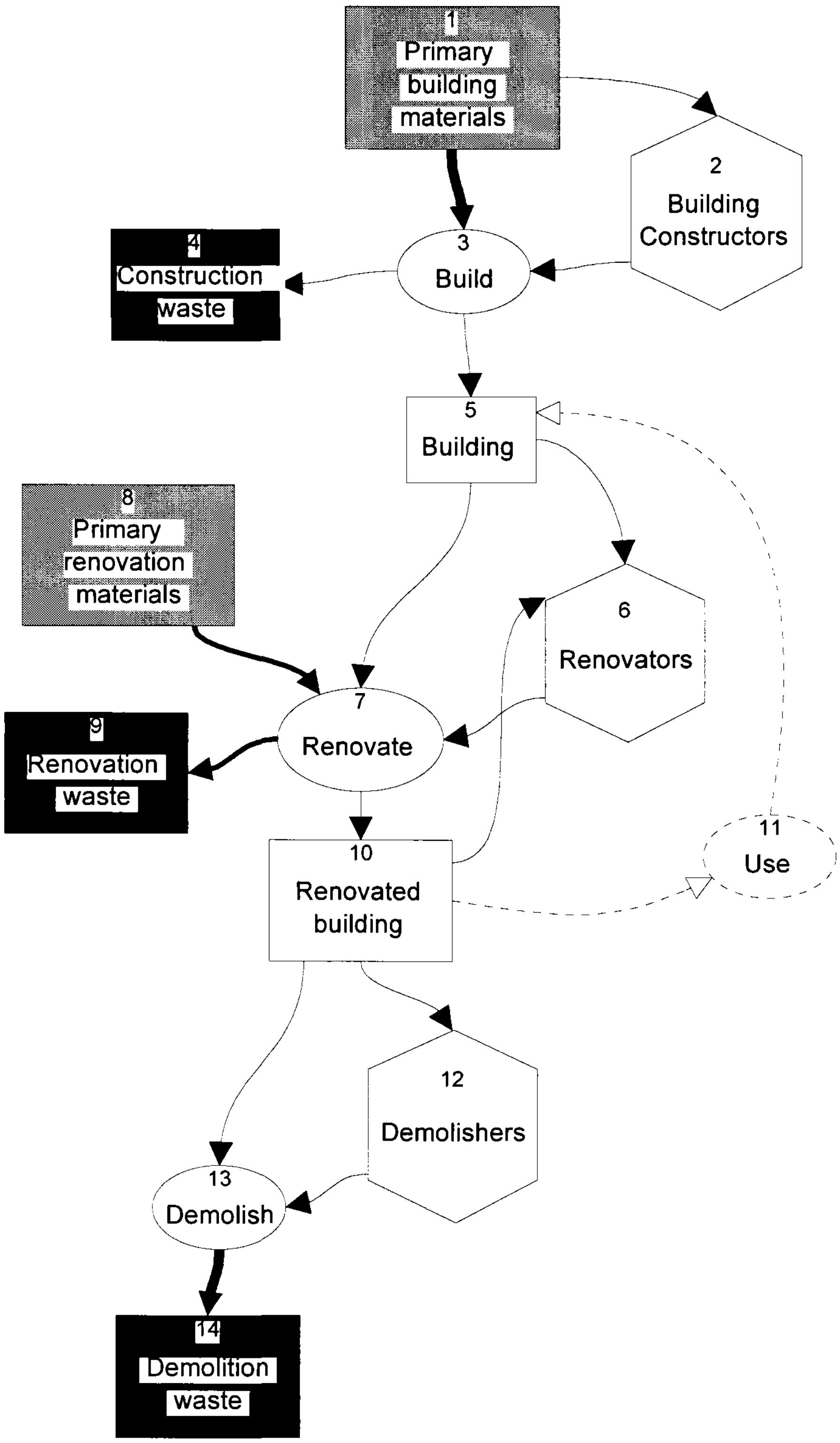


Figure 2: An essential "as is" model of the building and demolition sector.

9.3 "TO BE" MODELS OF THE BUILDING AND DEMOLITION SECTOR

"More coherence, knowledge and technology;
less chaos, matter and energy"

In this section, two "to be" models (an essential model and a more detailed model) of the building and demolition sector in the year 2010 will be presented.
At this point, some sidenotes with respect to the predictive value of strategic "to be" models are in place. Environmental policies, like any policy, are directed at intentionally changing parts of society. Society, however, is complex, and hardly a system that can be "engineered". Therefore, strategic "to be" models should be interpreted as both plausible and desirable futures, rather than as straightforward predictions. An environmental policy is directed at making small corrections towards such a plausible and desirable future: steps are small, and thorough evaluation of intermediate results as well as recurrent adjustment of the policy as a whole are well advised.

Notwithstanding the above, it is a major assumption of the *Trinity* methodology (as well as the central hypothesis of this dissertation as a whole) that, even within these boundaries, thinking about plausible and desirable futures, as well as thinking about (policy) actions that may lead towards these futures, strongly benefits from model-based support. The problem-solving process is both structured and supported by *Trinity* modelling processes. Only in cases that it is not possible to construct a perspective at all, is *Trinity* modelling of no help. In these cases, however, justifiable intentional actions are not possible: the policy would be *laisser faire* or an arbitrary intervention.

An essential model of the "to be" situation in 2010 is presented in figure 3. Figure 3, in a nutshell, pictures the concept of an *environmental chain closure* in *Trinity* notation: both the use of primary resources and the generation of building, renovation and demolition waste are minimised by means of introducing loops. This is an important strategy in attaining sustainable development. Waste materials are processed (not represented in this essential model), resulting in secondary building and renovation materials, which are used again in building and renovation processes at a later time. We call the model essential because the model focuses on the intentional activities of the main players, while neglecting other players that may be important at a more detailed level (the same three central actors that play a role in the "as is" model are present; their activities, however, are different).

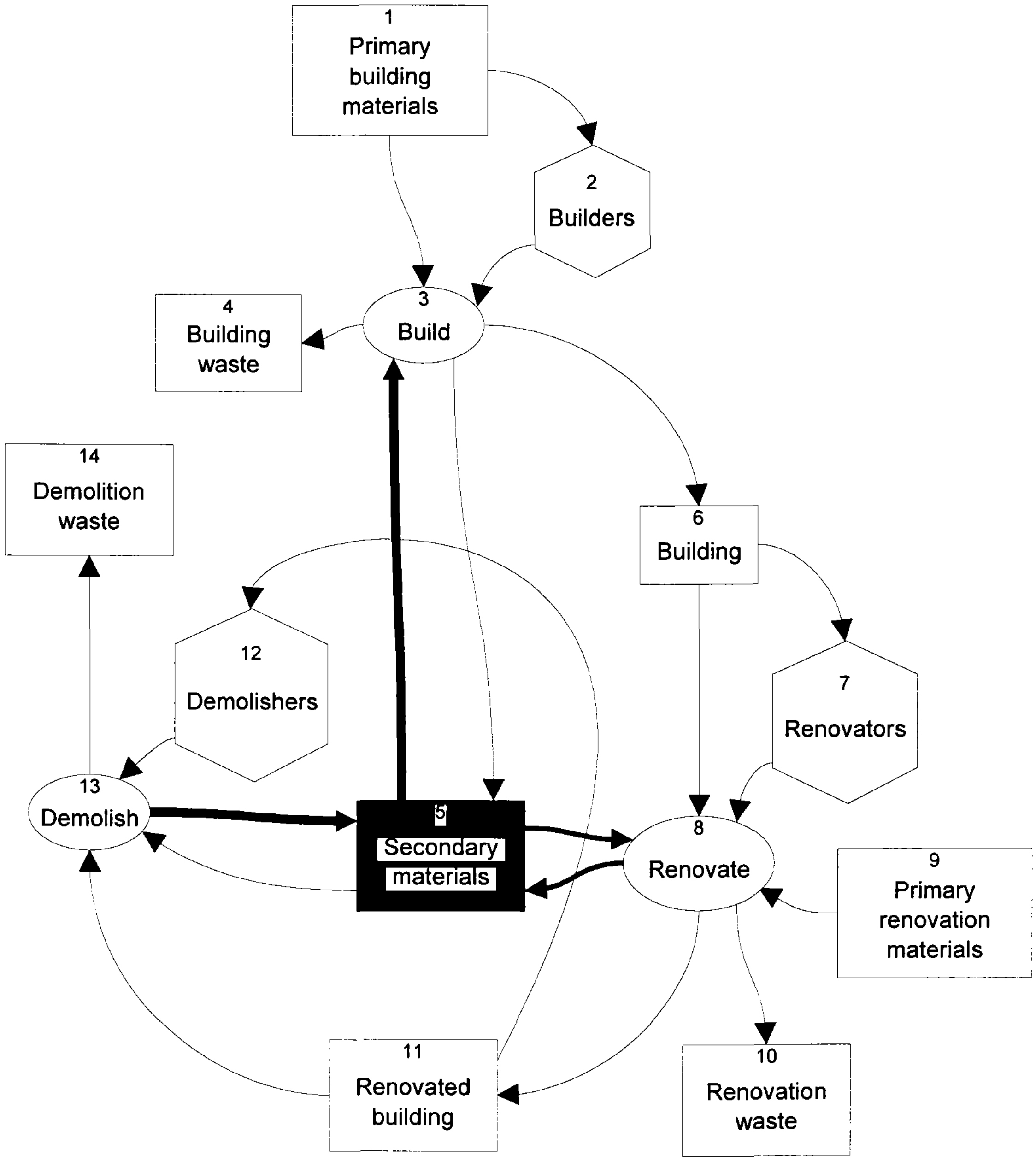


Figure 3: An essential "to be" model (for the year 2010) of the building and demolition sector: environmental chain closure.

Figures 2 and 3 are essential models: figure 2 describes the less desired "as is" situation and figure 3 prescribes a more desired "to be" situation. In combination, they emphasise the central thought behind improvement (i.e. transforming a sequential process into a cyclic process, in which matter is recycled into the same processes). Although essential, the "to be" model is far too abstract to present a clear image of the actually desired "to be" situation. It is deceptively simple. In order to obtain a more realistic model, more detail is required.

Figure 4 presents such a more detailed model. In terms of modelling strategies (Chapters 4 and 5), this model results from applying an expansion strategy (a parallel specification strategy) to figure 3. Although the same three central actors can be recognised as in the essential "to be" model, many more have been added. A salient feature of figure 4 is the introduction of several knowledge workers: in the same way that at this moment a builder is supplied with instructions by an architect, it is likely that in the future a renovator is informed by a renovation planner, and a demolisher by a demolition planner. In addition, the part of the chain that transforms the waste into secondary materials is made explicit, which illuminates logistic and waste processing activities.

In order for a *Trinity* "to be" model to be a feasible prediction of a realistic future, a minimal requirement is that *all* of its parts (both intentional and autonomous activities) should function. If one of the parts does not function, the whole network stagnates. This illuminates the main reason why intentionally changing networks of actors proves to be so difficult: many different actors should possess the *knowledge* to change their activities, should be in an *environment* that enables them to do so, and should be *willing* to do so, and this as a concerted action. Establishing such a coherence is not a trivial matter. Modelling processes, resulting in models like figure 4, support this process to a large extent. Some examples of this support are presented below.

Above we saw the introduction of several knowledge workers (notably the renovation planner and the demolition planner). Cognitive planning tasks and implementation tasks are separated to a large degree. The normal separation of architect and construction builder functioned as a template for these separations. As a matter of fact, looking at an intermediate *Trinity* model, in which the roles of architect and (construction) builder were separated, but the symbols of renovator and demolisher encompassed both the planning and the implementing role, resulted in the recognition that in the "to be" situation a separation is here likely too. In modelling terms, these intentional activities were submitted to a parallel specification (an example of such a procedure has been presented in Chapter 5: the specification of the fence renovator in an abraser and a painter). This is likely, as these roles become more knowledge-intensive, and in these situations a separation of planning tasks (performed by knowledge workers) and implementing tasks (performed by labourers) is common practice.

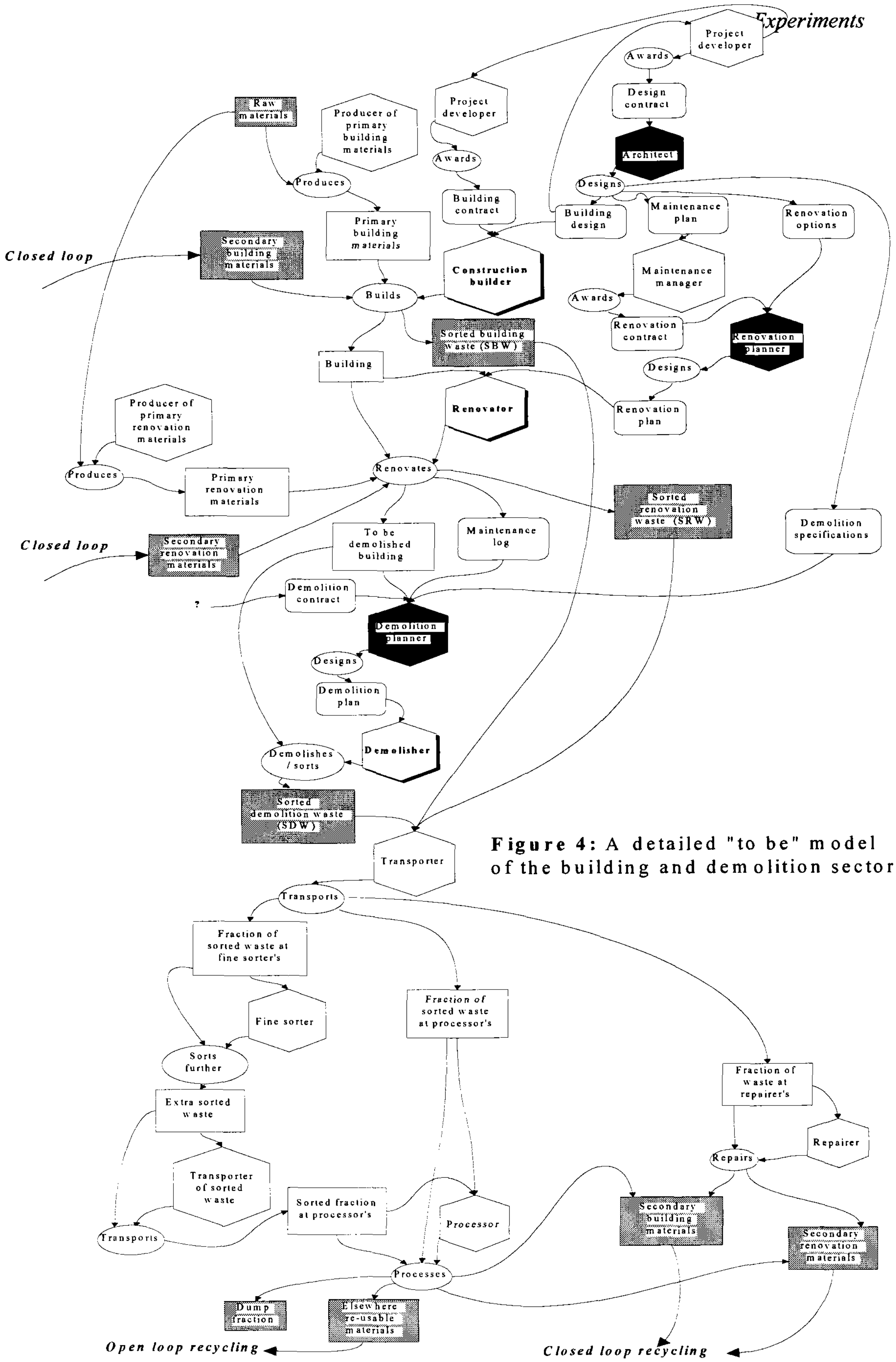


Figure 4: A detailed "to be" model of the building and demolition sector

Thinking further about the relations between building, renovating and demolition, the thought emerged that the cognitive parts of these three activities, performed by knowledge workers, will be (should be) tightly coupled in the future (this had already been suggested by figure 3, the essential "to be" model). An architect should be aware of the secondary building materials that are available, and should consider cyclic renovation and demolition processes in the future. A renovation planner should understand the cyclic design of the architect as far as renovation is concerned, should renovate in such a way that sustainable demolition is possible, and, on top of this, should take care that renovation waste is generated in such a way that it can be recycled or re-used, and that secondary renovation materials are used as much as possible. Finally, a demolition planner should understand the design of the architect as far as demolition is concerned, should be aware of the renovation history of the building, and should generate demolition waste in such a way that it can be recycled and/or re-used.

The picture that emerges is that architects, renovation planners and demolition planners are three roles that build upon one discipline, rather than three separate disciplines. In addition, an intensive communication infrastructure should be foreseen between these three roles. Figure 4 makes parts of this communication explicit by means of a series of rounded boxes: a building design, including renovation options, a demolition design, a renovation log, et cetera. We did not model this knowledge and communication infrastructure extensively, as our goal is not to dream up a grand design of the building and demolition sector in 2010 as researchers in isolation. Nonetheless, these issues should be part of a strategic debate about this sector, and *Trinity* models will play a supportive role in this.

Another issue, highlighted by figure 4, is the fact that both planners and craftsmen are merely hired to perform a job. They are not the principals, the decision makers. An architect, a renovation planner, a demolition planner only starts his activities after he has received an order. The decisive power, therefore, lies with quite a different type of actors: the project developers, the private or institutional owners of buildings, et cetera. These actors can and should be integrated in the model, as they play a crucial initiating role in the "to be" situation[80]. To be more specific: they play an important role in the transition process, and therefore should be modelled as a part of the script[81]. Being important players, they should participate in the strategic conference (which they did *not*: they were not invited). This points out another important use of *Trinity* models: next to setting up the agenda for meetings, they are supportive in determining (additional) crucial players that should participate in the debate.

[80] In modelling-strategic terms, the model of figure 3 is extended by means of parallel extensions.
[81] This once more emphasises the fact that it is very difficult, and even undesirable, to develop parts of a model of a perspective (in this case the "to be" part) in isolation from the other constituting parts (in this case, notably, the script).

In a nutshell, the general picture of figure 4 is that the network "to be" becomes more knowledge-, communication- and technology-intensive. In addition, considering the "to be" situation almost naturally shifts the focus on the script model. In the next section some explorations of this script will be presented.

9.4 THINKING ABOUT THE TRANSITION: FEASIBLE SCRIPTS

Above we described the way in which *Trinity* models support (in obtaining an) understanding (of) complex parts of society. The models presented so far in this chapter, however, described the situation "as is" and "to be". In this section, we will focus on the transition process, modelled by a *script*. In this section, we will not present a model of a script (which would be also a very useful activity), but we will reconsider some of the actions, agreed upon during the conference, making use of the additional understanding provided by the *Trinity* model of the "to be" situation (figure 4).

A script is a coherent composition of intentional and autonomous activities. Implementation of the script is expected to transform the situation "as is" into the situation "to be". As mentioned before, in case of complex parts of society, a script is not a recipe with guaranteed success. Nonetheless, modelling scripts supports perspective construction. In the strategic conference, the construction of what is called "scripts" in *Trinity* manifested itself in a very direct way: before the beginning of the conference an "input document" was prepared, containing the description of several potential actions and proposed actors. During the strategic conference, these proposed actions were discussed, and a selection was made. The "output document" of the strategic conference, therefore, was a reduced "input document". In addition, during the conference several actors declared that they would carry out specific actions of the output list.

In this section, we will review some of the actions that appeared on the output document of the strategic conference. In reviewing some of the actions agreed upon at the conference, four questions will function as guidelines. They are:

1. Are the actions adequate? (Do they lead towards the/a desired "to be" situation?)
2. Are the actions in combination a coherent "whole"?
3. Are important actions missing?
4. Have important actions been omitted from the input document?

We will not answer these questions exhaustively, as emphasis here is on illustrating the use of *Trinity* rather than on reviewing the strategic conference. Rather, by means of examples we will try to substantiate the claim that using the "as is" and the "to be" models presented before results in better scripts (i.e. scripts that do not suffer from flaws that were present in the output document of the conference). We want to stress that the fact that we did not actually implement our proposals is not a disclaimer with respect to the usefulness of *Trinity*. We talk about consequences that straightforwardly follow from the desired "to be" situation. It may indeed be the case that our proposals for improving the script are not realisable in practice, for example, because of the fact that participants of the conference would not have agreed to them. However, this would have to result in a reconsideration of the desired "to be" model[82], rather than in a denunciation of the *Trinity* methodology. In case that a "to be" model cannot be attained, this simply means that no script can be constructed, and, therefore, that a perspective is still lacking. Problem solving should continue until a perspective is obtained that models both the problem owner's intention and the problem owner's environment (see also Chapter 3). As mentioned in Chapter 3, the process of realising such a bi-directional model relation includes the possibility that the intention changes. Reconsidering the desired "to be" model because of pragmatic reasons is an example of this.

1. Are the actions adequate?

As a general rule, the answer to this question is yes: all the actions that were agreed upon during the conference were adequate in the sense that they started processes, assumed to lead towards the predicted "to be" situation. However, the *Trinity* models (figures 2-4) enable one to support some of the actions. For example, one of the actions is called "financial instruments":

> *"The government, together with actors to whom it concerns, will investigate whether and which financial instruments may be applied in order to stimulate sustainable building methods and re-allocation of existing buildings."*

This action is rather open-ended, which is not unusual for strategic conferences of this kind. Financial instruments are, for example, taxes (both higher and lower) and financial return systems.

Starting with the *Trinity* "to be" model, it is possible to investigate a) *for each actor* what financial instruments might be applied, b) the consequences for *neighbouring actors* in the network, and c) the effect on the "to be" network as a whole. In this way, the investigation as a whole becomes more thorough, and chain effects are included in a rather natural way. This investigation is likely to benefit significantly from model-based support, as in principle it is a D-type script synthesis. As a side-effect of supporting this investigation by means of *Trinity*, alterations to the "to be" model are likely to result as well (for example, new actors might emerge).

2. Are the actions in combination a coherent "whole"?

Do the actions in combination constitute a coherent system of self-supporting and self-enforcing parts? The answer to this question is that this is not the case since the overall

[82] Our advice would be to use *Trinity* to adjust this "to be" situation, or to model different alternative "to be's".

picture is rather fragmentary. For example, four different actions on the output list of the strategic conference are:

a) *Increase the recognition and the traceability of building parts and building materials, for example, by means of a coding system.*

b) *Improve the availability of product information with respect to building elements and building materials, in order to support environmental design choices in the design stage of a building (e.g. use specific secondary materials).*

c) *Develop a handbook to support the decision-making processes concerning re-using building parts and building materials.*

d) *Investigate the re-applicability of building parts and building materials.*

All these actions contribute to a transformation process in the direction of the desired "to be" situation. In principle, they should be tightly coupled. For example, action a) should inform action c) and d); actions c) and d) are rather inefficient if executed without co-ordination. Nonetheless, the designated executors of these actions are quite different parties, the actions take place parallel and a co-ordination structure or in-between information exchange is not foreseen. Double work, badly co-ordinated information supply and confusion with respect to the overall goal of these actions in combination are to be expected.

Again, what would help here is (the construction of) a script model that makes explicit *who* develops *which* aids and/or knowledge, and a "to be" model that clarifies *who* is the intended *user* of these aids or this knowledge (likely users are architects, renovation planners, demolition planners, sorters and processors, see figure 4). Such models are of great help in obtaining a better overview of the specific knowledge that different participants in the "to be" situation really need. They would help in the preparation of a far less fragmentary (see above) set of actions. We did not construct such a model, as, in our opinion, it should be based on a knowledge acquisition process intensively involving the actors to whom it concerns, rather than purely the ideas of a research team.

3. Are important actions missing?

We will make two suggestions with respect to crucial actions that follow from the *Trinity* models, and that were missing on the output document of the conference.

The first suggestion is derived from the observation that the "to be" models clearly express that designing a building, planning a renovation and planning a demolition should be three aspects of one and the same unifying discipline, rather than three separate disciplines. Such a linking is an essential prerequisite to be able to attain a sustainable building and demolition sector. An action directed at investigating this idea and exploring the contours of the unifying discipline, however, was missing.

The second suggestion is derived from the fact that figure 4 clearly shows that the three pairs of planning and implementing disciplines (architect/builder, renovation planner/renovator, demolition planner/demolisher) are merely contractors: the decisive power lies elsewhere. Therefore, it would have been wise to invite actors such as project developers, private or institutional owners of buildings, et cetera, to the conference as well. However, they were not invited: for this reason, actions in which they participate are missing.

It may be difficult to develop perspectives and to start actions in which different actors, possessing different possibilities, potentials and intentions, participate in a concerted action. If the models show that such actions are required in order to attain a better future, it should be tried despite these difficulties. If they turn out to be not feasible, the perspective being modelled simply is not pragmatically correct, and therefore not an action potential (the "to be" situation cannot be realised by means of the script: it is likely a utopia). This is the essence of a back-casting approach: models support the process of assessing which directions are worth striving for. If the route is realistic as well, a way is found to realise improvement.

4. Have important actions been omitted from the input-document?

The last question is whether important actions of the input document were omitted that should have been executed according to the models. The input document stated an action that concerned the observation that the profession of demolition is a low-skilled profession that should be upgraded. This observation is indeed reflected by the specified "to be" model (figure 4): the demolition planner takes care of the knowledge-intensive part of demolition activities. Specifically, the action on the input document proposed to appoint a professor in what might be called "demolition technology". However, this action did not survive.

It might be questioned whether appointing such a professor would address the underlying problem: the fact that architects, renovation planners and demolition planners are three roles that build upon one underlying discipline. A better action might have been to appoint a professor for "sustainable building, renovation and demolition".

Whatever the exact discipline of the professor would have been, with hindsight it is hardly surprising that the professor did not survive. Representatives of technical universities (who are in the best position to understand, appreciate and realise such an action) were not present at the conference. A lesson to be learned from this is that the list of participants of a strategic conference should be composed on the basis of one or several (more or less coherent) ideas about the situation "as is", a desired "to be" situation, and routes that may lead towards these better futures. In short, the list should be based on initial perspectives. We would like to add: the list should perhaps be based on *Trinity* models of these initial perspectives. These models are not indisputable: they should rather function as starting points (hence *initial* in *initial perspectives*) of the debate.

9.5 DISCUSSION AND CONCLUSIONS

Several topics that had already been addressed in this experiment require further discussion. They are: the *Trinity* modelling process applied in this experiment; the suitability of strategic conferences for D-type perspective construction; and the intentional nature of *Trinity* modelling.

Reviewing the Trinity modelling process applied in this experiment

In this experiment, we applied a *back-casting* approach. A *back-casting* approach from a *Trinity* point of view is a specific pattern in time with respect to the development of the three parts of a perspective of concern. *Trinity* does not prescribe or specify a specific pattern in time. Troubleshooting, back-casting, exploiting a specific trick (an approach in which a global version of the script is available first; this approach is often used by specialists), or any other (possibly complex) pattern can be supported by the *Trinity* methodology. Or rather, these patterns are empirical generalisations (they are distinguished bottom-up): a back-casting approach in *Trinity* terminology simply is a sequence of modelling steps in which the emphasis shifts from "to be" to a script. This is similar to the way in which (*Trinity*) modelling strategies are distinguished (Chapters 4 and 5). The difference between *Trinity modelling strategies* and *Trinity modelling approaches* is that *Trinity* modelling strategies are distinguished on the basis of differences during the modelling process in attributes like genericity, scope, level of detail, number of viewpoints or overall complexity of the model relation, whereas *Trinity* modelling approaches are distinguished on the basis of which part of a perspective ("as is"; script, "to be") is given attention first[83]. (In order to be complete: a third dimension that can be used to distinguish specific use of *Trinity* is the specific way of communication. For an overview, see section 5.7).

A back-casting approach is especially suitable in situations in which several, rather diverse, potential futures have to be explored. During the strategic conference this was not the case: rather, only *one* description of a desirable future was presented. Although this is perhaps not optimal, even in this "one-model" case adopting a back-casting approach is a good choice, because this approach furthers a rather explorative thinking process. Nonetheless, with hindsight we feel that explicitly describing different scenarios (see also section 5.6) and debating all of them would have been a better choice. The reason for this is that, in light of the strategic nature of the conference, it would be rather coincidental that exactly this future (although constructed on the basis of interviews with field players) would be "most desired" as well as "best realisable".

The *Trinity* modelling strategy that we used was predominantly a parallel building blocks strategy, although in the prediction stage several parallel specifications were applied as well (notably the specification of both the renovation and the demolition activity of the essential "to be" model into a planning and an implementing activity). Although firm proof is lacking at this moment, we believe to have noticed that in analytical modelling

[83] This issue was addressed from a methodical point of view in section 5.7.2.

stages (i.e. describing "as is"), typically, strong building blocks flavours can be noticed, whereas in predictive modelling stages transformation strategies play an important role as well. The reason for this perhaps is that in describing "as is" the referent is a rather confronting entity, whereas in predicting "to be" the referent is created mentally: more degrees of freedom are present. In analytic stages of the modelling process, attention tends to shift from identifying (physical) actors to their exact role and/or function (from "who is present" towards "what do they intend" or "what are they doing"), whereas in predictive stages of the modelling process attention tends to shift from thinking about roles and functions towards identifying or appointing likely or suitable actors (from what is needed towards who may do this). In script construction, it is difficult to make such a distinction: sometimes the presence of actors results in giving them a role in the script, and sometimes the presence of a role in a script results in appointing an actor. Table 1 summarises these tendencies.

Table 1: Shifts in attention between actors, on the one hand, and roles/functions, on the other.

STAGE IN PROBLEM SOLVING	SHIFT IN ATTENTION
Description of "as is" (analysis)	From actors to their roles/functions
Prediction of "to be"	From roles/functions to actors that may implement them
Script construction (prescription)	From actors to roles/functions as well as from roles/functions to actors

As this experiment was predominantly an *ex post* study, the mode of *Trinity* we predominantly used was *isolated use*. This is not the most preferable manner of using *Trinity* in strategic conferences. However, we used this mode because of pragmatic considerations (see also the introductory section of this chapter). In principle, *all* the research stages concerning a strategic conference (i.e. preliminary investigations; identification, selection and interviewing of actors; construction of *several* preliminary perspectives, identification of the conference topics, organisation of the conference, integrating the conference results, et cetera) should be supported by *Trinity in line*, preferably in a mixture of *hidden* and *participative* modes. This experiment showed that this is likely to improve processes involving strategic conferences (as well as strategic explorations in general) to a considerable extent.

The suitability of strategic conferences for D-type perspective construction

It is difficult to operate and intervene in D-type contexts. The experiment described above shows that an important reason for this is the absence of coherent perspectives: images of "as is", "to be" as well as coherent scripts that may realise the transition. The material present in the conference documents was rather fragmentary and very implicit, at least from a *Trinity* point of view. In addition, only one future was predicted. At several moments during our experiment we felt a strong urge to start developing script models as well, but for this the information was lacking. A lesson we learned is that, although shifting emphasis from one of the three parts of a perspective towards the other(s) has some important consequences for the nature of the problem-solving process (for example; an early emphasis on "to be" furthers explorative thinking), it is not wise to develop parts of perspectives in separation.

A general advice for organising strategic conferences is that more attention should be paid to obtaining, exchanging and discussing several coherent perspectives. In this respect it is important, though, to take into consideration very explicitly the fact that many participants must be characterised as "non-professionals" in D-type problem solving. Therefore, a well-balanced mixture of hidden mode and participative mode should be used, as burdening occasional participants in the problem-solving process with *Trinity* conventions is to be avoided (see also Chapter 10, the discussion of the experiments in combination, where several guidelines in designing problem solving processes are presented).

It may even be questioned whether strategic conferences in isolation are a suitable means to attain the goals of the conference, as presented at the beginning of this chapter. Strategic conferences should rather be embedded in a larger process. It is the goal of this process as a whole to result in coherent and feasible, pragmatically correct perspectives. Strategic conferences function as a means to initiate, explore, adjust, verify and prioritise perspectives that are elaborated, refined and worked out in background processes. Also during these background processes, intensive communication with "field players" is to be advised. The experiment presented above shows that *Trinity* offers important and valuable support for developing both clear and assessable parts of perspectives. There are no reasons to assume that this is only the case for this specific strategic conference, directed at the building and demolition sector: the foundations and assumptions of *Trinity* as a methodology are, to a large degree, independent of disciplines, sectors and specific intentions.

The intentional nature of Trinity

A *Trinity* model of a D-type perspective does resemble a blueprint of a transformation process of a part of society. It may be argued that society is far too complex for such an approach: according to this point of view, society is inherently unpredictable. On the other hand, far reaching interventions in (parts of) society are common practice in politics, business life and other areas of human endeavour. We agree that society cannot be "engineered" (in a technical sense). In addition, we acknowledge the fact that perspectives of parts of society are working hypotheses, and that especially the "to be" parts of societal perspectives are more or less plausible outcomes, rather than straightforward and logical consequences of executing the script. We also acknowledge that each step of a script implementation process should be monitored and evaluated thoroughly: the results of this may change the overall perspective to a considerable degree. Developing D-type perspectives may not be an easy task. But this certainly is not a reason to lower standards. Purposeful intervention should be motivated and guided by perspectives, even (and especially) in D-type situations. *Trinity* offers support in this respect.

In summary, the experiment described in this chapter makes clear that *Trinity* models and modelling support strategic conferences in several ways:

- in selecting/identifying potential participants;
- in addressing responsible or engaged actors;
- in understanding the problem context (as is, action, to be) as a coherent whole;
- in distinguishing strategic issues in the transformation process (i.e. the integration of different planning tasks);
- in composing the list of important topics for a conference;
- in thinking through the consequences and domino effects of candidate actions;
- in developing coherent visions and packages of actions (coherent scripts).

As a result of these findings, we were asked to evaluate the outcome of two additional conferences as well (work in progress). Future *in line* use of *Trinity* is under consideration, which would provide a more severe test of *Trinity* in action in (the processes encompassing) strategic conferences, including direct communication with participants of the conference (something that, for reasons already explained in section 9.1, is still missing in this experiment).
The different ways of support, summarised above, we consider to be generic for strategic environmental technological conferences to a large degree. This allows us to answer the central question of this chapter (see section 9.1) in the affirmative on the basis of a generalisation procedure: the use of *Trinity* is highly likely to result in added value in future strategic conferences. Future *in line* use of *Trinity* in (the processes involving) strategic conferences is a sensible next step, that will test this more rigidly.

CHAPTER 10

DISCUSSION AND CONCLUSIONS OF THE EXPERIMENTS

10.1 INTRODUCTION

In this chapter, on the basis of the three experiments that we conducted, the use and added value of *Trinity* will be discussed, and conclusions will be drawn. A methodological discussion of the use and added value of *Trinity* will be presented in the general discussion (Chapter 11).

10.2 USE AND ADDED VALUE OF *TRINITY*

In this discussion three aspects will be emphasised. First, we will discuss the different ways in which the *Trinity* methodical framework (i.e. its *methods layer*) has been used during the experiments. Second, several comments will be made with respect to its use in practice. Third, the added value that manifested itself in the experiments will be addressed.

10.2.1 Use of *Trinity* methods: rules of thumb

The settings of all three experiments could be interpreted as D-type problem contexts. In addition, the time span of the activities gave room for reflective thinking. In principle, this should be sufficient to be able to use *Trinity*. And indeed, on the basis of these arguments we started using it.

The specific way of using *Trinity*, however, was quite different in each of the experiments. This made it very clear to us that, at the methods layer, *Trinity* is a very flexible methodical framework, rather than a single method. From this point of view, the experiments were quite successful: apparently, we developed a methodology that functioned under different circumstances in different D-type problem contexts!

The other side of the coin, however, is that this made us realise that, although we developed a flexible methodical framework, we failed to present sufficiently clear guidelines and rules of thumb that guarantee a proper use of this framework.

An excuse for this is perhaps that guidelines and rules of thumb (should) develop in practice. Therefore, at this point we will present several of the lessons learned. In line with the introduction to the three experiments (Chapter 6), we will discuss the following aspects: the mode of communication; the order in which perspectives develop; the modelling strategies that were used.

The mode of communication

In the three experiments we used several modes of communication (see section 5.7.1). In most cases the mode of communication was dominated by practical circumstances: as we were operating as contract researchers, several boundary conditions were already present. Nonetheless, with hindsight it is possible to make some general remarks as to which mode of communication is to be preferred in what type of situation.

In evaluating the results of the strategic conference *Building and demolition waste* we predominantly used *isolated use*. In this case *isolated use* at first sight seemed to be an acceptable mode, given the fact that we operated on the basis of reports and documents, and all the necessary information seemed to be available. On many occasions, however, we felt an urge to consult field players and informers as, from a *Trinity* point of view, the conference reports were rather fragmentary, incomplete and abstract. This was re-established during several evaluations of other strategic conferences since then [Diepenmaat, van Lierop and Roorda (1997)]. In combination with the fact that *isolated use* introduces the danger of a) developing rather shallow and short-sighted perspectives, and b) limited support from field players, we are not very fond of this mode of communication. Therefore, generally speaking:

> *Isolated use should be used sparingly, and only as an intermediate part of a larger overall problem-solving design.*

In the *VOC2000* experiment, *hidden use* was our mode of preference. The reasons for this preference were that a) participants could be addressed in a language that they master completely: natural language, b) participants were not burdened with syntactical, semantic and pragmatic conventions of the *Trinity* language (see also appendix B), and c) because of the intensity of *Trinity* modelling sessions[84] we felt that the best setting is to have relatively few participants (< 10) who are rather familiar with the methodology. In summary, and again speaking in general:

> *Hidden use should be considered in case of interaction with many societal actors.*

Finally, *participative use* took place in the *VOC2000* and the *Indoor environment* experiments. *Participative use* is our mode of preference within relatively small groups of persons, that are intensively committed to and engaged in finding solutions for the problems at hand. Typically, in our case such a group consisted of primary problem owners, contract researchers that were hired to come up with solutions, and other

[84] They require quite a lot of cognitive effort indeed!

consultants/informers on an "if-needed" basis. The group continued to work together (with small variations) throughout the entire experiment. Therefore, generally speaking:

> *Participative use should be considered in cases where small teams work together intensively.*

These are first results only. However, two interesting things can be gathered.
The first is that, in really complex problems, involving many interactions with many different actors, *hidden use* may be used with respect to these actors, and *participative use* may be used as a background mode within the (typically smaller) research teams. In combination, this results in a bi-modal approach.
The second is that *participative use* requires a basic familiarity with *Trinity*. In our experience, *reading Trinity* models can be learned well within half an hour, but actually and actively *participating in* a modelling process, even guided by professional *Trinity* users, requires a more thorough understanding, that can be acquired within several hours. Becoming a *professional Trinity user* requires a thorough understanding of its philosophy, theories and methods, and above all a lot of *Trinity* modelling experience. In this sense, it is a practice rather than a science.

The order in which perspectives develop

In the three experiments, different shifts in emphasis with respect to the three parts of a perspective could be distinguished (see also section 5.7.2).
In the *Indoor environmental problems* experiment, we first emphasised the "as is" part. The reason for this is that, typically, indoor environmental problems are rather confronting; therefore, a first requirement is to obtain a thorough understanding of the situation as it manifests itself. Only after this first thorough analysis may the emphasis shift towards scripts and "to be" models. In terms of a rule of thumb:

> *In case of a rather confronting problem, first emphasise analysing the situation "as is" (adopt a trouble-shooting approach).*

In the *VOC2000* experiment, in a sense, two different levels of perspectives were at stake. The first level was the intentional transition towards innovated processes and products. However, this transition was stagnating. For this reason, this transition process as a whole was turned into the "as is" situation of an intentional activity, directed at improving the process. Note that this implies a *shift in problem context*. From the point of view of the problem owner of the second problem (see also figure 1 of Chapter 8), obtaining a clear understanding of the stagnating transition process perspective is an analysis, and therefore he is following a *trouble-shooting approach* (the rule of thumb mentioned above applies).

One of the lessons that we have learned especially from the *VOC2000* experiment is that, in multi-actor situations, it is *of eminent importance* to be *as specific as possible* about *who* is the problem owner of the perspective being developed[85], as this largely determines who and what is part of the problem context and who and what is not. Perhaps the most confusing aspect of D-type problem solving is that every actor has his own intentions, perspectives and environments and, therefore, a myriad of problem owners and problem contexts can be distinguished, that only seldomly constitute a coherent whole. The shift in problem context, discussed above, makes clear that these problem owners and problem contexts may even manifest themselves at different systemic levels. *This multiplicity cannot be avoided, but should be dealt with adequately. Trinity* offers support in this respect, as intentional activities (hence combinations of problem owner, perspective and problem context) are its focal point.

Finally, in the *Building and demolition waste* experiment it was important to develop an image of a far-away future. In cases like this, it is important to operate in a rather explorative fashion: (perhaps after an initial, global analysis of the "as is" situation) emphasis should be on *what could be*, rather than on what might be realised starting from the "as is" situation. Strategic and explorative thinking is highly facilitated by emphasising "to be" models early in the problem-solving process. This will prevent too shallow an exploration of the space of action potentials. It is only after that (by means of a backward reasoning process) that feasible scripts and more specific situations "as is" should be modelled. Unrealistic "to be" models will be exposed during this process, as it will turn out to be impossible to connect them to feasible scripts. Summarising this discussion in terms of a rule of thumb:

> *In case of strategic questions, emphasise the prediction of several situations "to be" early in the process (follow a back-casting approach).*

In Chapter 2, we stressed that many different routes may be followed, even when restricting the argument to the order in which parts of a perspective develop. It should be remembered, though, that at that point we also stated that it is *impossible* to emphasise one of the three parts, and ignore the other two parts altogether (remember the bucket analogy). For example, although emphasising "to be" models at the beginning of a strategic problem-solving process facilitates a rather explorative thinking, these very models are implicitly proposed as alternatives for an undesired (although perhaps only vaguely understood) "as is" situation. Distinction of approaches like *trouble-shooting* and *back-casting* merely points to relative differences in emphasis during the problem-solving process, and not to absolute ones.

It is not recommended to develop parts of perspectives (for example, the " as is" or the "to be" part) completely in isolation, i.e. without considering the other constituting parts. Interpreting perspectives as wholes typically results in re-interpretations of their parts. With hindsight, for example, the *Building and demolition waste* experiment was perhaps

[85] Problem ownership is allowed to change during the process, though. The point we want to make here is that, at *any* time during the problem-solving process, the relation between perspectives and problem owners should be as clear as possible.

focused too much on the "to be" model. Undoubtedly, modelling the corresponding two parts ("as is" and script) will induce modifications of this "to be" model. The lesson that we have learned is that, although it may be very effective to vary the order in which different parts of a perspective are emphasised, the general rule is that the complete picture should never be neglected. Pragmatically correct models of perspectives, although consisting of three parts, are meaningful by virtue of the fact that they refer to action potentials, and action potentials cannot be reduced to three parts in isolation.

Modelling strategies that were used

During all the experiments we used a mixture of many different modelling steps (see sections 4.3 and 5.4.4). However, when restricting ourselves to the main lines, we found ourselves using different modelling strategies (see section 4.4.2 and section 5.5).

In the *Indoor environment* experiment we operated rather top-down (a parallel specification strategy). The modelling process started with the generic "as is" model *"the minimal environmental situation of concern"*. Subsequently, a top-down transformation strategy (a specification strategy) was used in order to refine this model. Subsequently, a multi-referent restriction strategy was used in order to obtain generic models that address specific indoor environmental problem situations.

The *VOC2000* experiment was characterised by parallel extension strategies. The model was extended (and sometimes restricted) making use of information about actors and roles that were derived from interviews with field players.

Finally, the *Building and demolition waste* experiment was characterised by a parallel building blocks strategy.

When looking back at these strategies and their relation with the experiments we can derive some general guidelines.

The *Indoor environmental problems* experiment was strongly influenced by the background idea that models should be used *by other persons to support diagnostic processes* (i.e. knowledge transfer was at stake). They were to be used as aids in classification processes (see also the proposed knowledge infrastructure presented in section 7.5). In such circumstances, the use of a combination of parallel specification and multi-referent strategies is a good choice, as this enables one to build multi-level hierarchies of generic models that allow for a stepwise classification of a specific problem situation. The indoor environmental models provide a good example of such a hierarchy. (See also figure 10 in section 7.6.)
In the *VOC2000* and the *Building and demolition waste* experiments the level of abstraction was determined by the sources that we used to obtain model elements (predominantly interview reports and conference documents, respectively). This situation is likely to result in some sort of a building blocks strategy, as the granularity of the

building blocks is largely "provided", rather than "generated" by the modeller. Especially in the *Building and demolition waste* experiment the use of a parallel building blocks strategy was dominant. This was induced by the fact that we only used existing documents about the strategic conference, in which these building blocks were written down. With hindsight, we were not very pleased with this, as several times during the *ex post* analysis we had questions with respect to the level of abstraction present in these documents. To be more specific: this level was considered too high, the documents were found to be too abstract to make the connection to actions (which nonetheless was an important goal of the conference).

From the discussion above it follows that a complex relationship exists between characteristics of D-type intentional activities and the specific (combination of) *Trinity* methods to be preferred. At this moment we are not yet able to describe this relationship completely. This will be an important research issue for the (near) future.

10.2.2 Use in practice

From a general point of view, several practical aspects are worth a short discussion.

The first discussion point is concerned with the specific attitudes that different D-type problem owners expressed when being confronted with the *Trinity* methodology for the first time.

Typically, the first reaction was one of appreciation. D-type processes are known to be difficult, and any attempt to provide methodological support is considered to be very important.

In many cases a second reaction was: "But that's what I am doing already!". We consider this to be a major compliment for *Trinity* (too large a deviation from common practice should raise suspicion, rather than applause). After having participated in the experiments, these persons, however, had to admit that using *Trinity* structured, supported and influenced their dealing with multi-actor problem contexts (and in most cases even their interpretation of the very problem itself) in important ways (see also the separate discussions of the three experiments, and section 5.6). In practice, it proves to be very difficult to deal with the intrinsic complexity of D-type problem contexts without making use of any supporting methods (see also table 4 in section 5.6). *Trinity* in a sense "forces" users to *explicitly* address the peculiarities of D-type transition processes, and, therefore, results in both a more explicit and a more thorough understanding of the problem context of concern.

Finally, some persons did not like (the ideas behind) *Trinity* at all. They considered *Trinity* to be far too "theoretical" to be of any potential use in "practical situations". Consequently, they were not willing to participate in any experiment whatsoever. They did admit, however, that dealing with D-type problem contexts in "practical situations" is difficult indeed. We want to end this discussion point therefore with two maxims and a ground rule of *Trinity*, that may give these persons some food for thoughts:

1. nothing is as practical as a good theory;
2. methods should be as simple as possible, *but not any simpler*; and
3. when claiming to act intentionally, one should possess and be able to explain one's perspective, even and especially in D-type situations.

A second issue is that we, as experienced *Trinity* users, were surprised again and again about the difficulty of developing pragmatically correct models of D-type perspectives, *even in cases where we believed to thoroughly understand the problem context of concern.* A big difference exists between the intuitive understanding of a D-type situation and the ability to translate this understanding in a pragmatically correct *Trinity* model (see also table 4 in section 5.6.2). Bearing in mind that we consider the availability of pragmatically correct perspectives to be a *crucial* prerequisite for successful intentional action, it is not surprising that intentional improvement of D-type situations proves to be so difficult. For this reason, we consider the (further) development of methods to support D-type problem solving an important route to increase our success rate in dealing with them. Adequately solving D-type problems is of great importance to modern society. For this very reason, D-type problem solving should be turned into a profession, rather than remain an art.

10.2.3 Added value

A point of great practical value proved to be the process of establishing problem ownership. This was an interesting activity, as in two out of the three experiments the actor causing the problem-solving process did mention from the start that *in principle* others should take action: he was concerned, but not the envisaged actor. (Others were the envisaged actors, but not concerned.) We have encountered this phenomenon several times since then. Actors who for some reason are willing to take part in, and even start up the problem-solving process (they claim to be involved) are not always willing to act, not even in principle[86]. We consider *Trinity*'s explicit emphasis on establishing problem ownership (rather than on "problem definition") to be a very strong point, because this enables (and in a sense forces) us to link participants to actions.

Another very practical aspect of *Trinity* is that, in spite of the sometimes overwhelming complexity of D-type problem contexts, for a D-type problem solver it is very clear what should be done. The task is to develop a D-type perspective that is pragmatically correct, and the means to do this are provided by the methods layer. In all three cases this helped us very much in starting the experiments, notwithstanding the fact that at that point we were not especially knowledgeable in the respective problem areas.
In line with the *Trinity* approach (see also section 5.6, practical guidelines)[87] in all three experiments we started with the initial identification, selection and characterisation of

[86] This generalisation may be coloured by the fact that we predominantly operate in environmental problem contexts.
[87] In section 5.6.1 we introduced the structured intention list and process list to help in this respect. Section 5.6.1, however, was written at a rather late stage of preparing this dissertation; for

actors and autonomous processes of interest. *Trinity* constantly focuses our attention on actors, action potentials and actions, and this at several systemic levels. This is indeed a distinguishing and supporting feature of *Trinity*. The requirement of pragmatic correctness constantly poses the same questions over and over, and this at all systemic levels: does this model refer to (model) an intentional activity? Is the perspective being modelled indeed an action potential? And does implementing it result in an improvement (also in comparison with alternative action potentials)? Attempts to answer these questions in the affirmative, and taking appropriate knowledge acquisition steps in case this is not possible, constantly drives the problem-solving process towards action potentials (pragmatically correct perspectives). If all the intended actors are willing to act according to these perspectives, a solid basis for concerted action is present. The proof of the pudding, however, is in the eating. And a thorough evaluation is advised.

In many cases of real-world D-type problem solving, a clear and coherent overview of the intentional actions of concern does not exist. Especially in D-type improvement processes it turns out to be difficult to obtain such an overview. And yet, this is a (if not the) most important requirement for being able to intervene intentionally, and turn the less desirable situation into a better one. In all three experiments, *Trinity* provided the "machinery" to develop such an overview: the resulting models led to a clear understanding of the network dimension of D-type problem contexts. To be more specific:

> In the *Indoor environmental problems* experiment, the "as is" models describe different D-type problem manifestations in a generic manner. In addition, the models point out many different routes to "branch off" the causal flux of events resulting in sub-clinical health effects. The models, therefore, support both diagnostic as well as remedial activities in this D-type problem context.
>
> In the *VOC2000* experiment, the problem areas of the list proved to be causally connected in complicated ways. Understanding this network dimension was shown to be important in future attempts to improve the situation. Our recommendations, which followed from the model-induced understanding of the problem context as a whole, have been accepted by the steering committee of the VOC2000 programme, and will be the basis for future actions directed at improvement.
>
> Finally, in the *Building and demolition waste* experiment, the "to be" model induced a reconsideration of actions that were agreed upon in an earlier stage, and highlighted several missing aspects that are indispensable in attempts to realise this future.

In summary, we consider this to be added value indeed.

---

this reason we did not actually use these aids in their full form during the experiments described here. These lists are used, however, intensively in [Diepenmaat, van Lierop, and Roorda (1997)].

10.3 CONCLUSIONS

The experiments presented in this experimental part of the dissertation, although different in important respects, do not provide a complete test of the *Trinity* methodology in action. This will require far more field work. In addition, we are not yet able to report the successful construction, use and evaluation of complete perspectives.
The experiments do show, however, that *Trinity* performs well in supporting D-type problem-solving processes. *Trinity* proved to be supportive in important ways in all three experiments. Multi-actor situations are understood better. The design and thinking through of (domino effects of) potential interventions in and improvements of multi-actor networks are supported by the model(s). The *Trinity* process, in addition, guides and supports the communication processes with informers, consultants and field players during the problem-solving process. Therefore, on the basis of the experiments, we conclude that *Trinity* offers model-based support for multi-actor problem solving.

PART VI: GENERAL DISCUSSION AND CONCLUSIONS

CHAPTER 11

GENERAL DISCUSSION

11.1 INTRODUCTION

In this thesis, we presented the *Trinity* methodology. This methodology is directed at offering model-based support for multi-actor problem solving. In line with the research questions (see section 1.2) we conducted several research activities. We developed a philosophical and theoretical framework, described in the Philosophical background and Theory parts of this dissertation. We designed modelling methods that specifically enable support of problem-solving processes in multi-actor situations. These are described in the Methods part of this dissertation. Finally, we conducted experiments, which shed light on the use and added value of the *Trinity* methodology, when applied in real-world multi-actor problem solving. This is described in the Experiments part of this dissertation.

In this chapter, several aspects of the *Trinity* methodology will be discussed. We will restrict ourselves to four main themes. First, we will review and discuss the basis of the *Trinity* methodology. Second, we will highlight the key features of *Trinity*, and discuss their relevance for model-based support for multi-actor problem solving. Third, we will reflect on the work presented in this dissertation as a whole. In doing so, we will discuss the contours of a background theory that underlies and ties together the different chapters of this dissertation. In addition, we will position *Trinity* in the field of mainstream paradigms for dealing with complexity. Fourth, the added value of using *Trinity* in real-world problem solving will be discussed.

11.2 THE BASIS OF THE *TRINITY* METHODOLOGY

Two fundamentally different approaches can be distinguished in developing a methodology (these approaches, however, only seldom manifest themselves in their pure form). The first approach is to develop theories from practice. On the basis of repeated experiences in the domain of application, theories gradually emerge. The second approach is to develop theories, and to start testing these theories in practice.

When looking back on the development of *Trinity*, it is difficult to say which one of these two approaches was dominant. In terms of the methodological pyramid (see the Introductory part of this dissertation), the first approach *grosso modo* develops from the *use* layer towards the *philosophy* layer. The second approach develops from the *philosophy* layer towards the *use* layer. In Chapter 1 we mentioned that, in the present work, we started with a premature version of the *Trinity* modelling language (i.e. the *methods* layer), which suggests a middle-out approach in terms of the pyramid.
However, rather than giving an accurate historical account of its *development*, we want to show that the *Trinity* methodology *itself* (i.e. the *result* of this development) is consistent with both a bottom-up (practice -> theory) as well as a top-down (theory -> practice) point of view. First, we will highlight the empirical basis of the *Trinity* methodology (which emphasises its bottom-up nature). Second, we will provide a description of the *Trinity* methodology on the basis of a *first-principles* approach (which emphasises its theoretical structure).

On the one hand, the *Trinity* methodology is built on empirical generalisations. We departed from two very widespread and even common sense notions: autonomous activities and intentional activities (see also [Diepenmaat, van Lierop and Bruijnes (1997)]). In addition, we used the notions of states and processes to flesh them out. Although using these notions implies a paradigmatic choice (they imply a discrete paradigm, see section 11.4.2), they hardly constitute a theory; they are considered to be common knowledge. Furthermore, we distinguished states and processes in only three domains: the physical domain, the communication domain and the knowledge domain. Again, although our specific interpretation of these domains may be called theoretical, the distinction of physical, communication and knowledge domain phenomena is hardly disputed[88] and easy to explain. We consider it a strong point that *Trinity* departs from a limited number of widespread and general notions. The fact that the notions are general, helps in guaranteeing that important parts of real-world phenomena can be described by these notions. The fact that the number of notions is limited, on the other hand, does not restrict the domain of application nor the expressive power of the methodology, as the *systemic*[89]nature of *Trinity* (see also Appendix C) enables one to understand complex phenomena in terms of many more simple ones. This brings us to the theoretical point of view.

Once accepting this limited number of empirical generalisations as first principles, and turning from the bottom-up towards the top-down point of view, *Trinity,* in essence, is a systemic interpretation of phenomena in terms of these principles. This is expressed in the ***Trinity* principle** (Chapter 5):

---

[88] In daily life, that is. From a *philosophical* point of view, several fundamentally different stances can be distinguished.
[89] We call a systemic construct a construct that can be interpreted as recursively consisting of other systemic constructs, until an atomic level is reached (this is where the term "construct" is not appropriate anymore).

Any real-world referent can be interpreted as a systemic construct of intentional and autonomous activities, in either one domain, or a combination of the knowledge domain, the physical domain and the communication domain.

The *Trinity* principle states that the referent (the phenomenon of interest, the multi-actor problem context being modelled by means of a perspective) can be interpreted as a systemic construction. It assumes a model and a referent to exist separately (although not independently). By means of modelling steps, the modeller may change his/her interpretation. As was preluded in Chapter 3, theoretically elaborated in Chapter 4, and applied in the design of the *Trinity* modelling language in Chapter 5, transformation steps (abstractions and specifications) enable one to change interpretation of one and the same referent, and building blocks steps (extensions and restrictions) enable one to change scope.

In summary, *Trinity* is a systemic application (the theoretical elaboration) of the notions of intentional and autonomous activities in three different domains (the generalised empirical basis).

11.3 KEY FEATURES OF *TRINITY*

In this section, several key features of *Trinity* will be discussed. In combination, these key features provide an overview of essential aspects of the *Trinity* methodology as a whole.

The *first* key feature of *Trinity* that we will discuss is its explicit and precise (philosophical) definition of problem solving (Chapter 2). Perhaps the most intriguing characteristic of this definition is that problems actually *emerge* as a mis-correspondence, and *vanish* at the very moment that correspondence is re-established. The emergence of the mis-correspondence manifests itself as an awareness of an incomplete perspective and an intention to turn this perspective into a pragmatically correct one. Problem solving is a process during which this mis-correspondence is recognised, and explicit attempts are made to cause it to disappear. At the moment that a pragmatically correct perspective is obtained, the problem vanishes.

It is interesting to note that in this interpretation of "problem" and "problem solving" there is no place for the concept of a *problem definition*: problems cannot be defined, they are felt as situations of unease and uncertainty. It is possible to give an intermediate description of the problem context though (and this is what many persons rather imprecisely would call "problem definition"). For example, it is possible to give a description of an undesired "as is" situation, or a desired "to be" situation, and add that this situation should be deleted or achieved, respectively. In terms of *Trinity*, however, such statements are not problem definitions (indeed, this notion is considered to be a contradiction in terms). Rather, they are intermediate results of the problem-solving process in progress; they are intermediate (parts of) perspectives.

Another characteristic of the *Trinity* interpretation of problems and problem solving is that it explicitly takes into account that it is *not* per se required to reach a *solution.* The mis-correspondence between intentions and environment can vanish by means of constructing an appropriate perspective, which nicely matches the intention with the environment. However, other possibilities are that one can change the environment, or the intention. In addition, it may be the case that a solution presents itself autonomously (for example, by accident). A problem is simply considered to be solved at the very moment that correspondence is re-established, no matter how.

A *second* key feature of *Trinity* is its distinction of intentional and autonomous activities. In combination they compose the very essence of the *Trinity* methodology [Diepenmaat, van Lierop and Bruijnes (1997)]. The fact that *Trinity* enables one to model both these activities explicitly and coherently is, in our view, a crucial requirement for being able to support D-type processes. The reason for this is that they are subject to different cultural and paradigmatic mind sets, and open up different routes and repertoires for intervention, both of which are required *in combination* to intentionally change parts of society. For example, in designing scripts one can either attempt to change perspectives (a knowledge domain strategy), or environments, or both in combination (see also the "throw away battery" example in Chapter 5, where both a material and a human dimension are distinguished).

A *third* key feature is that *Trinity* enables one to model *three* different viewpoints with respect to complex phenomena in *one* model. The three viewpoints correspond with the three domains distinguished in the *Trinity* principle. They are: a) the physical viewpoint (focusing on material flows and material processes, a viewpoint that is rather dominant in the natural sciences); b) the knowledge viewpoint (focusing on perspectives (action potentials) of participating actors and the processes that may change them, a viewpoint that falls within the knowledge management paradigm) and c) the communication viewpoint (focusing on the flow of information between actors, a viewpoint that, for example, is rather dominant in information flow analysis and information technology in general). It is striking that (natural) scientific, knowledge management and information scientific paradigms regard these aspects in isolation. In solving D-type problems, this separation is artificial: the experiments clearly show that all three domains should be given attention, and this in a coherent manner. They constitute a trinity (this is one of the several reasons[90] why the methodology is called *Trinity*).

A *fourth* key feature of *Trinity* is its systemic nature: models can be thought of as recursive constructs of models, until at a certain level the models are considered to be atomic. A description that is too global can be specified, and a description that is too cluttered with detail can be abstracted by means of a transformation strategy. Multiple levels of detail can co-exist. The scope can be changed by means of extensions and restrictions. Multiple

[90] These reasons are explained in Appendix D.

viewpoints and generic models can be constructed and used by means of referent and representation operations. This systemic nature is the key to be able to deal with complexity (in its commonsensical meaning): in terms of KDS, it enables one to use balancing strategies, so as to keep the position of the model below the "cognitive threshold". (In Appendix C the three dimensions of the *Trinity* systems notion are discussed).

A *fifth* key feature of *Trinity* is the notion of *pragmatic correctness*. In addition to being syntactically correct (the model primitives and arrows are combined according to the rules for connecting them) and semantically correct (model primitives can be associated with parts of the three domains, and arrows can be interpreted as causalities), a model of a perspective should be pragmatically correct. Pragmatic correctness means that, when looking at a model, the model can be understood as an intentional activity: the observer may not share the intentions of the actors that play a role in the model, but the model as a whole makes sense to him[91]. It models an action potential. He understands that the intention (the purpose), the perspective (the knowledge), and the environment of the actors involved in combination constitute a coherent whole.

Pragmatic correctness is both a hairy and a practical concept. It is hairy because it depends on (the knowledge of) the observer. For example, for me the sense of bringing flowers home on November 22 is obvious, but for you this may not be the case. Pragmatic correctness is a relativistic notion. A difficulty with relativistic notions is that what is considered to be true depends on the observer, rather than on the observed (see, for example, [Bain (1871)]). According to many scientists, admitting a relativistic touch in a scientific dissertation washes away the firm grounds on which a scientific dissertation should be founded. Rabid adversaries of relativism even claim that relativism is equivalent to "anything goes". We are not fond of an extreme relativism either (for a critique see, for example, [Laudan (1990, 1996)]). But the opposite, rational and positivistic point of view, we consider far too dogmatic an alternative to be of any value in real-world multi-actor problem solving. The presumed "anything goes" nature of relativistic stances we consider to be sufficiently curtailed by the test implied by the *evaluation* stage of our model of intentional activities. Although discussions about recipes for puddings again and again give rise to emotional discussions, the only proof of the pudding is in the eating. In the words of the pragmatist: *"where there is no risk of failure, there is no test involved"* [Laudan (1990) p. 20].

The relativistic notion of pragmatic correctness, however, has a remarkably practical consequence. *Trinity* models reflect what is known of the D-type problem context so far. But, in this respect more importantly, at each step the new intermediate model *suggests ways to proceed as well.* For each intermediate model it is established whether it is pragmatically correct, and (if not) in what ways it deviates from pragmatic correctness. For example, actors (intentional activities) may be missing; the model relation between

---

[91] In more intuitive terms, a model is *pragmatically correct* if it explains why actors want to act, how they intend to do this, and when the actor believes that the environment allows for these actions (the *willing, knowing* and *being able to* of intentional activities).

perspectives and environments may be wrong; the output of an intentional action may disagree with the input of another one, perspectives or environments may be missing, et cetera. All such observations (identifications of pragmatic *in*correctness, missing knowledge) induce subsequent knowledge acquisition steps, for which many different methods may be appropriate, from thinking in isolation to workshops (see also Chapter 5). The requirement of pragmatic correctness is a constant driving and guiding force[92] for next steps in the problem-solving process.

In addition, the notion of pragmatic correctness offers a pragmatic solution to what we will call the *degeneration problem*. The degeneration problem[93] is that in complex situations it may become very difficult to demarcate what and who is part of the problem context and what and who is not. Defining the scope of a complex system of concern is claimed to be difficult, as "everything is related to something else". The system becomes fuzzy. In the case of a D-type problem context, the argument might look something like this:

> *In a D-type problem context many different actors can be distinguished. However, each of these actors, even the ones at the boundaries of the model, is in turn at the centre of a network of other actors. Each of these other actors, even the ones at the boundaries of these models, are in turn at the centre of another network, and so on. This makes it extremely difficult, if not impossible, to properly demarcate a systems boundary. The system degenerates.*

From a *Trinity* point of view, however, the central issue in problem solving is *not* to establish a systems boundary. The central issue *is* to develop a model of a D-type perspective (of which the actors involved assert) that (it) is pragmatically correct. If, in addition, these actors are willing to act in line with the overall perspective, the problem-solving process can stop (or, more to the point, has stopped already). Note that (in terms of the philosophy of Chapter 2) at this point a perspective is obtained that models both the environments and the intentions of actors involved. The degeneration problem is tackled, as the scope is clearly demarcated (albeit as a side effect), and action can begin.

A suspicious reader might remark that a tautology seems to be at stake. The difficult degeneration problem seems to be traded in for the relativistic notion of pragmatic correctness, which hardly seems to be an improvement. This remark seriously undermines the claims and strong points of *Trinity* as a methodology. It should, therefore, not be a

[92] This driving force can actually be experienced in a very direct manner: the model in a sense poses the questions that should be answered in the next knowledge acquisition steps. Discussing (participative mode) or considering (isolated mode) the pragmatic correctness of models sheds light on subsequent problem solving steps.

[93] The degeneration problem is related to the frame problem, which states that when changing a situation, it is extremely difficult (from a philosophical point of view) to define which part of the surroundings of the place of change changes as a result of this change. See also [McCarthy and Hayes (1969)].

surprise that we beg to differ. Whereas pointing at the presence of a degeneration problem merely is a statement of inability and inadequacy, striving for pragmatic correctness offers a route towards improvement. *Pragmatic correctness is what turns knowledge (the mere presence of perspectives, like knowing how and why to open a bottle of wine in general) into action potentials (perspectives coupled with an environment by means of a trivalent model relation).*

We identified manifestations of the degeneration problem several times in evaluating the results of strategic conferences directed at stimulating technological environmental innovation [Diepenmaat, van Lierop, and Roorda (1997)]. The idea of these conferences was that participants would develop plans for concerted actions, that (on a time scale from 5-20 years) would improve the situation concerning issues like the recycling rate of metals, the minimisation of building and demolition waste, and the reduction of packaging materials. It was striking to observe that in each of the three investigated cases, the participants of the conferences were invited on the basis of a procedure that could not be re-established. Typically, it was a mixture of "old chap networks", "feet on table" conversation results, and the merging of several rather personal short-lists; this was embedded in a sauce of best practical knowledge. Bearing in mind that the selection of participants is perhaps the most important variable that influences the outcome of such a conference (next to the organisation and the facilitation of the participative processes), several serious questions arise[94].

According to a *Trinity* point of view, organising a single strategic conference (or any gathering of quite diverse actors) might not be the procedure of preference for perspective development in D-type problem situations. The reason is that a "bootstrap phenomenon" manifests itself: the scope of the problem context is determined by the actors that are invited; and the actors that are invited are determined by the scope of the problem context. Paradoxically, the only way to know who to invite is to know the resulting perspective (the invitation paradox).

In order to avoid this bootstrap problem, strategic conferences should be used *either* to define and refine problem contexts (in which case the procedure for inviting actors, mentioned above, would be appropriate, provided the actors are numerous and diverse), *or* to decide among different perspectives (in which case it would be known in advance what actors to invite: they are participants in, and therefore are referred to in the *Trinity* models of these perspectives).

The bottom line is that in D-type perspective construction typically several, methodically quite different, knowledge acquisition steps will be required. At *every* step during the problem-solving process there should be a clear relationship between a knowledge demand (resulting from a pragmatically incorrect model), the actors to be involved in this step of the problem-solving process, and the methods to support this step. The, at first sight, rather theoretical requirement of pragmatic correctness turns out to be a very practical one: it guides and drives the problem-solving process, and replaces the intangible degeneration

[94] We hesitate to confess that we ourselves have used the same type of selection procedure many times before.

problem by a tangible (albeit not always monotonous) process of convergence towards a model that both motivates and guides intentional action (which is another way of formulating that the model is pragmatically correct).

A *sixth* key feature of *Trinity* is that it is a highly generic methodology: it is hardly restrictive with respect to specific characteristics of the multi-actor problem context of concern. Straightforwardly, this follows from the very general assumptions and principles it is built on (see also section 11.2). In Chapter 3 we defined the type of problems that *Trinity* may support as D-type: a hardly limiting definition indeed. Perhaps the most important requirement for using *Trinity* is that the setting should give room for rational reflective thinking, which presumes a time span in which (intermediate) modelling sessions can be conducted. Settings that claim to prepare complex intentional actions and do not allow for rational and reflective thinking, however, should be regarded with suspicion anyhow.
This does not mean that we consider all facets of D-type problem solving to be rational and open for reflective thinking. On the contrary, ultimately taking intentional action is motivated by a complex mixture of (private, disciplinary, cultural or common) experiences, feelings, likes and dislikes, implicit and explicit value systems. We have no pretensions whatsoever that D-type intentional activities can (or even should) be turned into completely rational processes. We merely think that it is wise to distinguish claims of *pragmatic correctness* (correspondence between *perspectives* and *environments*, dealing with the question whether a perspective is indeed an action potential) from claims of *improvement* (correspondence between *perspectives* and *intentions*, dealing with the question whether executing the script part of a perspective results in a better "to be", i.e. results in a more desired situation). A successful intentional action is based on a pragmatically correct perspective, and (in the eyes of the actor) results in improvement. Preferably, questions with respect to the pragmatic correctness of perspectives should be addressed from a rational point of view as much as possible. Discussions with respect to desirability and improvement should be based on, as clear as possible, descriptions of the situation at stake. *Trinity* offers such descriptions. Therefore, indirectly *Trinity* supports these discussions as well.

The *seventh* and last key feature that we will discuss is the fact that *Trinity* uses *one* language to model situations "as is", scripts, as well as situations "to be" (in combination constituting perspectives). In cases where *Trinity* is used to *design* multi-actor problem-solving processes (resulting in a model that describes the *process thought to result in a perspective*, rather than the very perspective itself, i.e. a shift in problem context is at stake), again the same language can be used. Descriptions, prescriptions, predictions as well as problem-solving designs[95] in many cases can be understood as D-type entities,

[95] *Problem-solving designs* may be understood as the script part *of an intentional activity directed at obtaining a perspective*, i.e. a "subcycle" of the overall parent intentional activity. The "as is" situation of this subcycle is a situation in which a perspective is missing, the "to be" situation is a situation in which the perspective is available.

which makes the *Trinity* modelling language a suitable language to model them. The obvious benefit is that *one* representation paradigm suits several purposes: also in this respect *Trinity* is generic. Until now, we did not use *Trinity* intensively in *designing* multi-actor problem-solving processes. However, on the few occasions that we did so, it proved very valuable because of the fact that *Trinity* offers support in differentiating several problem-solving steps, including the knowledge and skills profiles of participants that implement or participate in these steps.

11.4 REFLECTIONS ON THE *TRINITY* METHODOLOGY

This part of the general discussion is devoted to reflections on the research presented in this dissertation as a whole. First, we will reveal the contours of a background relation (the "backbone") that underlies and ties together the different chapters of this dissertation. Second, we will position *Trinity* in the field of mainstream paradigms for dealing with complexity. Notably, *Trinity* will be compared and contrasted with Systems Dynamics, a both well-known and widespread modelling methodology in complex situations ("messy problems").

11.4.1 Exposing the backbone

All the elements of this dissertation are described as parts of an overall methodological framework. Here, at the end of this dissertation, the backbone relation, that ties together all the parts of this dissertation into one theory, will be roughly explained.

Trinity provides modelling methods that enable one to model parts of society in terms of the *Trinity* principle. KDS provides a means to characterise (changes in) knowledge distributions. Also knowledge distributions model parts of society. The *first* key to the relation, therefore, is the recognition that *both* a *Trinity* model *and* a point in KDS model (are interpretations of) *a part of society*.

A *second* key is that *both* KDS *and Trinity* models are built on the concept of intentional activities.

A *third* key is the recognition that *both* KDS *and Trinity* are designed in accordance with the generic theory of qualitative modelling as presented in Chapter 4[96].

These three keys in combination enable one to derive a correspondence. From the above it follows that a change in a *Trinity* model implies a movement in KDS; and a movement in KDS implies a different *Trinity* model. Therefore, the backbone relation is:

> *Trinity modelling processes correspond with transitions in KDS*

[96] Actually, the generic theory of qualitative modelling was a spin-off of attempts to understand the exact relation between earlier versions of KDS and the *Trinity* modelling language. Once available, the theory proved very useful in improving both *Trinity* and KDS.

This correspondence has some remarkable consequences:

1. knowledge distributions mirror *Trinity* models, and vice versa;
2. knowledge processes mirror *Trinity* modelling steps, and vice versa;
3. *Trinity* modelling strategies can be visualised in KDS, and movements in KDS constitute *Trinity* modelling strategies.

These consequences will be explained below.

Consequence 1: Knowledge distributions mirror Trinity models, and vice versa.

When looking at a *Trinity* model, the number of hexagons in a model corresponds with the notion of **diversity** in KDS: the number of different points of view that are distinguished in the referent. Each of the perspectives represented in the model has a specific **complexity** and **adherence**. This enables one to transform a *Trinity* model into a position in KDS (see also the formulas in figure 1b of Chapter 2).
Conversely, a knowledge distribution (**c**, **a**, **d**) specifies that the corresponding *Trinity* model should encompass **d** hexagons, of which the mean **adherence** and **complexity** should be **a** and **c**, respectively.

Consequence 2: Knowledge processes mirror Trinity modelling steps, and vice versa.

This correspondence will be worked out by means of two examples: one concerning parallel building block steps and one concerning parallel transformation steps.

Building block steps
During a parallel extension of a *Trinity* model, a hexagon is added. This implies that the body of knowledge referred to by the *Trinity* model is increased. In terms of KDS, **diversity** increases by one, and **complexity** and **adherence** are adjusted in agreement with the formulas presented in figure 1b of Chapter 2 ($\mathbf{c.a.d}_{\text{final}}$ is larger than $\mathbf{c.a.d}_{\text{initial}}$). Parallel restrictions of a *Trinity* model would decrease diversity: ($\mathbf{c.a.d}_{\text{final}}$ would be smaller than $\mathbf{c.a.d}_{\text{initial}}$).

Transformation steps
As a result of a parallel abstraction, the number of hexagons in the model decreases (the systemic level increases), but the scope remains the same. In terms of KDS, **diversity** (the number of different perspectives distinguished at the systemic level of concern) decreases. However, as the new *Trinity* model refers to the *same* body of knowledge as before (it is a transformation; a conservation principle applies), **c.a.d** must remain constant. Obviously, **c**

and **a** in combination must compensate the decrease of **d**: the product **c.a** has to increase[97]. Indeed, the hexagons in the resulting more abstract *Trinity* model refer to a larger body of knowledge, which implies that the product of **c** and **a** is larger.

Consequence 3: Trinity modelling strategies can be visualised in KDS, and movements in KDS constitute Trinity modelling strategies.

The correspondence between KDS and *Trinity* enables one to visualise modelling strategies in terms of movements through KDS. We will use iso-planes starting from the origin (**c.a.d** = constant) as a help in explaining this.

Building blocks strategies (extensions and restrictions) imply a transition towards another iso-plane in KDS (figure 1a). Extension strategies correspond with moving towards a higher iso-plane in KDS; restriction strategies correspond with moving towards a lower iso-plane in KDS. This is necessarily so, because the number of atomic perspectives referred to by the model (which equals **c.a.d**, which in turn defines an iso-plane) increases and decreases, respectively.
For example, when you add an intentional activity to a *Trinity* model, a new hexagon is introduced; **d** increases by 1, **a** and **d** are adjusted in agreement with the formulas of figure 1b, Chapter 2.

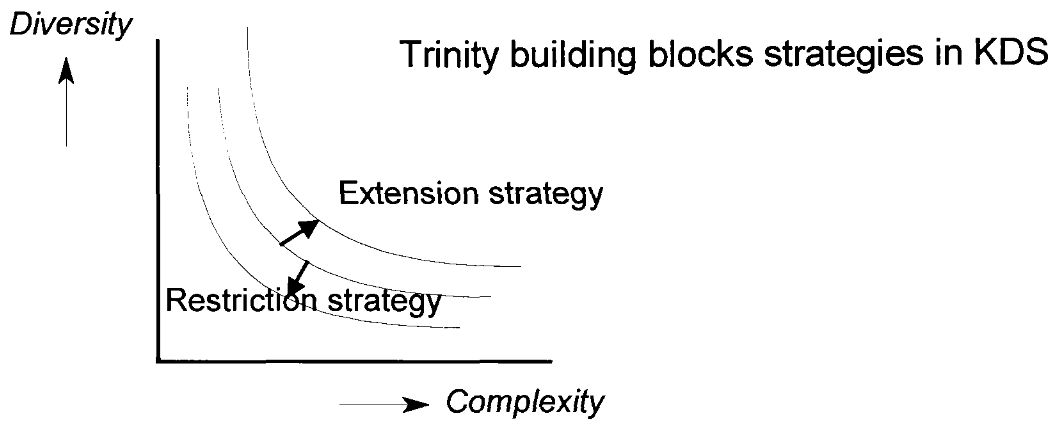


Figure 1a: Two-dimensional examples of *Trinity* building block strategies as movements in KDS.

[97] This can happen in a number of ways: **c** or **a** increases, **c** and **a** increase, **c** increases and **a** decreases, but the positive effect of **c** on the product is larger than the negative effect of **a** on the product, et cetera.

Transformation strategies (abstractions and specifications) correspond with changing position on a specific iso-plane in KDS (figure 1b). This, necessarily, must be so, as the total number of atomic perspectives that the model refers to (which equals **c.a.d**) remains constant (in a transformation, a conservation principle applies). It is merely the interpretation that changes. For example, parallel abstractions decrease **d**; parallel specifications increase **d** (**a** and **c** are adjusted in line with the conservation principle, which implies another position at the same iso-plane).

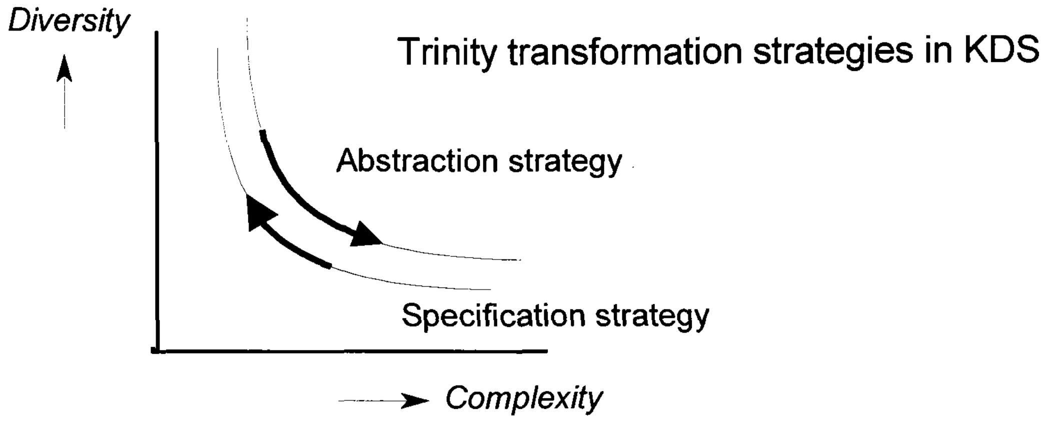


Figure 1b: A two-dimensional example of *Trinity* transformation strategies as movements in KDS.

Figure 1b appears to be rather contra-intuitive after superficial inspection. Abstraction increases **complexity**, and specification decreases **complexity**! It must be remembered, however, that the **complexity** *axis* is *not* the same as the commonsensical notion of complexity[98]. Abstraction increases the **complexity** of the perspectives that are explicitly distinguished in the model but, at the same time, the **diversity** (the number of different perspectives that are explicitly distinguished, the number of hexagons in the model) decreases. Abstraction reduces **diversity** but increases **complexity**. After all, abstraction changes the interpretation of the referent, and not the referent itself.

Balancing strategies perhaps are the most intriguing strategies. When operating near the cognitive threshold, it may be required to actually trade off the level of detail for scope (like in a parallel bird's eye strategy). In this case, first a parallel abstraction is applied, resulting in fewer hexagons referring to the same body of knowledge. Now imagine that the original (the more detailed) level is not maintained implicitly, but rather discarded (for

[98] Perhaps the product **c.a.d** would be better in agreement with a commonsensical notion of complexity.

example, forgotten). Now the perspectives at the new, more abstract level have turned into atomic perspectives (as a lower level of detail is not known anymore). A parallel abstraction, in combination with an upward shift in atomic level, actually decreases the number of atomic perspectives referred to by the model. To be more precise, the complexity of the abstract perspectives (hexagons) present in the *Trinity* model has become 1; **c.a.d** actually decreases. This gives some "space" for extending the scope of the model.

How does this look in KDS? Figure 1c provides an example. The first step, a parallel abstraction, results in a change of position on the same iso-plane. To be more specific, **d** decreases, and **c** and **a** are adjusted; **c.a.d** remains the same. The conservation principle is obeyed: it is a transformation indeed. In the case of a balancing strategy, however, a second step follows in which the lower systemic level is discarded. The complexity of the perspectives referred to in the *Trinity* model decreases to 1 (as they have become atomic). This second step can be visualised, therefore, as a transition through KDS in which **d** and **a** remain constant, and **c** is reduced to 1. The second step resembles a restriction, in that the final model is at a lower iso-plane. Indeed, it *is* a multi-representation restriction: models at different systemic levels can be interpreted as alternative representations of one and the same referent, and one of them is restricted. Finally, in the third step the *Trinity* model can be extended with new hexagons, until the cognitive threshold (which may be interpreted as a specific iso-plane in KDS) is reached again. This corresponds with an increase in **d**.
In short, a parallel bird's eye strategy can be visualised as a movement on an iso-plane towards a lower **d**, followed by a jump towards a lower iso-plane along the complexity axis, followed by a jump towards a higher iso-plane along the diversity axis. This journey through KDS ends where it began. Indeed, the initial model and the final model are identical with respect to their structural complexity (the number of simple model relations does not change, hence the name "balancing strategy"). However, the final model covers a *larger* scope at a *lower* level of detail.

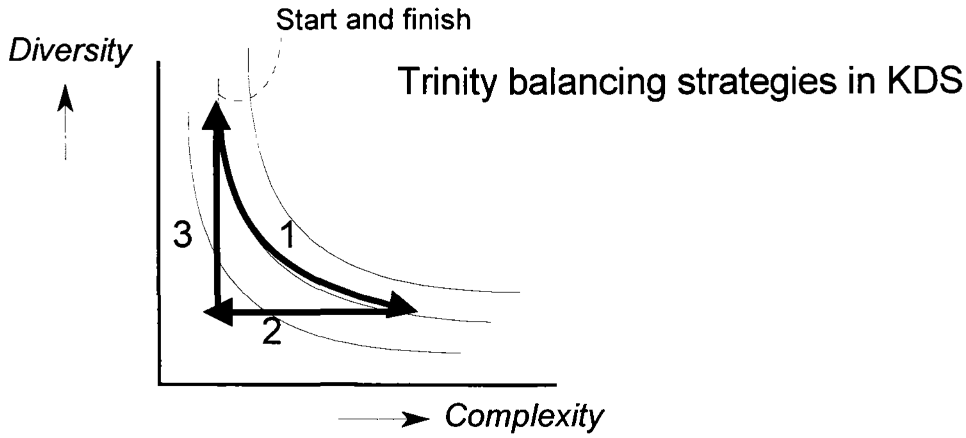


Figure 1c: A two-dimensional example of a *Trinity* balancing modelling strategy as a movement in KDS.

11.4.2 Positioning *Trinity*: mainstream paradigms for dealing with complexity

In this section, we will position *Trinity* in terms of mainstream paradigmatic distinctions. The first distinction is the generative paradigm versus the decision making and the acting paradigm. The second distinction is the continuous, quantitative paradigm versus the discrete, qualitative paradigm.

Generative versus decision making and acting paradigms. In Chapter 2 we positioned *Trinity* with respect to three different interpretations of problem solving: 1) problem solving is acting; 2) problem solving is decision making; and 3) problem solving is the generation of action potential (perspective construction). We showed there that all three interpretations fall within our model of intentional activities. In addition we argued that *Trinity* predominantly falls within the interpretation "problem solving is the generation of action potential". This generative interpretation we consider to be both the most fundamental and the least supported aspect of problem solving.
Generation is the *most fundamental aspect* of problem solving because actions (interpretation 1) *realise* action potentials (actions are motivated and guided by perspectives), and decision making (interpretation 2) *decides among* action potentials (it is not possible to act according to several perspectives). Therefore, both these interpretations depend on (presume) the presence of action potentials. In *Trinity* terminology, action potentials are pragmatically correct perspectives.
In addition, generation in our opinion is the *least supported aspect* of problem solving. An obvious reason for this is that it is difficult to guide and support the construction of

something that is inherently unknown[99]. Within the *Trinity* approach, we circumvented this intrinsic difficulty of perspective construction by emphasising structure above contents. The *Trinity* language structures and supports perspective construction in two important ways. The first way is that it offers a means to represent (intermediate) D-type perspectives. The second way is that it describes all the different ways in which models may be changed from a structural point of view[100]. These different ways are defined in the libraries of modelling steps (Chapter 5). *Trinity* offers syntactic rules. In addition, the semantics of arrows and reference model primitives are specified in a generic manner.
In principle, the decision-making paradigm is incorporated within the *Trinity* modelling language as well, as decision making implies that one or a subset of a larger set of action potentials is selected. This, in terms of the *Trinity* modelling language, amounts to either a representation restriction (in the case of different interpretations of one and the same referent), or a referent restriction (in the case of different possibilities to apply the same perspective). Also, the more complex case may manifest itself, in which only part of the referent or part of the interpretation overlaps (this is possible because decision making typically takes place on the basis of systemically complex perspectives). The decision-making paradigm, however, is not actively *supported* other than by the presence of clear representations of action potentials (*Trinity* models of perspectives).
Which intentional and autonomous activities to distinguish, how to arrange them into a coherent whole, and which methods to select to support knowledge acquisition steps - these are content-related questions that *Trinity* does not answer in specific. The notion of pragmatic correctness, however, offers an important driving and guiding principle in these processes. It is this relativistic notion that covers the content dimension. Future use of *Trinity* is likely to result in further guidelines and rules of thumb in the contents dimension. This opens an enormous research agenda.

Continuous, quantitative versus discrete, qualitative paradigms. A different distinction is provided by the dichotomy of continuous and discrete paradigms. In order to elaborate upon this, we will compare *Trinity* with perhaps one of the most influential methodologies in supporting complex problem-solving processes: Systems Dynamics.
First, we will give a nutshell account of Systems Dynamics (for a more elaborate account, see, for example, [Meadows, Meadows and Randers (1991)], [Vennix (1996)], [Geurts and

---

[99] This does not mean that we consider it easy to decide among known alternatives. Decision making is a very complex phenomenon as well. This is substantiated by the large amount of decision-supporting methods that are described in management science (for example, multi-criteria analysis, expert judgement (delphi's) et cetera).

[100] Decision-making steps correspond with *Trinity* modelling steps that operate on complete perspectives (for example, representation restrictions, referent restrictions). This once more substantiates our remark that almost any method supporting a knowledge acquisition process can be integrated within a *Trinity* process. Decision-support methods (like multi-criteria methods) support restriction steps. Knowledge is the potential to act intentionally, and therefore decisions (although of a restrictive nature) increase knowledge: they increase the potential to act intentionally.

Vennix (1989)], [Senge et al. (1994)]). After that, we will highlight some typical features of Systems Dynamics, and rephrase these features in terms of paradigm characteristics. This in turn enables us to show that the corresponding characteristics of *Trinity* are fundamentally different, and that both *Trinity* and Systems Dynamics have different strong points.

In a minimal description, the Systems Dynamics language might be described as follows:

> *"From any element in a situation (or "variable"), you can trace arrows ("links") that represent influence on another element. These, in turn, reveal cycles that repeat themselves, time after time, making situations better or worse."* [Senge et al., 1994, Chapter 17]

Some authors claim that actually *simulating* the behaviour of system models in time by means of quantitative (mathematical) computer programmes is an important aspect of Systems Dynamics. Senge, for example, uses the heading "Why simulation is essential for systems thinking" in his fifth discipline workbook [Senge et al., 1994, p. 181]. Vennix discusses the pros and cons of both qualitative and quantitative Systems Dynamics in a more balanced manner, and refers to several other authors having commented on this issue before. The main line in these arguments is that, although simulation enables one to obtain a deeper understanding of the behaviour of complex systems in time, in many cases the (often participative) *construction process* of conceptual models (i.e. models that are not formalised by means of a mathematical, quantitative language) in itself already offers important benefits for problem solving and decision making processes [Vennix (1996) p. 108-111]. Checkland, in his Soft Systems Methodology, even emphasises on qualitative models altogether and considers the use of mathematical models (for example, in Operations Research and hard systems engineering) to be limiting with respect to the application area, which does not encompass what he prefers to call "human activity systems" (a concept familiar to our notion of D-type situations and processes) [Checkland (1981)].

From this it follows that Systems Dynamics models contain *variables*, that *at any moment in time* refer to a specific *quantity* (consider, for example, a temperature range). In addition, these variables are related by means of (mathematical) relations. In terms of paradigmatic characteristics, Systems Dynamics models are quantitative and continuous.

In this dissertation, we presented the *Trinity* methodology, a methodology that is discrete, rather than continuous, and qualitative[101], rather than quantitative.

---

[101] We use the term "qualitative" for models that refer to qualities rather than quantities. We want to stress that we consider it a serious misconception to regard qualitative models as imperfect precursors of quantitative models, a misconception that is rather persistent among large sections of the scientific community (for an extreme example, see [Hulthage (1988)]). Qualitative models use a fundamentally different way of describing referents than quantitative models do, a way that

Trinity is *discrete* in that each symbol that is part of a model refers to a *discrete part of the domain-time continuum*. The domain-time continuum is the whole of knowledge domain, physical domain and communication domain states and processes, as it develops in time. As far as *Trinity* is concerned, this continuum is all there is, although the part of interest[102] as well as the interpretation[103] may differ. Note that this is an enormous deviation from the notion of a variable, as manifest in Systems Dynamics: a variable assumes different quantitative values during a continuous time line. The variable does not cease to exist, is not restricted to a small episode, whereas a *Trinity* symbol is.
Trinity is *qualitative* in that each symbol refers to qualities of part of this domain-time continuum. These qualities give meaning to the referent of this symbol. This also is a large deviation from the notion of gradually changing values, as manifest in Systems Dynamics models. It is difficult to perceive *Trinity* models as gradual (continuous) processes of change. Rather, they are systemic constructs of state transitions, each state and transition assigned to a specific *part* of the three domains (specified in terms of qualities, rather than quantities), and a specific *episode* that is small with respect to the episode covered by the model as a whole.

Trinity and Systems Dynamics, therefore, are straightforward opposites in these respects. Table 1 summarises the comparison between *Trinity* and Systems Dynamics, presented above.

Table 1: Paradigm characteristics of *Trinity* and Systems Dynamics.

	DISTINCTIONS	ATTRIBUTES
TRINITY	discrete	qualitative
SYSTEMS DYNAMICS	continuous	quantitative

These differences do have some very practical consequences.
In Chapter 5, for example, we stated that we are not fond of too many cycles in *Trinity* models, as they seriously confuse the time-space continuum that is discretised by means of *Trinity* models. To be more specific: we explained that time cycles in *Trinity* models introduce ambiguity. Cycles should be removed as much as possible (for example, by means of applying parallel dynamic specifications). The reason is that *Trinity* models containing cycles cannot be localised in the domain-time continuum in a clear manner. This is problematic when talking about intentional activities. Intentional activities take

is not superior or inferior but simply emphasises discrete features of referents rather than continuous ones.

[102] Changing the part of interest implies moving *between* iso-planes in KDS.

[103] Changing interpretation of a specific part of the continuum implies moving *in* an iso-plane in KDS.

place in a well established idea of "this-now-here" (see also Chapter 2), which implies awareness of one's action potentials, which in turn implies an intentional actor. Intentional activities are examples *par excellence* of discrete and qualitative entities. In addition, the very notion of "this-now-here" imposes a discrete world view: *this* implies a that; *now* implies a past and a future; and *here* implies a there. In combination they cover all there is[104].

In Systems Dynamics, on the other hand, variables and cycles are considered to be of great importance, as they enable us to understand the changes (the behaviour) of systems in time in terms of changing variable values. As was explained above, this, however, is not in compliance with the discrete nature of actors and actions, as worked out in *Trinity*. From a *Trinity* point of view, this is considered to be an omission: only intentional actors can realise intentional change, and in our view a discrete and qualitative paradigm is more appropriate in this respect.

From the discussion above, it follows that *Trinity* and Systems Dynamics are quite different, and even opposites, in rather fundamental respects. In *Trinity* we impose a *discrete* point of view on a domain-time *continuum*; in Systems Dynamics a *continuous* point of view is imposed on a discretised referent (discrete in terms of the variables, as these encompass all that can be known about the referent by means of the model).

Trinity focuses on actors, which we consider to be one of its major strengths. Actors are the entities that may cause and experience, like and dislike environments. Actors, as defined within the *Trinity* methodology, tend to think in action-related, discrete, qualitative, subjective terms such as support, obstacles, ways through, and high temperatures. What is more: actors **act**: they intervene in the environment, i.e. the referent of their perspective. In Systems Dynamics rather neutral, continuous, quantitative, objective terms such as temperature, mass, economic growth indicators dominate its structure. *Trinity* and Systems Dynamics are complementary, in this respect.

It has been mentioned before that such differences in paradigm characteristics impose different restrictions. For example, Michie and Johnston discussed a) differences between physics, as an example of formal, mathematical disciplines, and b) coping with daily life, as an example of a goal-directed, qualitative "discipline". They conclude that, while physics includes important concepts such as mass, force and energy, there are many more that are just as important but are not included. Examples are closure, containment, support, contact, obstacles, ways through. Without these, physics is incapable of dealing with the real world [Michie and Johnston (1984) p. 186]. We agree with Mitchie and Johnston that mathematical, continuous approaches are incapable of dealing with real-life settings. On the other hand, we acknowledge the importance of a good understanding of the behaviour of complex systems in time, an aspect that is addressed very explicitly by Systems Dynamics.

---

[104] All there is, that is, *with the exception of the observer.* Gödel had a point.

From the above, it follows that the relation between continuous, quantitative paradigms and discrete, qualitative paradigms is not an easy matter to discuss. A good approach would perhaps be to design a methodology that explicitly and discretely models systems of intentional and autonomous activities, and on top of this enables one to simulate the behaviour of such systems in time (which would perhaps require the use of discrete event simulation, see [Zeigler (1984)]). In this way, the benefits of understanding the behaviour of complex systems in time and the ability to model purposeful schemes for intervention and improvement could be merged.

Systems Thinking. Above, we emphasised paradigmatic differences between Systems Dynamics and *Trinity*. However, there is also an important agreement between the two approaches: both are based upon a kind of systems thinking. The notion of systems thinking appeared already several times in this dissertation. Checkland [1981] mentions as most important features of systems approaches that they deal with two pairs of concepts: emergence and hierarchy, and communication and control.
In Chapter 2, we discussed emerging and vanishing properties and systemic levels. The notion of hierarchy we traded in, though, for the notion that every level is appropriate (and indeed distinguished) for a different purpose. This interpretation emphasises that systems levels are symbiotic, rather than hierarchical. It is true that a general leads his troops, but it is equally true that troops allow the general to lead them. In combination, this allows for achieving a mutual goal, the goal that enables one to understand the army as a whole.
In Chapter 5, we discussed communication. We reserved one of the three *Trinity* domains for communication phenomena: a central position indeed. The notion of control we traded in, though, for the notion that every intentional activity should encompass an evaluation, as the real world as we know it is too complicated to be interpreted as a deterministic object. This emphasises that intentional activities as parts of larger systems are symbiotic, rather than dependent upon controllers. It is true that a controller controls the activities of other persons, but it is equally true that co-operation will break down if partners in co-operation do not experience some benefit during their private evaluations. It is mutual benefit that is the rationale for multi-actor contexts, and not control.
From the above it follows that, although emergence and communication are *central and essential* features of our systems concept, notions like hierarchy and control (and related notions like power) are not. It is possible, though, to distinguish situations in *Trinity* models that call for descriptors as hierarchy, control and power. These are, however, extremes in a spectrum of interpretations, rather than elementary features.

Above, we positioned *Trinity* within the field of mainstream paradigms in supporting the dealing with complex situations. At the moment of writing this, we are conducting a research directed at positioning *Trinity* in the field of a number of more specific methods and approaches. The results of this research will be published as [van Lierop (1997)].

11.5 ADDED VALUE OF *TRINITY*: A METHODOLOGICAL POINT OF VIEW

The added value of using a methodology that is intended to support real-world problem solving is difficult to establish in a truly objective manner. The most important problem is the lack of an independent *reference* problem-solving process (i.e. a process in which the methodology is *not* used: the *blanco*): each D-type process is unique, and therefore such a reference does not exist. It is, however, possible to describe as clear as possible the features of a methodology that one considers to be supportive. This is exactly the procedure that we will follow in this section.
The added value of *Trinity* can be addressed from many different angles. Perhaps the first thought coming to the mind is that added value is to be found predominantly in the *use* layer of the methodology, as this is where the benefits of using a methodology are experienced. However, in the introductory part we mentioned that, in our view, the five layers of the methodological pyramid are *symbiotic*, rather than *hierarchic*. A logical consequence of this stance is that *all* the layers encompass a part of the support, hence added value that *Trinity* has to offer. It is true, though, that the arena where this added value is to be established is the *use* layer (for an in-depth discussion of this aspect see the experimental part of the dissertation). Below, we will address the five layers of the pyramid, and highlight the aspects in which *Trinity* offers support from a methodological point of view.

1. Philosophy
In the Philosophical background part of this dissertation, we have given a firm underpinning to the central concept of this dissertation (problem solving). The interpretation of problem solving as the attempts of an actor to re-establish correspondence between intentions and environments constitutes the philosophical corner stone of *Trinity*. In line with this, the goal of an actor, engaged in a problem-solving process, is to come up with a pragmatically correct model of a perspective: *problem solving is perspective modelling*. The availability of this well-defined description of problem solving and the task of a problem solver, in our view, is a very supportive feature of *Trinity*: it provides problem owners with a concrete goal (a model of a three-fold perspective) as well as a concrete means to attain this goal (the *Trinity* modelling language). In addition, the notion of pragmatic correctness constantly drives and guides the problem-solving process.

2. Theory
In the Theory part of this dissertation, two main themes were elaborated: the concept of knowledge distributions (Chapter 3) and a generic theory of qualitative modelling (Chapter 4). As these themes are made operational in the methods layer, it is not required to address their added value at this place.

3. Methods

In essence, the *methods* layer consists of the *Trinity* modelling language and the possible variations in its use. This language is the "operational machinery" to develop D-type perspectives. The design of the *Trinity* language is completely intertwined with the philosophy and theory layer of the *Trinity* methodology and, as a result, the language is especially dedicated to support the process of D-type perspective construction. This support is manifold. The language:

- supports the distinction of autonomous and intentional activities;
- (therefore) emphasises on actors, their intentions and their relations;
- offers *one* representation convention that supports analysis, script construction as well as prediction activities (and this in combination);
- explicitly supports thinking in terms of knowledge domain, communication domain and physical domain states and processes, as well as the ways in which these domains interact (multi-domain networks);
- provides flexible (systemic) ways to *adapt* models (including multi-layer, multi-representation and multi-referent models) on top of *representing* them;
- provides a rich vocabulary of modelling strategies that are supported by libraries of modelling steps;
- enables one to use a variety of different modelling approaches (examples are isolated, hidden, participative use; trouble shooting, trick exploiting, back-casting approaches; the development of alternative candidate perspectives, the modelling of different (possibly conflicting) points of view);
- encompasses the notion of pragmatic correctness, a notion that guides and affects the knowledge acquisition process in important ways.

From the above it follows that at the methods level *Trinity* is perhaps a methodical framework, rather than a single method. The philosophy layer provides the contours of problem-solving processes, but within these contours a large number of different approaches may be followed. This may be rather confusing at first[105]. It should be emphasised, though, that a basic understanding of *Trinity* requires an understanding of the self-contained notions of perspectives, intentional and autonomous activities, and the knowledge, communication and physical domain. In addition, the ways in which these notions can be recursively modelled by means of the four model primitives and the five types of arrows should be understood. The umbrella concept of pragmatic correctness will drive the problem-solving process. From a first principle's point of view, this is all there is. All the methods are variations on these themes.

---

[105] This can perhaps be compared with staff notation in music. By means of a simple notation scheme the playing of quite different types of music (jazz, blues, baroque) and the use of quite different composing strategies (isolated, participative, using schemes, et cetera) can be supported.

4. Tools

The *tools* layer is the least developed layer of the *Trinity* methodology. As a tool we are currently using a flowcharting programme, adapted to suit the notation conventions of the *Trinity* modelling language. The tools we prefer to use in initial stages of the modelling process are whiteboard and markers, or pencil and paper (A3) in smaller groups.

5. Use

The use layer is the layer where all the different forms of support and benefits mentioned above should manifest themselves. In the experiments part of this dissertation, we already discussed the specific added value of using *Trinity* in the three experiments (we will not repeat those discussions here). Indeed, the different aspects of added value described above in generic terms (notably the philosophy and methods layer) manifested themselves constantly throughout the experiments. When confronted with a D-type problem context, *Trinity* provides the problem solver with a goal as well as a "machinery" to achieve it. In a sense, *Trinity* is a methodological lens that highlights and magnifies certain aspects of D-type problem contexts, notably the players and the multi-actor network in transition, that are very crucial in attempts to intentionally realise improvement. Although practical guidelines with respect to the relation between problem characteristics and specific *Trinity* methods have to be elaborated further (a first attempt to describe this relation is presented in Chapter 10), in all three experiments we were able to tune the methodical framework quite easily to the specific needs. Indeed, nothing is as practical as a suitable methodology.

CHAPTER 12

CONCLUSIONS, RECOMMENDATIONS AND FUTURE RESEARCH

12.1 GENERAL CONCLUSIONS

The overall goal of the research, presented in this dissertation, was to develop a means that supports multi-actor processes in general. We decided to focus our research on developing modelling means that support the very process of obtaining a coherent and in-depth understanding of multi-actor problem contexts. Such an understanding, although generally considered to be an important prerequisite for purposeful intervention, again and again proves to be difficult to attain, and therefore is lacking in many cases.

In line with this, we addressed one central research question and two sub-questions (see also figure 1 in section 1.2):

> Is it possible to design *modelling methods* that specifically support problem-solving processes in multi-actor situations?
>
> Is it possible to develop a *philosophical and theoretical basis* that positions the central concepts of this dissertation ("problem solving", "multi-actor", "modelling", and "model-based support") and, as such, provides a foundation for the envisaged modelling methods?
>
> What can be said about the *use* and *added value* of the envisaged methods, when applied in real-world multi-actor problem solving?

We started with the first sub-question. In answer to this question, we developed a philosophical and theoretical basis for the envisaged modelling methods.
The central element of the philosophical basis is our definition of problem solving. This definition proved to be the cornerstone of the *Trinity* methodology as a whole, as it provided both a goal (a perspective) and a means to attain this goal (modelling).
Central elements of the theoretical basis are KDS (and more specifically the notion of D-type situations and processes) and the generic theory of qualitative modelling. As was

envisaged in section 1.2, these theories were of crucial importance in designing clear methods.
In conclusion, the philosophical and theoretical basis that we developed turned out to be a solid foundation for designing modelling methods.

This enabled us to start addressing the *first part* of the central research question: the design of modelling methods. In full compliance with the philosophical and theoretical basis, we designed a methodical framework that supports D-type problem-solving processes. The methodical framework consists of:

- a modelling language, especially tailored to model multi-actor situations and processes, and to adapt these models;
- several modelling strategies that enable us to change model relations in complex ways, and, in addition, provide an instrument to deal with complexity in flexible ways;
- the means to use *trouble-shooting*, *trick-exploiting*, *back-casting*, or combined approaches;
- the possibility to develop and use libraries of generic models; and
- the possibility to insert different knowledge acquisition methods, and to use different communication modes.

The availability of methods enabled us to turn to the *second part* of the central research question, concerning the support of multi-actor problem solving. This issue is addressed by the second sub-question. Three different experiments in real-world multi-actor problem solving were conducted. This showed, in agreement with our purpose, that *Trinity* can be used and offers support in quite different D-type problem contexts (for a detailed account of experiment-specific support, we refer to the Experiments part of this dissertation):

> In the *Indoor environmental problems* experiment, the "as is" models describe quite different D-type problem manifestations in a generic manner. In addition, the models point out many different routes to "branch off" the causal flux of events resulting in sub-clinical health effects. The models, therefore, support both diagnostic as well as remedial activities in this D-type problem context.
>
> In the *VOC2000* experiment, the problem areas of the list proved to be causally connected in complicated ways. Understanding this network dimension was shown to be important in future attempts to improve the situation. Our recommendations, which followed from the model-induced understanding of the problem context as a whole, will be the basis for future activities directed at improvement.
>
> Finally, in the *Building and demolition waste* experiment, the "to be" model induced a reconsideration of actions that were agreed upon in an earlier stage, and highlighted several missing aspects and actors that are indispensable in attempts to realise this future.

In more general terms, *Trinity* proved to be supportive in the following ways. Multi-actor situations became better understood. The design and thinking through of (chain effects of) potential interventions in and improvements of multi-actor networks were supported by the model(s). The *Trinity* process in addition guided and supported the knowledge acquisition processes (including the communication with informers, consultants and field players). Notably its actor orientation, both in isolation and as parts of networks, provides the basis for these different ways of support.
In all three experiments it proved to be relatively easy to integrate the use of *Trinity* in "business as usual". Further research is required to clarify the precise relation between subtypes of D-type problems and specific ways of using the methodical framework.

Our final conclusion is that we succeeded in designing modelling methods that specifically enable support of problem-solving processes in multi-actor situations. *Trinity* is the case in point: *Trinity* offers model-based support for multi-actor problem solving.

12.2 RECOMMENDATIONS

Although several users so far have claimed that *Trinity* supports and makes explicit what they were doing already, albeit implicitly, our experience is that explicit attempts for a coherent understanding of D-type problem contexts are a rather rare phenomenon. The general practice is that D-type interventions and processes are performed on the basis of a rather fragmentary and intuitive understanding of the problem context of concern. Considering the importance of possessing a thorough overview of D-type problem contexts and the added value that *Trinity* has to offer in achieving such an overview, we strongly recommend the use of *Trinity* in D-type problem solving.

This implies a knowledge diffusion process. In order to further this diffusion process, three different routes may be followed simultaneously:

1. *the recognition and awareness of D-type problem ownership should be furthered.* We have presented the ABCD typology on many occasions. It was striking to observe that this was a great help for members in the audiences to start realising that the multi-actor characteristic was a dominating feature in their line of work. Obviously, the notion of D-type problems must be known, in order to be able to recognise them. One step further, *Trinity* can be presented as a powerful help.
2. *a practical handbook should be made available*[106]. A first step towards such a handbook has been made [Diepenmaat (1997)].
3. *several Trinity tools should be developed.* Examples are tools that specifically support participative use, and tools that specifically support the construction and adaptation of *Trinity* models (see also section 12.3, research topic 8).

[106] A good example is "The Fifth Discipline Fieldbook" [Senge et al. (1994)].

12.3 FUTURE RESEARCH

This last section will present several topics that require further research. Several of them have already been eluded to in other places in this dissertation.

Research topic 1: Extend experimental basis.
The experiments we presented in this thesis, although quite different (see Chapter 6 and 10), did not cover all the different ways in which *Trinity* can be used. Since then, we have used *Trinity* in the evaluation of several more strategic conferences [Diepenmaat, van Lierop, and Roorda (1997)]. In addition, we used *Trinity* in a context analysis for a national environmental information centre [Roorda, Wiersma, de Zeeuw, and Diepenmaat (1997)]. In the latter study, the emphasis was predominantly on activities in the communication domain (information flow analysis). In all these cases, using *Trinity* proved to be very useful. However, we did not yet implement and evaluate complete perspectives in complex D-type problem contexts. The empirical evidence that we possess at this moment, however, is quite encouraging and promising in these respects. For this reason, we foresee a period in which the experimental basis of *Trinity* (i.e. its use in all stages of complex intentional activities, as well as reflection on this use and feed-back towards the other layers of the methodology) will be extended.

Research topic 2: Obtain a better understanding of the relations between D-type problem characteristics and specific modes of using *Trinity*.
In this dissertation we described the application area of *Trinity* as "D-type problems". However, at the same time we presented several different modelling strategies and modes of using *Trinity*. Although we presented some guidelines (for example, in case of a need for exploring the future, use a back-casting approach; in case of an urgent problem, follow a trouble-shooting approach; in case of a lot of study material (documents; interview reports), perhaps a parallel extension strategy followed by a parallel abstraction strategy will do well; in case of many non-professional D-type problem solvers, perhaps *hidden use* is to be preferred), they are rather scattered, incoherent and incomplete. Parallel with increasing the experimental basis (see topic 1), clearly described approaches and a knowledge base relating D-type problem characteristics to these approaches should be developed.

Research topic 3: Develop libraries of knowledge acquisition methods
The goal of a *Trinity*-supported problem-solving process is to obtain a pragmatically correct model of a perspective. During this process, the evolving model is necessarily pragmatically *in*correct. Different forms of incorrectness may be defined, inducing different knowledge needs. Different knowledge needs may require different knowledge acquisition methods and knowledge acquisition tools (acquisition used in a very broad sense). Obviously, there is an enormous amount of scientific literature concerning knowledge acquisition methods and tools. What is missing, however, is the link with problem-solving processes in practice. Future research should be directed at typologies of

situations in models that are pragmatic incorrect, typologies of the knowledge needs they induce, methods and tools that may be used to meet these knowledge needs, *and especially the relations between these typologies.*

Research topic 4: Elaborate the notion of D-type problem-solving design
Several times we used *Trinity* in designing multi-actor problem-solving processes (not reported). In these situations, the referent being modelled by means of a *Trinity* model is the process *resulting in* a perspective, rather than the very perspective *resulting from* this process. Especially in D-type problem solving, designing problem-solving processes is problematic as such, and for this reason we consider this to be an important research topic. The results of research topic 3 will be very valuable in this research.

Research topic 5: The influence of using *Trinity* on D-type problem-solving processes
Using *Trinity* in general can be integrated in "business as usual" without any problems: problem owners do not feel that this disturbs the process or negatively changes it. On the other hand, using *Trinity* changes the perspective of the problem solver, and therefore the problem-solving process is likely to change. With respect to "business as usual" we consider this change to be an improvement. In a sense, *Trinity* provides an "ontological lens". As any lens, however, the *Trinity* lens shows some features more clearly, but others less clearly. For this reason, comparative research should take place, preferably comparing "unsupported" D-type problem solving with several forms of support (for example Systems Dynamics and *Trinity*).

Research topic 6: Simulation of *Trinity* models
Simulation is a strong point of Systems Dynamics. In *Trinity*, at this moment, simulation is not possible. Therefore, the development of some sort of *discrete event simulation* for *Trinity* models is an important research goal.

Research topic 7: Libraries of generic models
Libraries of generic models, provided that they a) cover the domain of application and b) exhibit a delicate balance between re-usability and non-triviality, might have an enormous influence on both theory development and D-type problem solving in the real world. In this dissertation we presented some examples (notably the minimal environmental situation of concern and the indoor "as is" model). However, both domain-independent and domain-dependent libraries could be useful in many different problem situations, not as cookbooks but rather as examples that help understand problem contexts in terms of larger chunks. This provides an enormous research agenda in itself.

Research topic 8: Tools layer research
With tools layer research we mean research directed at tools that support the very *Trinity* modelling process. It is to be expected that a computational tool to construct and adapt *Trinity* models, *tailored to the ways in which Trinity modelling takes place in practice*, will lower the threshold for novice users, and will greatly enhance the speed and ease with

which complex and multi-level models can be constructed. We plan to develop a knowledge brokers workbench: a programme that supports the problem-solving process from a *Trinity* point of view in a number of different ways (for example by means of supporting the construction and maintenance of process and intention lists, see section 5.6; by means of providing more flexible model editors; by means of supporting correspondence maintenance between process/intention lists and *Trinity* models; by means of providing facilities to manage and use libraries of generic models). In addition, we plan to develop tools to further support participative use of *Trinity*. An example of such a tool is a combination of a large whiteboard, prefabricated magnetic model elements (hexagons, ellipses, et cetera) that allow for putting text on them, and the use of text markers to connect these model elements, when positioned on the whiteboard, by means of arrows.

Research topic 9: Backbone research: the systems theory underlying *Trinity*
As was discussed in the General Discussion (Chapter 11), the *Trinity* methodology can be thought of as being built on one background theory. The notions of KDS, the theory of qualitative modelling and *Trinity* modelling strategies are different facets of one underlying systems concept. This systems concept distinguishes *three* different, yet tightly related dimensions (this in contrast with most other systems notions, that distinguish *two* dimensions: wholes and parts). This systems concept is worth further clarification (which would imply a systems theoretical line of research). Appendix C provides some first directions.

The topics mentioned above constitute quite a research agenda. This is not surprising, as the long-term research goal is to turn D-type problem solving into a profession, rather than an art, and D-type problems are perhaps the most challenging problems that exist. This thesis contributes to attaining this goal: *Trinity* offers model-based support for multi-actor problem solving.

APPENDIX A

A COMBINATORIAL SCHEME OF MODELLING PROCESSES

Input relation	**Output relation**	**Type of change**	**Direction**	**Name of modelling step**
S	S	*	*	not primitive
S	P	*	D	not primitive
S	**P**	**B**	**I**	**Parallel extension**
S	**P**	**T**	**I**	**Parallel specification**
S	Mf	*	D	not primitive
S	**Mf**	**B**	**I**	**Multi-referent extension**
S	**Mf**	**T**	**I**	**Multi-referent specification**
S	Mp	*	D	not primitive
S	**Mp**	**B**	**I**	**Multi-representation extension**
S	**Mp**	**T**	**I**	**Multi-representation specification**
P	S	*	I	not primitive
P	**S**	**B**	**D**	**parallel restriction**
P	**S**	**T**	**D**	**parallel abstraction**
Mf	S	*	I	not primitive
Mf	**S**	**B**	**D**	**multi-referent restriction**
Mf	**S**	**T**	**D**	**multi-referent abstraction**
Mp	S	*	I	not primitive
Mp	**S**	**B**	**D**	**multi-representation restriction**
Mp	**S**	**T**	**D**	**multi-representation abstraction**
P	**P**	**B**	**I**	not primitive
P	**P**	**B**	**D**	not primitive
P	P	T	*	not primitive
P	Mf	*	*	not primitive
P	Mp	*	*	not primitive
Mf	**Mf**	**B**	**I**	not primitive
Mf	**Mf**	**B**	**D**	not primitive
Mf	Mf	T	*	not primitive
Mf	P	*	*	not primitive
Mf	Mp	*	*	not primitive
Mp	**Mp**	**B**	**I**	not primitive
Mp	**Mp**	**B**	**D**	not primitive
Mp	Mp	T	*	not primitive
Mp	P	*	*	not primitive
Mp	Mf	*	*	not primitive

(legend see next page)

Legend:

S	=	simple model relation
P	=	parallel model relation
Mf	=	multi-referent model relation
Mp	=	multi-representation model relation
B	=	building blocks approach
T	=	transformation approach
I	=	increase complexity
D	=	decrease complexity
*	=	any

Elaboration of the asterixes would result in 64 combinations (4*4*2*2).

APPENDIX B

TRINITY MODELLING CONVENTIONS: PRAGMATIC CORRECTNESS

A *Trinity* modelling process stops if the modeller thinks that the model models both his environment and his intention. The model is elaborated upon sufficiently to facilitate (motivate and guide) intentional actions. The modeller is prepared to act according to the script part of the model.

A *Trinity* model of a perspective should meet several modelling conventions. We distinguish three different types of modelling conventions: *syntactical conventions, semantic conventions* and *pragmatic conventions.*

Syntactical conventions are the *notation rules* that a model must obey. Checking the syntactical correctness of a *Trinity* model is possible without understanding anything about the referent this model is modelling. A *syntactically correct* model obeys the syntactical conventions.
Semantic conventions deal with the *meaning* of a model: a model can be syntactically correct, but semantically wrong. This is the case if its referent cannot be understood (does not make sense). Conversely, a model can be semantically correct but syntactically wrong. Consider, for example, two rectangles connected by means of an arrow. The referent of rectangle 1 may indeed be caused by the referent of rectangle 2, but the syntactical conventions require that a process (an ellipse) should be modelled in between. A *semantically correct* model obeys the semantic conventions.
Pragmatic conventions deal with the question of whether a model describes an intentional activity. Pragmatic correctness implies that the model obeys both the syntactical and semantic conventions. Being pragmatically correct, therefore, is the most severe requirement for a *Trinity* model. The goal of a *Trinity* modelling process, however, is to obtain a pragmatically correct model: at the very moment that this is achieved, the problem-solving process can stop and implementation can begin.

The syntactical, semantic and pragmatic conventions are systemic conventions: every part of a *Trinity* model should meet them. For example, when an "as is" model is part of a pragmatically correct *Trinity* model, the fact that the model as a whole is pragmatically correct (the actor is the problem owner) implies that all the hexagons that are part of this "as is" model obey the pragmatic conventions as well. When a multi-level model is pragmatically correct, this implies that all the levels are pragmatically correct. When a multi-representation model is pragmatically correct, this implies that the partial models of points of view are pragmatically correct.

Some of the conventions have already been presented in section 5.4. An overview is presented below:

Syntactical convention 1:
A syntactically correct model may consist of *four* types of reference model primitives (domain *states*: rectangles, rounded boxes, hexagons; and domain *processes*: ellipses) and *five* types of arrows (from a state to a process; from a process to a state; from a reference model primitive to corner 1 of a hexagon; from corner 5 of a hexagon to a process; and from a state to corner 6 of a hexagon).

Syntactical convention 2:
In a syntactically correct model, the input states of an ellipse must be of the same type as the output states of this ellipse. For example, it is illegal to have a hexagon as input, and a rectangle as output (this would imply that a knowledge domain state is converted into a physical state). The only exception is an arrow that departs from a hexagon (position 5): this signifies that the process is caused by the perspective referred to by the hexagon (it is a "start implementation" or "start action" arrow).

Syntactical convention 3:
It is illegal to connect rectangles (referring to physical domain states) and rounded boxes (communication domain states) by means of an arrow: they should be separated by means of an intermediate ellipse. The same holds true for hexagons, as far as connections at the middle of the sides are considered.

Syntactical convention 4:
Reference model primitives must be separated by means of arrows.

Semantic convention 1:
Reference model primitives are given a name that allows for semantic correspondence: the modeller should be able to link this model primitive to its referent (i.e. to understand the model relation between the primitive and the referent it represents). Hexagons must refer to perspectives; rectangles must refer to physical states; rounded boxes must refer to communication domain states; and ellipses must refer to processes in a domain.

Semantic convention 2:
Arrows derive meaning from the two reference model primitives they connect. It should be possible to interpret them as a causation. If several arrows enter a model primitive, the causations *in combination* cause the referent of this model primitive (this is equivalent with a logical "and"). Likewise, if several arrows leave a reference model primitive, these causations all result from the referent of this model primitive.

Pragmatic convention 1:
An arrow from a reference model primitive to a hexagon (position 1) should be interpretable as a perception, resulting in an acknowledgement.

Pragmatic convention 2:
An arrow from a hexagon (corner 5) to an ellipse should be interpretable as the start of an action (an implementation of the script part of the perspective modelled by the hexagon). The ellipse should refer to the same process as the script part of the perspective.

Pragmatic convention 3:
An arrow from a domain state to a hexagon (corner 6) should be interpretable as an evaluation, in which the domain state is being compared with the "to be" part of the perspective, modelled by the hexagon. The domain state should refer to the same referent as the "to be" part of the perspective.

Pragmatic convention 4:
Domain states (rectangles, rounded boxes and hexagons) must be connected to a perspective by means of an arrow. This convention ensures that a model does not contain redundant parts. For example, a physical state that does not trigger an actor, provides an "as is" situation, results from an action, or is evaluated, does not fulfil any purpose (is useless), and as such it should be left out of the model. This convention does not apply to *domain processes* (ellipses). The reason for this is that an ellipse that is not connected to a hexagon simply refers to an autonomous process. See, for example, the autonomous process in the *Minimal environmental situation of concern* generic model.

Pragmatic convention 5:
A pragmatically correct model is a *parallel* model. This must be so, as a multi-referent or multi-representation model relation would imply that still several options to take action are left open, and a decision has yet to be made (which implies a decision-making process, which in modelling terms amounts to a referent reduction strategy or a representation reduction strategy).

Typically, the modelling process is a rather explorative process, with many intermediate models that do *not* meet the syntactical, semantic and pragmatic conventions. Modelling steps are a great help in this respect, as they are syntax-preserving. Starting with a syntactically correct model and applying modelling steps, it is impossible to violate the syntactical conventions. The **goal** of a *Trinity* modelling process, nonetheless, is to obtain a pragmatically correct model: a model that obeys *all* the conventions mentioned above. The pragmatic correctness of the model of a perspective at the moment just before acting is, however, a *hypothesis*: the forthcoming actor believes it to be pragmatically correct. The proof of the pudding is in the eating: his knowledge may be wrong. Only with hindsight (by means of an evaluation process) can pragmatic correctness be established. This is the way in which experience is gained. It is good, however, to keep in mind the wise words of Heraclites: "One cannot step into the same river twice".

Figure 1 presents some example models that illustrate the conventions.
The first example is syntactically incorrect: a hexagon (perspective) cannot be transformed into a rectangle (a physical state).
The second and third examples are syntactically correct, but semantically incorrect. In the second example, the rectangles "Blzzbrp1" and "Blzzbrp2" do not facilitate semantic correspondence: they are in conflict with semantic convention 1 (which implies that they are in conflict with semantic convention 2). In the third example, the reference model primitives "Giraffe", "Corrosion" and "Camels" do facilitate semantic correspondence: semantic convention 1 is obeyed. However, the arrows cannot be interpreted as causations: giraffes do not usually corrode, resulting in camels: semantic convention 2 does not apply.
The fourth example is semantically correct, but syntactically wrong. Indeed, the sun may cause a wall to become hot. Both semantic conventions are obeyed. However, syntactical convention 3 is violated. An ellipse, referring to a process, is missing.
The fifth example is syntactically and semantically correct. However, the model is not connected to a hexagon: it does not meet the pragmatic conventions. The question to be asked would be: who is interested in this model, and for what reason?

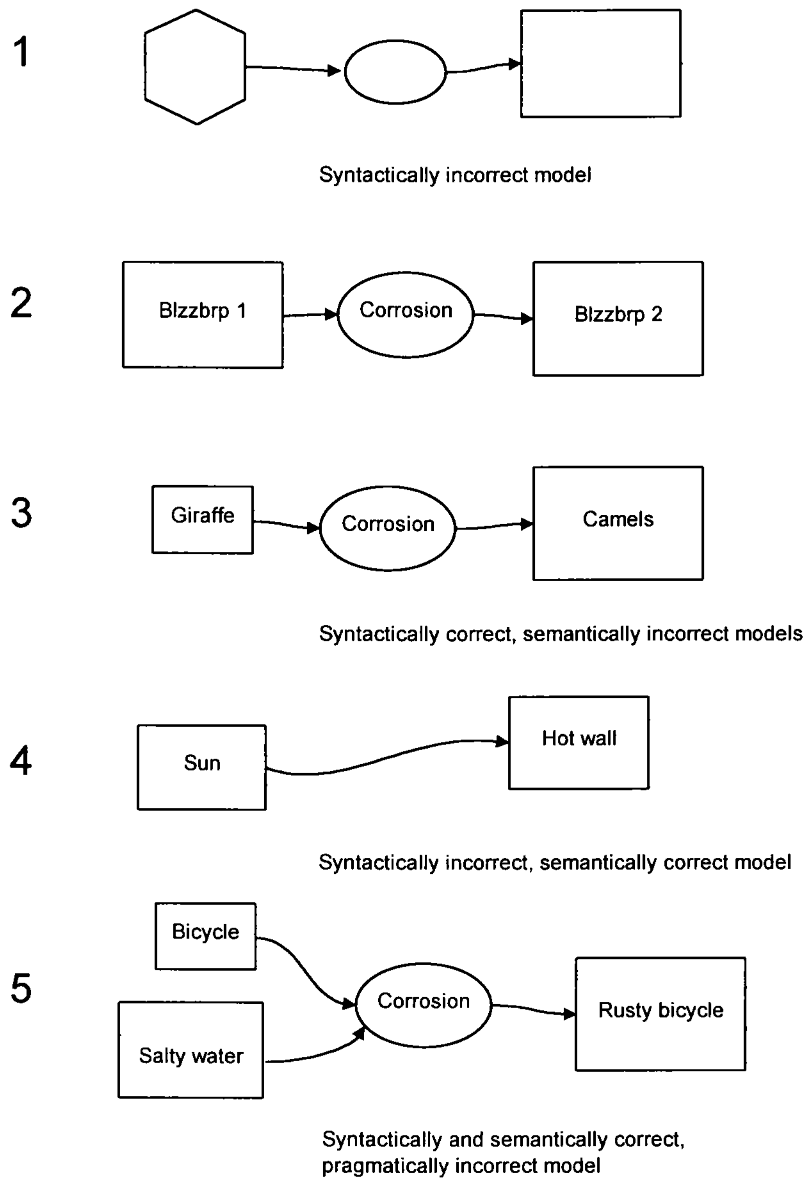


Figure 1: Examples of (in)correctness.

Some further examples will make clear whether an intentional activity is modelled correctly (i.e. whether a model is pragmatically correct). Consider the three models presented in figure 2. All three of them are syntactically correct.

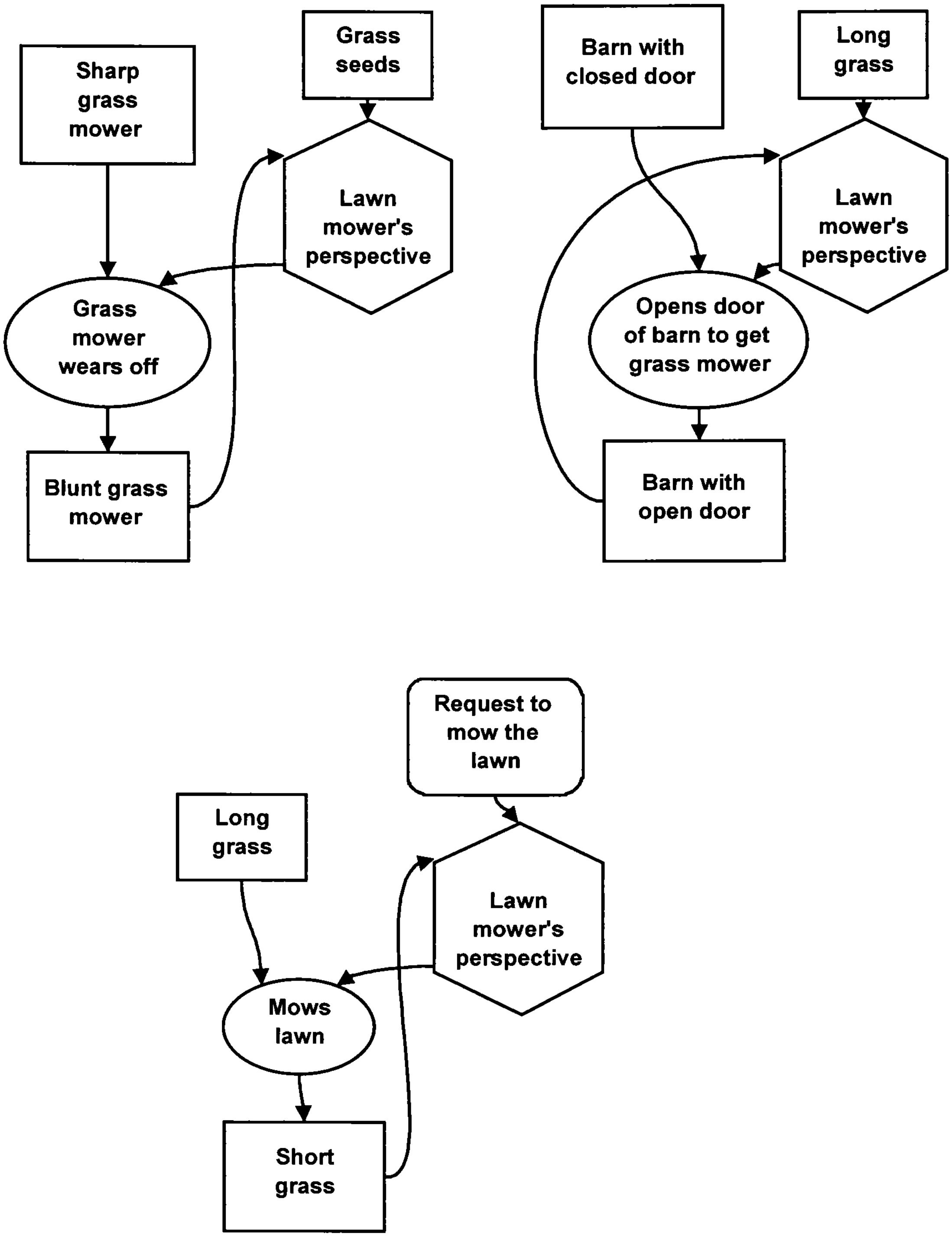


Figure 2: Three alternative "Lawn mower" models.

Example A is *semantically correct*: grass seeds may cause grass to grow, which may cause the "Lawn mower" perspective to become applied, which may cause a lawn mower to mow, which may cause a grass mower (a machine) to wear off. Each of the model primitives can be given a referent, and each of the arrows can be given a causal meaning. However, the model is *not* pragmatically correct: the rectangle "Sharp grass mower", the ellipse "Grass mower wears off" and the rectangle "Blunt grass mower" do not refer to instantiations of the "as is" part, the "script" part and the "to be" part of the "Lawn mower" perspective. In addition, the arrow causing the "Lawn mower" perspective to become applied cannot be interpreted as an observation (this should be an *observation* as the connected reference model primitive is a physical domain state). The model fails pragmatic conventions 1-3.

In example B, applying the "Lawn mower" perspective causes the process "Opens door of barn to get grass mower"; it is semantically correct. However, this process seems to be too specific to be coupled to the "Lawn mower" perspective: it is part of the lawn mower's preparatory actions, rather than a complete implementation of the script. The same holds true for the rectangles "Barn with closed door" and "Barn with open door"; they are not instantiations of the "as is" and "to be" parts of the Lawn mower perspective. The model seems to mix up different systemic levels. The arrow between the rectangle "Long grass" and the hexagon "Lawn mower" in this case can, however, be interpreted as an observation that causes the perspective to become applied. The model fails pragmatic conventions 2 and 3.

Example C is pragmatically correct. It is clear that the environment of the Lawn mower corresponds with its perspective. The intention (which is actually implied by the model, rather than explained) makes sense: the model can indeed be understood as a model of an intentional activity.

APPENDIX C

GENERIC SYSTEMS SPACE

As was mentioned in Research topic 8 (Future research) and the General Discussion (Chapter 11), the *Trinity* methodology can be thought of as being built on a three-dimensional systems concept. This systems concept distinguishes *three* different, yet closely related dimensions: **complexity**, **diversity** and **similarity** (similarity being a more neutral and generic term than adherence). The complexity dimension emphasises that a system, although possibly consisting of many atomic parts, is a *whole*. The diversity dimension enables one to emphasise that, within this overall system, quite different subsystems may be distinguished. Finally, the similarity dimension enables one to emphasise that within this overall system comparable (similar) subsystems may be distinguished.

What is especially interesting about this three-dimensional systems notion, is that in principle a system can be regarded as *an isoplane ($c.d.s = k$, k being a constant $\in N$) in a three-dimensional, KDS-like space* (see also Chapter 3). The system consists of *k* atomic subsystems. **All** the points on this isoplane, obeying the (additional) constraint that c and d and s $\in$ N, are alternative interpretations of the system of concern: interpretations that emphasise different complexity, diversity and similarity aspects of this overall system. Moving *in the plane* implies changing interpretation; *moving from one plane to another* implies changing the very system itself (see also Chapter 3).

Position [k, 1, 1,] emphasises that all k atomic subsystems are part of one whole. Position [1, k, 1] emphasises that all k atomic subsystems are different. Position [1, 1, k] emphasises that all k atomic subsystems are similar in some respect (after all, they all are atomic subsystems). These examples provide extreme interpretations of this system (that in principle can assume all the interpretations on the iso-plane c.d.a=x). Intermediate positions on this plane emphasise that the system consists of s similar and d different parts, their mean complexity being c (c.d.s should equal k). An example will explain this further (see next page):

Example:

Consider a Lego house. It comprises four walls, each of them consisting of 12 blocks, and two roof sides, each of them consisting of 15 blocks.
We consider a block to be atomic (smaller blocks do not exist): they must, therefore, by definition be described as systems [1, 1, 1].
A wall can be described as a system [1, 1, 12]. This interpretation stresses the fact that it consists of 12 similar atomic blocks ($s = 12$).
An alternative interpretation of a wall (as a "whole") might be [12, 1, 1]. This interpretation stresses that it consists of 12 atomic subsystems. We do not know whether they are identical or different.
A third interpretation of a wall might be [1, 12, 1]. This interpretation stresses that twelve different atomic subsystems can be distinguished (indeed all the blocks have a different identity. For example, when you smash one of them, the remaining 11 are not smashed as they do not share the same location).
Note that other interpretations are still possible. For example, we might split the wall in two similar parts, resulting in the interpretation [6, 1, 2]: 2 similar subsystems of complexity 6. The boundary conditions (in agreement with the KDS formulas of Chapter 3) are that $c.d.s=12$ and $c, d, s \in N$.

Likewise, a roof side can be described as a system [1,1,15], emphasising that it consists of 15 identical atomic parts. Alternative descriptions are [15, 1, 1]; [1, 15, 1], and any description that obeys $c.d.s=15$ and $c, d, s \in N$.

The house as a "whole" can be described as a system in many different ways. The boundary conditions are that $c.d.s=78$ (note that the scope changes: now we operate at the iso-plane characterised by $c.d.s = c.d.s_{walls} + c.d.s_{roof} = 78$. We add iso-planes). An example is the interpretation [13, 2, 3]. This interpretation stresses that there are 2 different types of parts (walls and roof sides), the mean number of each type being 3 (4 walls, 2 roof sides), the mean number of atomic blocks per part being 13 (cf. the formulas of Chapter 3). Indeed, $c.d.s=78$.

APPENDIX D

EXPLANATION OF THE NAME "*TRINITY*"

The methodology presented in this dissertation is named "*Trinity*" for three reasons.
First, within the *Trinity* methodology three different domains are distinguished: the *knowledge* domain, the *communication* domain and the *physical* domain. These domains can be distinguished, but (in D-type problem situations) cannot exist without each other in any meaningful manner. When including the "Language of thought", i.e. the language that we use to think, and other (e.g. visual) mental representation "languages" in the communication domain, it can even be argued that every domain is "mirrored" by the two other domains. The three domains therefore constitute a trinity.
Second, a perspective consists of three parts: an *"as is"* part, a *script* part and a *"to be"* part. Only in combination are the parts meaningful (in this case they refer to an *action potential*). The three parts of a perspective therefore constitute a trinity.
Third, KDS distinguishes three axes: *complexity*, *diversity* and *adherence*. These three axes refer to the three fundamental concepts of the systems notion underlying *Trinity* (see also Appendix C): *complexity* emphasises that a system, although consisting of parts, is a whole; *diversity* emphasises that parts, although contributing to the whole, are different in some respects; and *adherence* emphasises that parts, although contributing to the whole, are similar in some other respects. The notion of iso-planes explains that these axes are connected with each other in a subtle manner: different interpretations of one and the same referent imply movements on the iso-plane, and thus imply relative shifts with respect to the three axes. Therefore, the axes constitute a trinity.

REFERENCES AND LITERATURE

1. Aamodt, A. (1990). A Computational Model of Knowledge-Intensive Learning and Problem Solving. In: B.Wielinga et al (eds.). Current Trends in Knowledge Acquisition. Amsterdam: IOS.
2. Allen, R.E., Allen, S.D. (1996). Winnie-de-Poeh en het Oplossen van Problemen, waarin Poeh, Knorretje en hun vrienden ontdekken hoe je problemen oplost met de LOS-OP-Methode. Uitgeverij Sirius en Siderius.
3. Apostel, L. (1960). Towards the formal study of models in the non-formal sciences. Synthese **12**.
4. Association of the Dutch Chemical Industry (1991). Integrated Substance Chain Management.
5. Badiru, A.B., Simin Pulat, P., & Kang, M. (1993). DDM: Decision support system for hierarchical dynamic decision making. Decision Support Systems 10, pp. 1-18.
6. Bain, A. (1871). Remarks made at the Metaphysical Club, Old Cambridge, quoted in Murphy, J.P., 1990.
7. BEA (1995). Vooronderzoek Stratcon bsa - Een eco-tech scan voor meer preventie en hergebruik van bouw- en sloopafval. Bureau voor Economische Argumentatie.
8. Beach, L.R. (1990). Image Theory: Decision Making in Personal and Organizational Contexts, Chichester, England: John Wiley and Sons.
9. Beauvoir, S. de (1954). The Mandarins. Dutch translation. Agathon.
10. Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. IEEE Computer, May, p. 61-72.
11. Buffa, F., Marzano, G., Norese, M.F. (1996). MACRAME: a modelling methodology in multiactor contexts. Decision Support Systems 17, pp. 331-343.
12. Boulding, K.E.(1956). The Image: Knowledge in Life and Society. The University of Michigan Press, Ann Arbor, MI.
13. Breuker, J., van de Velde, W. (1994). CommonKADS library for expertise modelling, Reusable problem solving components. Amsterdam: IOS Press.
14. Bylander, T., Chandrasekaran, B. (1988). Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. Knowledge-based systems, 1, pp. 65-77.
15. Chandrasekaran, B. (1988). Generic tasks as building blocks for knowledge-based systems: the diagnosis and routine design examples. The knowledge engineering review, pp. 183-210.
16. Checkland, P. (1989a). Soft Systems Methodology. In: Rosenhead, J. (1989), pp. 71-101.
17. Checkland, P. (1989 b). An application of soft systems methodology. In: Rosenhead, J. (1989), pp. 101-121.
18. Checkland, P. (1981). Systems Thinking, Systems Practice. Chichester, England: John Wiley & Sons.
19. Checkland, P., & Scholes, J. (1991). Soft systems methodology in action. Chichester, England: John Wiley & Sons.
20. Clancey, W.J. (1985). Heuristic Classification. Artificial Intelligence 27, pp. 289-350.

21.Cornelis, A. (1990). Logica van het gevoel, filosofie van de stabiliteitslagen in de cultuur als nesteling der emoties. Amsterdam: Stichting Essence.
22.Cramer, J. (1991). De illusie voorbij. Op weg naar een brede aanpak van de milieuproblemen. Utrecht: Uitgeverij Jan van Arkel.
23.Cramer, J. (1997). Milieumanagement: van "fit" naar "stretch". Utrecht: Uitgeverij Jan van Arkel.
24.Davis, J.P., Bonnel, R.D. (1990). Producing visually-based knowledge specifications for acquiring organisational knowledge. In: B. Wielinga et al. (eds). Current trends in knowledge acquisition., pp. 105-122. Amsterdam: IOS Press.
25.Dieng, R., Corby, O., Labidi, S. (1994). Agent-Based Knowledge Acquisition. In: A future for knowledge acquisition, Proceedings EKAW '94, pp. 63-82. Berlin: Springer-Verlag.
26.Diepenmaat, H.B., Leupen, M.J. (1991). BAS: het Binnenmilieu Advies Systeem. Concept-rapport.
27.Diepenmaat, H.B. (1993a). A comparison of models in the natural sciences, models in engineering and models of human expertise. TNO report IMW-R 93/117.
28.Diepenmaat, H.B. (1993b). Kennismodellering als een bindend element in de milieuwetenschappen: de milieukennismakelaar. TNO rapport TNO-IMW-P93/008.
29.Diepenmaat, H.B. (1996). *Trinity*: Model-based support for multi-actor environmental problem solving. In: Denzler et al. (eds), Environmental Software Systems. New York: Chapman & Hall Ltd.
30.Diepenmaat, H.B., van Lierop, L., Bruijnes, C. (1997). *Trinity*: modelling methods to support multi-actor problem-solving processes. Proceedings AAAI Spring Symposium Artificial Intelligence in Knowledge Management, March 24-26, Stanford University, Stanford, California, USA.
31.Diepenmaat, H.B., van Lierop, L., Roorda, A. (1997). Een evaluatie van strategische milieutechnologische conferenties vanuit een *Trinity* optiek. TNO rapport.
32.Diepenmaat, H.B. (ed), Bruijnes, C., Derksen, L., Heslinga, D., Krozer, J., van Lierop, L., Weenk, A. (1997). Knelpuntenanalyse KWS2000. Deel A: Beleidsnotitie en deel B: Achtergronddocumenten. Publicatiereeks KWS2000, Ministerie van VROM. (Also published as TNO-rapport TNO-MEP - R 97/044).
33.Diepenmaat, H.B. (1997). Handleiding *Trinity*, Modelmatige ondersteuning bij milieubeleidsprocessen (wat is het en hoe gebruik ik het). TNO-rapport TNO-MEP - R 97/123.
34.Dreyfus, H.L., Dreyfus, S.E. (1986). Mind Over Machine, The power of Human Intuition and Expertise in the Era of the Computer, Chapter 1: Five steps from Novice to Expert. Basil Blackwell Limited.
35.Eden, C. (1989). Using cognitive mapping for strategic options development and analysis (SODA). In: Rosenhead, J. (1989), pp. 21-43.
36.Eden, C. (1989). SODA and cognitive mapping in practice. In: Rosenhead, J. (1989), pp. 43-71.
37.Fridsma, D.B., Gennari, J., Musen, M. (1997). Acquiring, Maintaining, and Customizing Organizational Work Process Descriptions. Proceedings AAAI Spring Symposium Artificial Intelligence in Knowledge Management, March 24-26, Stanford University, Stanford, California, USA.
38.Geurts and Vennix (1989), Verkenning in beleidsanalyse, theorie en praktijk van de modelbouw en simulatie. Kerckebosch BV, Zeist.
39.Glasbergen, P. (1994). Milieubeleid, een beleidswetenschappelijke inleiding. s'-Gravenhage: VUGA.

40.Gossling, W. (1960). The Design of Engineering Systems. London: Heywood.
41.Groot, W. T. de (1992). Environmental Science Theory, Concepts and Models in a One-World, Problem-Oriented Paradigm. Amsterdam: Elsevier.
42.Heidegger, M. (1927). Being and Time. Translation by John Macquarrie and Edward Robinson, Harper, San Francisco, 1962.
43.Heidegger, M. (1924), The concept of time. German-English version, translated by William McNeill. Blackwell Press, Oxford UK and Cambridge USA, 1992.
44.Heijungs, R. (ed.) (1992a). Environmental life cycle analysis of products: backgrounds. In Dutch. Leiden, Netherlands: Centre for Environmental Science.
45.Heijungs, R. (ed.) (1992b). Environmental life cycle analysis of products: manual. In Dutch. Leiden, Netherlands: Centre for Environmental Science.
46.Hendriks, M.M.W.B., de Boer, J.H., Smilde, A.K., Doornbos, D.A. (1992). Multicriteria decision making. Chemometrics and Intelligent Laboratory Systems, 16, pp. 175-191.
47.Hofman, M.E. (1994). Cognitive mapping methods: an explorative study. General Social Sciences, University of Utrecht, the Netherlands.
48.Holland, J.H., Holyoak, K.J., Nisbett, R.E. and Thagard, P.R. (1986). Induction, processes of inference, learning and discovery. Cambridge, Massachusetts: The MIT Press.
49.Hulthage, I. (1988). Quantitative and qualitative models in artificial intelligence. In: J.S. Kowalik and C.T. Kitzmiller (eds). Coupling symbolic and numerical computing in expert systems, II. Elsevier Science Publishers B.V. (North Holland).
50.Kämpke, T., Radermacher, F.J., Wolf, P. (1993). Supporting preference elicitation. Decision Support Systems 9, pp. 381-391.
51.Kangassalo, H. (1990). Foundations of Conceptual Modelling: A Theory Construction View. In Information Modelling and Knowledge Bases (H. Kangassalo, S. Ohsuga, H. Jaakkola, ed.), pp. 19-36. Amsterdam: IOS Press.
52.Karbach, W., Linster, M., Voss, A. (1990). Models of problem solving: one label - one idea?. In: B. Wielinga et al (eds). Current trends in knowledge acquisition, pp. 173-189. Amsterdam: IOS Press.
53.Kok, R. (1992). Een generiek model voor het oplossen van binnenmilieuproblemen. TNO-rapport Instituut voor Milieuwetenschappen, Delft.
54.Kolb, D. (1984). Experiental Learning: Experience as the Source of Learning and Development. New Jersey: Prentice-Hall.
55.Laske, O. (1990). Several Seminars on Knowledge Management.
56.Laudan, L. (1990). Science and Relativism. Some key controversies in the philosophy of science. The University of Chicago Press.
57.Laudan, L. (1996). Beyond Positivism and Relativism: theory, method and evidence. Westview Press.
58.van Lierop, L. (1997). Positionering van *Trinity* ten opzichte van andere methodologieën. To be published as TNO-report.
59.Machiavelli, N. (1513). De Heerser (Il Principe). Uitgave Atheanaeum-Polak & van Gennep, 1995.
60.McCarthy , J. and Hayes, P.J. (1969). Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: Meltzer, B., Michie, D., and Swann, M. (eds). Machine Intelligence **4**, pp. 463-502.
61.McGraw, K.L., Seale, M.R. (1988). Knowledge elicitation with multiple experts: considerations and techniques. Artifical Intelligence Review **2**, pp. 31-44.
62.Meehan, E.J. (1993). The Thinking Game. Chapter 5: Description. Chatham House Publishers.

63.Meadows, D.H., Meadoms, D.L., Randers J. (1991). Beyond the Limits. Confronting Global Collapse; Envisioning a Sustainable Future. London: Earth Scan Publications.
64.Millward, R.B. (1980). Models of Concept Formation. In Aptitude, Learning and Instruction Volume 2: Cognitive Process Analyses of Learning and Problem Solving (R.E. Snow, P. Federico, W.E. Montague eds.). Hillsdale New Jersey: Lawrence Erlbaum Associates.
65.Minsky, M. (1975). A framework for representing knowledge. In P.H. Winston (ed.). The Psychology of Computer Vision. New York: McGraw-Hill.
66.Minsky, M. (1985). The society of mind. New York: Simon and Schuster.
67.Mintzberg, H. (1990). Strategievorming: tien scholen (Strategy formation: schools of thought). Scriptum Management, the Netherlands.
68.Mittelstrass, J. (1993). Unity and Transdisciplinarity. Interdisciplinary Science Reviews **18**, pp. 153-157.
69.Murphy, J.P. (1990). Pragmatism from Peirce to Davidson. Boulder: Westview press.
70.Newell, A. (1982). The Knowledge Level, Artificial Intelligence, 18:1 (Jan. 1982), pp.87-127.
71.Newell, A. and Simon. H.A. (1972). Human Problem Solving. New York: Prentice Hall.
72.Quakernaat, J. (1991). De scenariovrije verkenning, hulpmiddel bij het verkennen van technologische ontwikkelingen. Intern TNO-rapport IMET-M&E.
73.Quakernaat, J. (1995). Milieutoekomstverkenningen, ontwerp voor een aanpak, met een accent op back-casting. Intern TNO-rapport ME-MiT.
74.Reinders, M. (1991). Reflective Strategic Control of Diagnosis. Technical Report of the Netherlands Energy Research Foundation (ECN), Petten, The Netherlands.
75.Riviere, J.M.W. la (1991). Co-operation between natural and social sciences in global change research: imperatives, realities, opportunities. International Social Science Journal **130**, 691-626.
76.Roorda, A., Wiersma, T., de Zeeuw, J. and Diepenmaat, H.B. (1997). Een contextanalyse van Infomil met behulp van *Trinity*. TNO-rapport 1997.
77.Rorty, R. (1988). Solidarität oder Objektivität?: drei Philosophische Essays. Stuttgart: Reclam.
78.Rosenhead, J. (1989). Rational analysis for a problematic world, problem structuring methods for complexity, uncertainty and conflict. Chichester, England: John Wiley and Sons.
79.Safranski, R. (1994), Heidegger. Dutch translation by M. Wildschut. Amsterdam: Atlas Publishers.
80.Schank, R. and Abelsen, R.P. (1977). Scripts, plans, goals, and understanding: an inquiry into human knowledge structures. Hillsdale, New Jersey: Lawrence Erlbaum.
81.Schön, D.A. (1983). The reflective practitioner, how professionals think in action. Basic Books, Harpers Collins Publishers.
82.Searle, J.R. (1984). Mind, brains and science. Harvard University Press, Cambridge, Massachusetts
83.Second National Environmental Policy Plan (1993-1994). The Hague: Dutch State Press.
84.Senge, P., Kleiner, A., Roberts, C., Ross, R., Smith, B. (1994). The Fifth Discipline Fieldbook. Strategies and tools for building a learning organisation. New York: Currency Doubleday.
85.Simon, H. (1957). Models of Man. New York: Wiley.
86.Simon, H. (1976). Administrative behaviour. New York: The Free Press.
87.Singh, M.G., Bennavail, J.C., Chen, Z.J. (1992). A group decision tool for combining subjective estimates based on an optimisation approach. Decision Support Systems 8, pp. 541-549.

88.Shaw, M.J., Fox, M.S. (1993). Distributed artificial intelligence for group decision support. Integration of problem solving, coordination and learning. Decision Support Systems 9, pp. 349-367.
89.Sowa, J.F. (1984). Conceptual Structures. Information Processing in Mind and Machine. Addison-Wesely.
90.Spijkervet, A., van der Spek, R. (1996). Knowledge Management. Uitgave Kenniscentrum CIBIT, Utrecht.
91.Steels, L., Schreiber, G., van de Velde, W., Proceedings 8th European Knowledge Acquisition Workshop. Springer Verlag.
92.Tolido, R.J.H. (1995). Inleiding IAD (Iterative Application Development). Het evolutionair ontwikkelen van informatiesystemen.
93.Vennix, J.A.M. (1996). Group Model Building, facilitating team learning using System Dynamics. Chichester, England. John Wiley and Sons.
94.Vermeulen, M., Diepenmaat, H.B. (1993). *Trinity*, Een evaluatie van een modelleerwijze gericht op complexe milieuproblemen, aan de hand van de pesticidenproblematiek in de fruitteelt. TNO-rapport IMW-R 93/168.
95.Weenk, A., Annokkée, G.J., van Veen, H.J. (1993). Een gestructureerde benadering voor het prioriteren van milieumaatregelen (STEPS). TNO-rapport 93-296.
96.Weick, K.E., Bougon, M.G. (1986). Organizations as Cognitive Maps: Charting Ways to Success and Failure. In: H. Simms, A. Goia and Associates, The Thinking Organization. San Francisco: Jossey-Bass Publishers.
97.Wielinga, B.J., Boose, J., Gaines, B., Schreiber, G., van Someren, M. (1990). Current trends in knowledge acquisition. Amsterdam: IOS press.
98.Wielinga, B.J., Schreiber, A.T., de Greef, P. (1989). Synthesis report. Esprit Project P1098, Deliverable Y3, University of Amsterdam.
99.Wiig, K.M. (1993). Knowledge Management Foundations: thinking about thinking, how people and organizations create, represent, and use knowledge. Arlington, Texas. Schema Press.
100.Winograd, T., Flores, F. (1986). Understanding Computers and Cognition, a new foundation for design. Reading, Massachusetts. Addison-Wesley Publishing Company, Inc.
101.Zeigler, B.P. (1990). Object-Oriented Simulation with Hierarchical, Modular Models, intelligent agents and endomorphic systems. Academic Press, Inc.
102.Zeigler, B.P. (1984). Multifacetted Modelling and Discrete Event Simulation. Academic Press.

SUMMARY

Contemporary society is complex. Its central activities (like production and consumption processes) consist of many different partial activities, that depend upon each other to a degree that was never observed before. As a result, in many of the problems that our society is facing, many different parties are involved: the problems are *multi-actor*.
In order to improve multi-actor problem situations, different parties have to make complex decisions of their own. Simultaneously with being beneficial for these actors in isolation, these decisions should further the overall improvement process. We consider a coherent understanding of the problem context as a whole to be an important prerequisite for successful intervention (i.e. intervention resulting in improvement). In multi-actor situations, however, the problem context is typically highly complex and only partially understood. Bearing in mind that the success of individual actions highly depends upon the actions of other actors participating in the problem context, it is not surprising that multi-actor problems again and again prove to be extremely difficult to solve. Environmental problems provide manifest examples of this.
For this reason, we focus in this dissertation on developing a methodology that supports the process of obtaining a coherent understanding of multi-actor problem contexts: an understanding that both motivates and guides interventions directed at improvement. The name of this methodology is *Trinity*.

This dissertation focuses on the design of *modelling methods*. In this context, the central research question is:

> *Is it possible to design modelling methods that specifically support problem-solving processes in multi-actor situations?*

The design of such methods is situated in an overall methodological research approach that encompasses the elaboration of *four* layers:

1. a *philosophy layer* (addressing the most general principles and assumptions underlying the methodology);
2. a *theory layer* (fleshing out the *philosophy* layer and providing a conceptual framework that justifies and positions the methods);
3. a *methods layer* (providing the conceptual toolbox of the methodology); and
4. a *use layer* (encompassing the use of the methodology in practice).

The *philosophy* and *theory layers* provide a foundation for the methods, and the *use layer* is the arena where the support and added value of using these methods is to be established. Each of the four layers is described in depth in separate parts of this dissertation.

The **Philosophical background** part addresses the central concept of the *Trinity* methodology: *problem solving*. This concept is provided with a philosophical basis and a specific meaning.
First, several interpretations of problem solving are discussed. By means of a generic model of intentional activities, it can be shown that these interpretations emphasise different aspects of one and the same scheme. On the basis of this scheme we present an interpretation of problem solving that is used consistently throughout this dissertation. According to this interpretation, problem solving is the process of obtaining a coherent overview of a problem context, an overview that both motivates and guides intentional action. Such an overview we call a *perspective*. Therefore, *problem solving is perspective construction.*
Second, several aspects of the generic model of intentional activities are discussed. Potential critiques are addressed and the philosophical stance underlying the model is presented.
Finally, on the basis of the results so far, an operational definition of the notion of model-based support for problem solving is developed. It is argued that *modelling* is a tool that matches many of the needs of a multi-actor problem solver. This makes it possible to interpret problem solving as an activity that can be *supported by* an explicit modelling process, and that *results in* a (qualitative) model of a perspective.

In the **Theory** part the remaining two central notions, *multi-actor* and *modelling*, are elaborated.

First, *multi-actor situations* are addressed. In multi-actor situations, the knowledge distribution (i.e. the way in which knowledge of relevance is distributed over actors) is typically rather complex. In addition, the relation between knowledge distribution and problem solving is rather complex. Two questions are addressed: is it possible to classify problem situations based on the way in which relevant knowledge is distributed over individuals, and (if so): does this provide methodological guidelines for the subsequent problem-solving process?
In order to answer these questions, an abstract theory of *knowledge distributions* is developed. Within this theory, the notions of multi-actor situations and multi-actor processes are defined in a systemic manner. The basis of this theory is *Knowledge Distribution Space*: a conceptual framework that distinguishes problem situations on the basis of *three* attributes: the complexity of points of view, the number of different points of view and the number of individuals, adhering to points of view. A qualitative interpretation of this space results in a problem typology based on knowledge distribution, which answers the first question. A methodological interpretation of the framework shows that each of the problem types exhibits specific problem-solving bottlenecks, and

consequently requires different problem solving principles and problem solvers. This permits us to answer the second question.
In many problem situations, both the number of different points of view and the complexity of these points of view are large. This problem type is called the *D-type*. As a typical D-type problem solver the knowledge broker is introduced: a mixture of a domain knowledge generalist and a knowledge management specialist, whose focus is primarily on obtaining an understanding of the points of view of actors, involved in the problem context of concern. At this moment, methodological support for knowledge brokerage activities is scarce. Providing this support is an important research activity (and the development of *Trinity* is an attempt to provide such support).

Second, *qualitative modelling processes* are investigated from a generic point of view. The concept of a model relation is described, relating a model to a referent. After that, four different types of model relations are distinguished. They are used to derive a typology of primitive modelling steps: steps that enable one to convert an initial model relation into a new one. The introduction of primitive modelling steps enables one to define a modelling process as a sequence of these steps. A modelling strategy now can be defined as a typical (recurrent) sequence of modelling steps. Several families of modelling strategies are distinguished and explained, each of them offering a different mix of modelling steps, especially suited to meet a specific goal.
The notions of model relations, modelling steps, and modelling strategies offer a conceptual vocabulary to analyse, synthesise and discuss qualitative modelling processes. Moreover, in combination they constitute a theory that provides guidelines for designing qualitative modelling approaches. The theory will be used in the Methods part (see below) to develop modelling methods that are dedicated to supporting multi-actor problem solving.

The Philosophical background and Theory parts provide together a thorough basis for the **Methods** part of this dissertation, which is devoted to the development of modelling methods that support D-type problem-solving processes. The central element of these methods is the *Trinity qualitative modelling language*: a flexible scheme convention, designed in full compliance with the Philosophy and Theory parts of this dissertation. The *Trinity* modelling language is explained in detail, and several of its modelling strategies are explained.
In the *methods layer*, *Trinity* turns out to be a methodical framework, rather than a single method. Several different ways of using *Trinity* (modes) are distinguished, enabling one to support quite different D-type problem-solving processes. Examples are *trouble-shooting* mode (directed at taking away a rather confronting situation of discontent) and *back-casting* mode (facilitating a rather explorative problem-solving process by means of emphasising potential futures first). In addition, a practical guideline to use *Trinity* in real-world multi-actor problem-solving processes is provided.

The availability of these methods enables us to turn our attention to the second part of the central research question, dealing with *support*. This topic is addressed in the **Experiments** part of the dissertation. In order to investigate *Trinity*'s use and added value in practice, three quite different experiments in multi-actor problem solving were conducted, that took place in environmental problem contexts.
In the *Indoor environmental problems* experiment, a rather diagnostic use of *Trinity* is presented: the emphasis is on a model describing problematic indoor environments "as is". The problems of concern are characterised by their confronting nature and a rather short time scale for remedial action. *Trinity* models proved to be valuable in obtaining a coherent understanding of problem situations. In addition, the resulting models support diagnostic processes, as well as the preparation of remedial actions.
In the *VOC2000* experiment, the emphasis is on the first steps in an attempt to improve the VOC2000 programme: a national (Dutch) environmental policy process, based on an agreement between the Dutch government and the corporate sector, to reduce the emission of volatile organic compounds (VOCs) considerably. In some sectors, this process is slowing down. The problems of concern are characterised by a time scale of one year and up. In this experiment, the use of *Trinity* proves to be supportive in obtaining a coherent understanding of the VOC2000 process as a whole. In addition, the resulting model is of help in establishing the contours of future policies directed at improvement.
Finally, in the *Building and demolition waste* experiment, a strategic conference is reviewed from a *Trinity* point of view. The conference was directed at establishing agreement about future strategies in this problem area, and starting up concerted actions. The problem of concern is characterised by a time scale of several decades. Emphasis is on a future situation. Several aspects of this future are clarified by means of the construction of a "to be" model. In addition, implications for actions that intend to realise this "to be" are contrasted with the actions actually agreed upon at the conference. Here also *Trinity* proved to be supportive in obtaining a clear understanding of the multi-actor situation of concern.

Although the three experiments were quite different in several respects, in all three cases the use of *Trinity* offered considerable added value. Multi-actor situations became better understood. The design and thinking through of potential interventions in and improvements of multi-actor networks were supported by the models. In addition, *Trinity* guided and supported the communication processes with informers and field players during the problem-solving process. The exact relation between D-type problem characteristics and specific modes of using *Trinity*, however, requires further elaboration. Notwithstanding this, on the basis of the three experiments we conclude that *Trinity* offers support in multi-actor problem solving.

Finally, in the **General discussion and conclusions** part of this dissertation, several aspects of the *Trinity* methodology as a whole are discussed.
In the general discussion, first a concise review of the fundaments of the *Trinity* methodology is presented. On the one hand, *Trinity* is built upon strong empirical

generalisations. On the other hand, *Trinity* is founded upon a systemic design. Therefore, the fundaments of *Trinity* comply with both a top-down (theory → practice) as well as a bottom-up (practice → theories) point of view.

Second, several key features of the methodology are re-iterated and discussed. Notably: its precise philosophical definition of problems and problem solving; the distinction of both intentional and autonomous activities; its distinction of a physical, a knowledge, and a communication domain as part of an integral model; its systemic nature; its emphasis on pragmatic ways to improve situations of concern; its domain-independence (it supports multi-actor problem solving in general); and the fact that it uses (only) *one* modelling language to refer to both situations ("as is" and "to be") as well as the improvement processes relating them.

Third, the "backbone" is exposed: a theory that ties together the different parts of this dissertation into one systemic background framework.

Fourth, *Trinity* is positioned in the field of some mainstream paradigms in dealing with complexity. Specifically the agreements and differences with Systems Dynamics are discussed. Whereas Systems Dynamics is a quantitative and continuous approach, *Trinity* is a qualitative and discrete approach. This is a direct result of its emphasis on actors and intentional activities: these notions can be easily expressed in a qualitative, discrete paradigm.

Finally, the added value of using *Trinity* is discussed from a methodological point of view. This added value *manifests itself* in the *use layer* (see the Experiments part), but *follows from Trinity*'s layered methodological design as a whole. Especially the *philosophy layer*, providing persons engaged in D-type problem solving with both a goal (a model of a multi-actor perspective) as well as a means to attain this goal (modelling), and the *methods layer*, providing a rich conceptual toolbox, contribute to this support.

Our final conclusion is that we succeeded in designing modelling methods that specifically support problem-solving processes in multi-actor situations. *Trinity* is the case in point: *Trinity* offers model-based support for multi-actor problem solving.

It is general practice that D-type interventions and improvement processes are performed on the basis of a rather fragmentary and intuitive understanding of the problem context of concern. In order to prevent this, we strongly recommend the use of *Trinity* in D-type problem solving. The dissertation, therefore, ends with several routes to further the diffusion of its use.

SAMENVATTING

Onze samenleving is complex. Haar centrale activiteiten (zoals produktie- en consumptieketens) bestaan uit een verscheidenheid aan deelprocessen die, vanuit een historisch perspectief, op een nog nooit eerder vertoonde wijze van elkaar afhankelijk zijn. Als resultaat hiervan worden de problemen, waarvoor onze maatschappij zich gesteld ziet, gekarakteriseerd door de betrokkenheid van vele spelers: de problemen zijn *multi-actor*.
Om multi-actor probleemsituaties te verbeteren, moeten verschillende spelers complexe besluiten nemen. Deze besluiten dienen in eerste instantie doorgaans de eigen belangen. Daarenboven moeten deze besluiten *in samenhang* leiden tot een verbetering van de *totale* probleemcontext (dus inclusief de wensen en belangen van andere betrokken actoren).
Wij beschouwen het bezit van een *samenhangend* overzicht van de *gehele* probleemcontext als een belangrijke randvoorwaarde om doelgericht in te kunnen grijpen. Juist in multi-actor situaties is de probleemcontext echter in hoge mate complex, en doorgaans slechts gedeeltelijk begrepen. Mede gezien het feit dat het succes van individuele acties in hoge mate afhankelijk is van de acties van andere spelers, is het niet opzienbarend dat multi-actor problemen keer op keer zo moeilijk oplosbaar blijken te zijn.
Om deze reden richten we ons in dit proefschrift op het ontwikkelen van een methodologie, die behulpzaam is bij het verkrijgen van een samenhangend beeld van multi-actor probleemcontexten; een beeld dat doelgerichte interventie zowel motiveert als ondersteunt. De naam van deze methodologie is *Trinity*.

Het onderzoek richt zich op modelleermethoden. De centrale onderzoeksvraag is:

> *Is het mogelijk om modelleermethoden te ontwerpen, die specifiek ondersteuning bieden bij probleemoplosprocessen in multi-actor situaties?*

Het ontwerp van dergelijke methoden is geplaatst in een methodologische onderzoeksbenadering die de uitwerking van vier "lagen" omvat:

- een *filosofielaag* (die de meest fundamentele principes en assumpties van de methodologie beschrijft);
- een *theorielaag* (die enerzijds de filosofielaag verder uitwerkt, en anderzijds een raamwerk biedt voor het onderbouwen en positioneren van de methoden);
- een *methodenlaag* (die voorziet in de methoden, de conceptuele "gereedschapskist" van de methodologie); en
- een *toepassingslaag* (die het gebruik van de methodologie in de praktijk omvat).

De *filosofie-* en *theorielaag* voorzien in een basis voor de methoden, en de *toepassingslaag* is de arena waar de ondersteuning en toegevoegde waarde van deze methoden moet blijken. Ieder van de vier lagen wordt in een apart deel van dit proefschrift beschreven.

Het deel **Filosofische achtergronden** gaat met name in op het centrale concept van de *Trinity* methodologie: probleemoplossen. Dit concept wordt voorzien van een filosofische basis en een specifieke betekenis.
Ten eerste worden verschillende interpretaties van het begrip probleemoplossen aan een discussie onderworpen. Door middel van een *generiek model van intentionele activiteiten* wordt aangetoond dat deze interpretaties verschillende aspecten benadrukken van één achterliggend schema. Gebruik makend van dit schema presenteren we een additionele interpretatie van probleemoplossen: een interpretatie die aan de basis staat van de *Trinity* methodologie en die consistent gebruikt wordt in dit proefschrift. Volgens deze interpretatie is probleemoplossen het verkrijgen van een samenhangend beeld van een probleemcontext, een beeld dat intentioneel handelen motiveert en ondersteunt. Een dergelijk beeld noemen we een *perspectief*: *probleemoplossen is perspectief-constructie*.
Ten tweede worden verschillende aspecten van het generieke model van intentionele activiteiten aan een diepgaande discussie onderworpen. Potentiële kritieken ten aanzien van dit model worden beantwoord en de filosofische positie die eraan ten grondslag ligt wordt gepresenteerd.
Tenslotte wordt, op basis van de resultaten tot dusverre, een operationele definitie ontwikkeld van het begrip *modelgebaseerde ondersteuning van probleemoplossen*. Er wordt gesteld dat het instrument modelleren invulling geeft aan veel behoeften van een multi-actor probleemoplosser. Dit maakt het mogelijk probleemoplossen te interpreteren als een activiteit die *ondersteund* kan worden door middel van een expliciet modelleerproces, en die *resulteert* in een (kwalitatief) model van een perspectief.

In het **Theorie** deel van dit proefschrift worden de andere twee centrale noties, *multi-actor* en *modelleren*, uitgewerkt.
Eerst wordt het begrip *multi-actor situatie* uitgewerkt. In multi-actor situaties is de kennisdistributie (de wijze waarop relevante kennis is gedistribueerd over actoren) doorgaans complex. Twee vragen staan centraal: is het mogelijk om probleemsituaties te classificeren op basis van de wijze waarop relevante kennis is gedistribueerd over individuen, en (zo ja): levert dit methodologische richtlijnen op voor het bijbehorende probleemoplosproces?
Om deze vragen te kunnen beantwoorden, is een abstracte theorie van *kennisdistributies* ontwikkeld. Binnen het kader van deze theorie zijn de noties van multi-actor situaties en multi-actor processen op een systemische wijze gedefinieerd. De basis van de theorie is *KennisDistributieRuimte*: een conceptueel raamwerk dat probleemsituaties onderscheidt op basis van drie attributen: de complexiteit van standpunten, het aantal verschillende standpunten, en het aantal individuen dat deze standpunten aanhangt. Een kwalitatieve interpretatie van deze ruimte resulteert in een typologie van problemen, gebaseerd op

kennisdistributie. Dit beantwoordt de eerste vraag. Een methodologische interpretatie van de ruimte toont aan dat ieder van de onderkende probleemtypen gekarakteriseerd wordt door specifieke moeilijkheden, en dus andere probleemoplosprincipes en probleemoplossers vereist. Dit is de basis voor beantwoording van de tweede vraag.
In vele probleemsituaties is zowel het aantal verschillende standpunten als de complexiteit van deze standpunten hoog. Dit probleemtype wordt het D-type genoemd. Als een specifieke D-type probleemoplosser wordt de kennismakelaar geïntroduceerd: een mengvorm van een domeinkennis-generalist en een kennismanagement-specialist, die voornamelijk gericht is op het doorgronden van de standpunten van actoren die een rol spelen in de probleemcontext. Op dit moment is methodologische ondersteuning voor de activiteiten van een kennismakelaar een schaars goed. Voorzien in dergelijke ondersteuning is een belangrijk onderzoeksthema (en het ontwerpen van *Trinity* is een poging om in dergelijke steun te voorzien).

Ten tweede worden kwalitatieve modelleerprocessen onderzocht vanuit een generiek standpunt. Het concept modelrelatie wordt beschreven, dat de relatie legt tussen een model en zijn referent (onderwerp). Vier verschillende modelrelaties worden onderscheiden. Zij worden gebruikt om een typologie van primitieve modelleerstappen af te leiden: stappen die het mogelijk maken om een initiële modelrelatie om te zetten in een nieuwe modelrelatie. De introductie van primitieve modelleerstappen maakt het mogelijk om modelleerprocessen te beschrijven als een reeks van dergelijke stappen. Een modelleerstrategie kan nu gedefinieerd worden als een typische (vaker voorkomende) sequentie van modelleerstappen. Verschillende families van modelleerstrategieën worden onderscheiden en toegelicht. Ieder van deze families biedt een speciale "mix" van modelleerstappen, speciaal geschikt om specifieke modelleerdoelen te bewerkstelligen.
De noties modelrelaties, modelleerstappen en modelleerstrategieën bieden een conceptueel vocabulaire om kwalitatieve modelleerprocessen te kunnen analyseren, samen te stellen en te bediscussiëren. Bovendien vormen zij in combinatie een theorie die richtlijnen geeft voor het ontwerpen van kwalitatieve modelleerbenaderingen. Deze theorie zal in het Methoden deel van dit proefschrift (zie hieronder) gebruikt worden om modelleermethoden te ontwikkelen, die specifiek zijn toegesneden op het ondersteunen van multi-actor probleemoplosprocessen.

In samenhang bieden het Filosofische achtergronden deel en het Theorie deel een grondige basis voor het **Methoden** deel van dit proefschrift, dat gericht is op het ontwikkelen van methoden die D-type probleemoplosprocessen ondersteunen. Het hart van deze methoden is de *Trinity* kwalitatieve modelleertaal: een flexibele schemaconventie, die volledig in overeenstemming met het Filosofie deel en het Theorie deel (zie hierboven) is ontworpen. De *Trinity* modelleertaal wordt in detail toegelicht, en verschillende van haar modelleerstrategieën worden uitgelegd.
Trinity blijkt een methodisch raamwerk, in plaats van één enkele methode te zijn. Verschillende manieren worden onderscheiden om dit raamwerk toe te passen. Deze verscheidenheid maakt het mogelijk sterk verschillende D-type probleemoplosprocessen te

kunnen ondersteunen. Voorbeelden van verschillende manieren van toepassing zijn de "*trouble-shooting*" werkwijze, gericht op het wegnemen van de oorzaak van een probleem, en de "*back-casting*" werkwijze, die een sterk explorerend probleemoplosproces bevordert door middel van het reeds in een vroeg stadium van het probleemoplosproces benadrukken van de verkenning van mogelijke toekomsten.
Tevens wordt voorzien in een praktische handleiding om *Trinity* in multi-actor probleemsituaties toe te kunnen passen.

Het voorhanden zijn van deze modelleermethoden maakt het ons mogelijk de aandacht te richten op het tweede deel van de centrale onderzoeksvraag, betreffende het *bieden van ondersteuning*. Dit aspect wordt beschreven in het **Experimenten** deel van dit proefschrift. Om het gebruik van *Trinity*, en meer specifiek de toegevoegde waarde hiervan in de praktijk, te onderzoeken zijn drie verschillende experimenten in multi-actor probleemoplossen verricht. Alle drie de experimenten vonden plaats binnen de setting van een milieuprobleem.
In het *Binnenmilieuproblemen* experiment wordt een diagnostisch gebruik van *Trinity* beschreven. De nadruk ligt op een generiek "as is" model dat problematische binnenmilieus beschrijft. De problemen kunnen worden gekarakteriseerd door hun confronterende karakter en de vrij korte tijdschaal die beschikbaar is voor het vinden van een remedie. De *Trinity* modellen bleken waardevol in het verkrijgen van een samenhangend begrip van probleemsituaties. Tevens ondersteunen de modellen de diagnose en het opstellen van een remedie voor specifieke binnenmilieuproblemen.
In het *KWS2000* experiment ligt de nadruk op de eerste stappen in een poging om het verloop van het *KWS2000* programma te ondersteunen. Het *KWS2000* programma is een nationaal milieubeleidsproces, dat gebaseerd is op een afspraak tussen de nationale overheid en het bedrijfsleven om de emissie van vluchtige organische stoffen (KoolWaterStoffen) aanzienlijk te reduceren. Hoewel het programma grotendeels succesvol is, verloopt dit proces in sommige industriële sectoren niet helemaal naar wens. Bij dit experiment geldt een tijdschaal van één tot enkele jaren. In dit experiment bleek het gebruik van *Trinity* ondersteunend te werken bij het verkrijgen van een samenhangend overzicht van de uit de *KWS2000* afspraken resulterende activiteiten in zijn geheel. Verder bleek het vervaardigde model behulpzaam bij het vaststellen van de contouren van toekomstige aanvullende beleidsprocessen gericht op verbetering.
Tenslotte ligt in het *Bouw- en sloopafval* experiment het accent op een evaluatie van een strategische conferentie. Deze conferentie was gericht op het vaststellen van toekomstige strategieën in dit probleemveld, en het starten van de benodigde acties om deze strategieën te realiseren. Het probleem wordt gekarakteriseerd door een tijdschaal van verschillende decennia. De nadruk ligt op de toekomstige situatie; Verschillende aspecten van deze toekomstige situatie zijn verhelderd middels een *Trinity* "to be" model. Verder zijn de acties, waarover tijdens de conferentie consensus bestond, gecontrasteerd met de implicaties voor acties die volgen uit dit "to be" model. Dit heeft geleid tot de vaststelling van een aantal discrepanties. Ook in dit experiment bleek *Trinity* in hoge mate behulpzaam bij het verkrijgen van een coherent beeld van de multi-actor problematiek.

Hoewel de experimenten op belangrijke punten van elkaar verschilden, bleek de toepassing van *Trinity* in alle drie de gevallen aanzienlijke en belangrijke additionele inzichten op te leveren. Multi-actor situaties werden beter begrepen. Het ontwerpen en doordenken van potentiële interventies in en verbetering van multi-actor netwerken werd ondersteund door (het maken van) modellen. Verder richtte en ondersteunde het gebruik van *Trinity* de kennisacquisitieprocessen (inclusief de communicatie met informanten en veldspelers) die tijdens het probleemoplosproces verricht werden. De precieze relatie tussen karakteristieken van D-type problemen en specifieke gebruikswijzen van *Trinity* vereist nog aanvullend onderzoek. Desondanks concluderen wij, op basis van bovengenoemde experimenten, dat *Trinity* ondersteuning biedt bij multi-actor probleemoplosprocessen.

Tenslotte worden in het **Algemene discussie en conclusies** deel van dit proefschrift een aantal aspecten van de *Trinity* methodologie in zijn geheel besproken en eindconclusies getrokken.
In de algemene discussie wordt eerst een bondige terugblik op de fundamenten van de *Trinity* methodologie geworpen. Enerzijds is *Trinity* gebaseerd op empirische generalisaties. Anderzijds is *Trinity* gebaseerd op een systemisch ontwerp. Hierdoor kan *Trinity* zowel vanuit een "bottom-up" (van praktijk naar theorie) als vanuit een "top-down" (van theorie naar praktijk) benadering begrepen worden.
Verder wordt een aantal onderscheidende aspecten van de methodologie herhaald en bediscussieerd. Meer specifiek wordt aandacht gegeven aan: de filosofische definitie van problemen en probleemoplosprocessen; het onderscheid van zowel intentionele als autonome activiteiten, het onderscheid van zowel een fysisch domein, een kennisdomein en een communicatiedomein, en dit als onderdeel van een integraal model; de systemische opbouw van *Trinity*; de nadruk op pragmatische manieren om multi-actor situaties te verbeteren; de domein-onafhankelijkheid (vrijwel ieder D-type probleem komt in aanmerking); en het feit dat slechts één modelleertaal wordt gebruikt om zowel de "as is" situatie, het tussenliggende veranderingsproces, als de resulterende "to be" situatie weer te geven.
De "ruggegraat" wordt ontvouwd: een gedachtenlijn die de verschillende delen van dit proefschrift samensmeedt in één achtergrondtheorie.
Trinity wordt gepositioneerd in het veld van een aantal belangrijke paradigma's in het omgaan met complexiteit. Meer specifiek worden de overeenkomsten en verschillen met de systeemdynamica bediscussieerd. Systeemdynamica is een kwantitatieve, continue benadering, terwijl *Trinity* daarentegen kwalitatief en discreet is. Dit is een direct gevolg van de nadruk die *Trinity* legt op actoren en intentionele activiteiten: dergelijke noties kunnen eenvoudig uitgedrukt worden in een kwalitatief en discreet paradigma.
Tenslotte wordt de toegevoegde waarde van *Trinity* bediscussieerd vanuit een methodologische invalshoek. Toegevoegde waarde *manifesteert* zich in de *toepasssingslaag* (zie het Experimenten deel), maar wordt *ontleend* aan de gelaagde methodologische opbouw in zijn geheel. Vooral de *filosofielaag*, die gebruikers voorziet van zowel een doel (een model van een multi-actor perspectief) als een middel

(modelleren), en de *methodenlaag*, die een uitgebreide conceptuele "gereedschapskist" verschaft, dragen bij aan deze toegevoegde waarde.

Onze eindconclusie is dat we erin geslaagd zijn modelleermethoden te ontwerpen die specifieke ondersteuning bieden bij multi-actor probleemoplosprocessen. *Trinity* is het treffende voorbeeld: *Trinity* biedt modelgebaseerde ondersteuning bij multi-actor probleemoplosprocessen.

In de praktijk blijkt dat D-type interventies en D-type verbeteringsprocessen doorgaans plaatsvinden op basis van een slechts fragmentarisch en intuïtief begrip van de onderhavige probleemcontext. Om dit te voorkomen, bevelen wij in D-type probleemoplosprocessen het gebruik van *Trinity* sterk aan. Om deze reden eindigt het proefschrift met een aantal aanbevelingen om de toepassing van *Trinity* verder te verbreiden.

CURRICULUM VITAE

Henk Diepenmaat werd geboren op 29 juni 1962 te Brunssum. Hij bracht hier een plezierige jeugd door, als lid van een gezin met vier kinderen. Te Brunssum volgde hij het kleuteronderwijs (St. Jozefschool), de basisschool (Titus Brandsmaschool) en het atheneum B (Romboutscollege).
In 1981 startte hij met de opleiding Scheikunde aan de Katholieke Universiteit Nijmegen. Na een hoofdvak Analytische Chemie te Nijmegen en een hoofdvak Luchthygiëne en -verontreiniging aan de Landbouw Universiteit te Wageningen sloot hij deze studie in 1987 af.
Direct aansluitend trad hij in dienst bij TNO. Hier bekleedde hij verschillende functies bij de verschillende milieu-instituten van deze organisatie, zowel te Apeldoorn als te Delft. Zijn professionele aandachtsgebied was in eerste instantie de toepassing van Kennistechnologie in de verschillende milieudisciplines van TNO. Hij volgde, naast zijn werk, verschillende opleidingen in deze richting: in 1990 behaalde hij een Tweede Fase diploma in de Kennistechnologie (CIBIT, Utrecht) en in 1993 een "Masters Degree" (MSc) in "Knowledge and Information Technology" (Middlesex University, Great-Britain).
Geïnspireerd door de milieuproblematiek ontwikkelde hij een speciale interesse voor multi-actor problemen. In 1990 startte hij een promotie-onderzoek aan de Universiteit van Amsterdam, gericht op dit onderwerp. Het hieruit resulterende proefschrift, dat voor een belangrijk deel naast zijn reguliere activiteiten bij TNO tot stand kwam, ligt voor u.
In de tussentijd trouwde hij met Hilda Wolters, en kregen zij een zoontje, Frank.
Op dit moment is hij werkzaam als Coördinator van het onderzoeksgebied Multi-Actor Processen, en wel bij de afdeling MilieuManagement van de divisie MilieuKwaliteit en Veiligheid van het instituut Milieu, Energie en Procesinnovatie van de Nederlandse organisatie voor Toegepast Natuurwetenschappelijk Onderzoek te Apeldoorn[107].

[107] TNOers zeggen dan: Coördinator MAP bij MM-MKV-MEP-TNO-Apeldoorn. Dat is niet helderder, maar wel veel korter.

DANKWOORD

Zeist, Mei 1997

Dit proefschrift is geschreven in de "wij"-vorm[108]. Dit is gebeurd om twee redenen. Ten eerste is het goed te erkennen dat dit werk, impliciet en expliciet, gebaseerd is op het werk van vele onderzoekers uit verleden en heden. Ten tweede, en zeker niet minder belangrijk, is het schrijven van een proefschrift onmogelijk zonder de directe en indirecte hulp van velen, zowel uit de professionele sfeer als uit de privé-sfeer, die dus óók hun steentje (en in sommige gevallen een volwassen zwerfkei) hebben bijgedragen. Op deze plaats wijd ik graag aan hen allen een woord van dank. Dank!

Een aantal mensen wil ik speciaal bedanken.

Ten eerste een bijzonder woord van dank aan mijn (co-)promotoren: Jacqueline Cramer, Bob Wielinga en Wim Harder. Het was bij tijd en wijle niet eenvoudig wijs te worden uit het palet aan onderzoekstradities dat aan tafel vertegenwoordigd was (het was soms een D-type situatie *an sich*). Vooral in het begin, toen mijn omschrijvingen van wat later *Trinity* zou worden nog erg rammelden en kraakten. De positieve kant was, dat ik mij vanuit vele onderzoekstradities gesteund wist. Het onderzoek was dan ook een ervaring, waar ik met zeer veel plezier aan terug zal denken. Ik wil een paar facetten van de ervaren steun eruit lichten, in het besef dat dit zeer fragmentarisch is. Jacqueline dank ik met name voor haar intensieve betrokkenheid en haar gerichte adviezen (vooral) bij het experimentele deel van het proefschrift. Bob dank ik met name voor de verhelderende discussies over (vooral) theoretische en methodische aspecten van modelleertalen. En Wim dank ik voor zijn niet aflatende steun, in woord en daad, bij vrijwel alle aspecten van het onderzoek, naar zijn eigen zeggen op de achtergrond, maar door mij steeds als erg aanwezig ervaren.

Verder mijn TNO-collega's. In vele hoedanigheden, als lijnmanagers, samenwerkende collega's, geïnteresseerde collega's, motiverende collega's, heftig discussiërende collega's, kritische collega's en soms ook verbouwereerde collega's hebben jullie mij gesteund. Ex-MT'ers, ex-M&E'ers, ex-IMW'ers en MEP-pers; ex-FMT'ers, ex-MC'ers en MM'ers. Wat je iedere dag ervaart wordt heel gewoon, en schat je wellicht niet genoeg naar waarde. Zonder jullie had dit proefschrift wellicht niet bestaan en had het er in ieder geval heel anders uitgezien.

---

[108] Vanzelfsprekend draagt de auteur de volledige verantwoordelijkheid voor de inhoud van dit proefschrift.

Een bijzondere rol hebben met name de volgende mensen gespeeld. Peter Engelmann, die in het prille begin een lans brak voor mijn plannen. Anton van der Linden, die mij leerde analyseren. Vele ex-MC'ers, maar in het bijzonder Huygen. De massa van kennis is een begrip dat we toch nog eens netjes uit moeten werken! Joost Quakernaat, met wie ik diepgaande discussies heb gehad over KDS en macht (dat varkentje is nog niet gewassen...). Addie Weenk, "brother in arms", en dat al vele jaren. Pieter van Langen, gedetacheerd geweest vanuit de VU. Ik heb veel opgestoken van onze discussies (en dat ondanks dat je van de foute universiteit komt ;-). Rob van der Spek, afstudeerbegeleider bij het CIBIT. De echo's van mijn scriptie zijn nog duidelijk in dit proefschrift hoorbaar.

Dan mijn mede-auteurs en de mensen die anderszins bij de experimenten behulpzaam zijn geweest. Maarten Leupen. Door jou ben ik de diepte van het begrip "expertise" pas goed gaan begrijpen. Je wilde "in kunnen zoomen" op het binnenmilieu. Nou, ik heb mijn best gedaan. Chris Bruijnes, met wie ik regelmatig op VOSsenjacht ga. Ik gebruik zelfs een andere deodorant om ze niet te verspreiden. Frans Vollenbroek, door wie ik betrokken ben geraakt bij strategische conferenties. Helmut Beckhuis, die mij de mogelijkheid heeft gegeven *Trinity* verder uit te werken en in de beleidspraktijk te introduceren. De leden van het projectteam "Knelpuntenanalyse KWS2000", met wie ik een erg boeiend traject heb doorgemaakt. En Hannie Tijman, die voor redactionele ondersteuning zorg gedragen heeft.

Verder een woord van dank voor een aantal dappere afstudeerders. Rieke Kok, die samen met Maarten menige boom heeft opgezet over het binnenmilieu. Mark Vermeulen, die er na zijn afstudeeropdracht nog een jaartje *Trinity* aan vastknoopte. Mark, ik wacht met spanning op je finale interpretatie van *Trinity*. Lars van Lierop, tevens mede-auteur van het KWS2000 experiment, die het aandurfde bij mij zowel zijn stage als zijn afstudeeropdracht te doen. Of dat ooit nog goed komt?

Mijn ouders dank ik, aan wie dit proefschrift opgedragen is. Het spreekwoord schiet hier tekort: een goed begin is veel meer dan het halve werk.
Mijn overige familie, en vrienden. Jullie waren steeds belangstellend. Pas maar op, nu mijn proefschrift er ligt word ik weer sociaal!

Dan Frankje. Voor jou was het wel duidelijk waar je mij, luidkeels en enthousiast "papa, papa" roepend, vinden zou als je me weer eens niet zag. Je liep dan blindelings naar de studeerkamer, naar mijn bureau, waar ik langzaam maar zeker een onderdeel van aan het worden was. Je hebt zelfs nog enthousiast meegeschreven aan dit proefschrift, waarschijnlijk onder het motto van "Als hij dan niet met mij mee doet, dan doe ik wel met hem mee!". Jouw bijdragen hebben het weliswaar niet gered (ik kon er geen soep van trekken), maar je geestdriftige steun waardeerde ik des te meer. Ik ga veel meer met je stoeien, dat is beloofd (ôh-ôh). Als kleine tegemoetkoming voor gederfde stoeipartijen geef ik mijn ruime studeerkamer voor jou op. Ik verhuis naar de zolder. Dan moet jij wel je huidige kamertje opgeven. De babyspulletjes kun je laten staan.

Tenslotte jij, Hilda. Dank je voor het feit dat je me al die tijd gesteund hebt, op al die verschillende manieren. Vooral als alle dingen tegelijk komen, valt dat soms niet mee. Het is fijn je leven te kunnen delen met iemand met wie je enerzijds samen je knulletje in bed stopt (dat is altijd een mooi moment), en die anderzijds, als ze constateert dat je "op slot zit" (en de tijd dringt), tot diep in de nacht met je meewerkt aan de laatste delen van je proefschrift. De zomer van mijn leven is mooi met jou, en ik zie de rest met veel plezier tegemoet.

Henk

LIST OF FIGURES

Chapter 1
Figure 1: The relations between the research questions. 4
Figure 2: The five symbiotic elements of a methodology (derived from the methodological pyramid [Wielinga et al. (1989)]). 6
Figure 3: An overview of the research approach. 8
Chapter 2
Figure 1: A generic model of intentional activities. 18
Figure 2: A perspective is a mental model consisting of three tightly coupled parts. ..19
Figure 3a: "As is" and "to be" are relative to taking action. 25
Figure 3b: A typology of intentional activities. 25
Figure 4: A perspective is linked to an *environment* by means of a trivalent model relation. 34
Figure 5: A perspective is linked to an *intention* by means of a trivalent model relation. 35
Figure 6: A perspective bi-directionally modelling an intention and an environment. 36
Figure 7: A perspective may model an intention and/or an environment, and may be modelled in turn by an (explicit) model. 44
Chapter 3
Figure 1: Two points of view, one body of knowledge. 54
Figure 2: Knowledge Distribution Space. 55
Figure 3a: A three-dimensional iso-plane 59
Figure 3b: A cross-section of a sequence of iso-planes. 60
Figure 4a: Knowledge types according to the adherence attribute. 66
Figure 4b: The ABCD distinction. 68
Figure 5: Different examples of bodies of knowledge (a qualitative representation)...74
Chapter 4
Figure 1: The meaning triangle and a model relation. 82
Figure 2: Model relations. 83
Figure 3: Extremes in multi-referent and multi-representation model relations 85
Figure 4: Two modelling processes, structurally the same, semantically different. 88
Figure 5a: Parallel modelling steps. 91
Figure 5b: Referent modelling steps. 93
Figure 5c: Representation modelling steps. 95
Figure 6: Constraints in sequencing modelling steps (the *tetrahedron* figure). 96

Figure 7a: Sequencing referent and parallel steps via an intermediate simple relation enables one to convert a multi-referent relation into a parallel one ..97
Figure 7b: Specifying a complex relation via an intermediate simple relation. 98
Figure 7c: The Deux-Chevaux beats the Porsche. 99
Figure 8: Strategies of the structural family. 101
Figure 9: Parallel modelling processes. 102
Figure 10: Referent modelling strategies. 104
Figure 11: Examples of applying a referent strategy. 105
Figure 12: Building blocks and transformation strategies. 106
Figure 13: Reduction and expansion strategies with respect to **s** (models with a fixed number of systemic levels). 109
Figure 14: The cognitive threshold. 112
Chapter 5
Figure 1a: A stage-based model of an intentional activity. 120
Figure 1b: A recursive spiral of intentional activities. 121
Figure 1c: *Trinity* is directed at supporting knowledge brokers in D-type problem solving. 122
Figure 2a: The *Trinity* approach: a bird's eye view. 124
Figure 2b: A D-type problem solver using *Trinity* is engaged in three core activities. 126
Figure 2c: A sequence of *Trinity* models represents an evolving interpretation of a problem context in terms of a perspective. 129
Figure 3: Reference model primitives. 133
Figures 4a and 4b: Complex models that do not make sense. 135
Figure 5a: Causation in *Trinity* models: an autonomous process. 136
Figure 5b: Representing intentional activities. 137
Figure 5c: Pragmatically incorrect models of intentional activities. 139
Figure 5d: The neighbour modelled as an intentional actor. 141
Figure 5e: A model of the *perspective* of the neighbour (which is a more detailed model of his environment). 142
Figure 6: A model of a perspective, aiming at changing both the physical environment and the perspectives of citizens. 143
Figure 7: Mixing different levels of genericity. 145
Figure 8: Combining different representations into one model. 147
Figure 9: Modelling steps (the tetrahedron figure). 148
Figure 10: Examples of parallel extensions and restrictions. 151
Figure 11a: Static parallel extensions and restrictions (the shading denotes that something is added/deleted). 152
Figure 11b: Dynamic parallel extensions and restrictions in the physical domain. ...152
Figure 11c: Dynamic parallel extensions and restrictions in the communication domain. 153
Figure 11d: Dynamic parallel extensions and restrictions in the knowledge domain.153
Figure 11e: Inter-domain parallel extensions and restrictions. 154

Figure 11f: "Knowledge about knowledge" extensions and restrictions in the knowledge domain. 155
Figure 12: Examples of parallel transformations (both static and dynamic). 157
Figure 13: Static parallel transformations. 158
Figure 14a: Dynamic parallel transformations of domain states. 159
Figure 14b: Dynamic parallel transformations of processes. 160
Figure 14c: Two examples of dynamic parallel transformations of arrows. 161
Figure 15: Parallel *Trinity* strategies. 165
Figure 16: Sequencing parallel specification steps results in a parallel specification strategy. The inverse is a parallel abstraction strategy. 166
Figure 17: A parallel specification strategy consisting of several steps (the inverse is a parallel abstraction strategy). 167
Figure 18: A transformation strategy. 168
Figure 19: A dynamic extension strategy (inverse: a dynamic restriction strategy)... 169
Figure 20: A simple bird's eye strategy. 170
Figure 21a: A specification strategy in modelling a perspective 171
Figure 21b: An extension strategy in modelling a perspective 171
Figure 22: Construction and use of a generic model. 172
Figure 23: The minimal environmental situation of concern. 174
Figure 24a: A generic physical domain script. 176
Figure 24b: A generic knowledge domain script. 176
Figure 25: The *Trinity* process from a bird's eye view. 185

Chapter 7

Figure 1: The indoor environmental problem solver in his problem context, following a trouble shooting approach. 204
Figure 2a/2b: Generic "as is" models. Figure 2a presents the "minimal environmental situation of concern". Figure 2b presents the "minimal **indoor** environmental situation of concern". 206
Figure 3: A more detailed generic "as is" model of indoor environmental situations of concern. 207
Figure 4: A generic "as is" model for house dust mites causing allergic asthma. 210
Figure 5: A generic model for chipboard causing several complaints. 211
Figure 6: A generic model for photocopiers causing headache. 212
Figure 7a: A generic script that alters the physical domain. 214
Figure 7b: A generic script that alters the knowledge domain of an indoor environment. 214
Figure 8: Remedying the ozone-related complaints caused by a photocopier. 217
Figure 9: A proposal for an infrastructure that is based on model-based support for indoor environmental problem solving. 221
Figure 10: A hierarchy of generic "as is" models. 223

Chapter 8

Figure 1: An overview of the *VOC2000* experiment in *Trinity* notation. 230
Figure 2: Stages in the knowledge acquisition process. Stages that encompassed

intensive interaction and communication with field players are shadowed. 234
Figure 3: A *Trinity* model of the *VOC2000* process. 240
Chapter 9
Figure 1: The strategic conference *Building and demolition waste* in its problem context. 247
Figure 2: An essential "as is" model of the building and demolition sector........249
Figure 3: An essential "to be" model (for the year 2010) of the building and demolition sector: environmental chain closure........251
Figure 4: A detailed "to be" model of the building and demolition sector. 253
Chapter 11
Figure 1a: Two-dimensional examples of *Trinity* building block strategies as movements in KDS. 285
Figure 1b: A two-dimensional example of *Trinity* transformation strategies as movements in KDS. 286
Figure 1c: A two-dimensional example of a *Trinity* balancing modelling strategy as a movement in KDS. 288
Appendix B
Figure 1: Examples of (in)correctness. 309
Figure 2: Three alternative "Lawn mower" models. 310

LIST OF TABLES

Chapter 3
Table 1: A problem typology based on knowledge distribution. 70
Table 2: Problem types, methodological principles and typical problem solvers 73
Chapter 4
Table 1: A selection of the typology of ways to change a model relation. 87
Table 2: A typology of primitive modelling steps. 89
Chapter 5
Table 1: A typology of primitive modelling steps. 149
Table 2: Example elements of intention and process lists 182
Table 3: Distinctions on the basis of the intention list. 184
Table 4: Languages, formats and strong points. 187
Chapter 8
Table 1: Examples of *VOC2000* problem areas and the actors that were derived from them. 236
Chapter 9
Table 1: Shifts in attention between actors, on the one hand, and roles/functions, on the other. 260
Chapter 11
Table 1: Paradigm characteristics of *Trinity* and Systems Dynamics. 291

INDEX

A

ABCD problem typology 68
abstraction function of models 44
abstraction modelling steps 87
abstraction strategies
in *Trinity* syntax 164
acknowledgement stage 138
action potential 127; 190
a perspective coupled to an intention and an environment 22
and pragmatic correctness 279
as meaning (in footnote) 190
perspectives as 282
problem solving as the generation of 288
action potentials
contrasted with scenario's (in footnote) 177
pragmatically correct perspectives as 267
actor
as an agent capable of realising bi-directional coupling 36
definition of 18
from actor to role 260
actor engaged in an intentional activity
Trinity rules to represent an 144
actor, intentionally acting in an environment
Trinity representation of 137
actors
deriving roles from actors 235
A_{cu}-type problems 68
added value
a methodological point of view 294
Building and demolition waste 262; 270
Indoor environmental problems 224; 270
of the methods layer 295
of the philosophical background 294
of the tools layer 296
of the use layer 296
VOC2000 .... 243; 270
added value of *Trinity*
discussion on the basis of the experiments 263
adherence
a qualitative interpretation 65
adherence of a point of view 51
A_{di}-type problems 68
Advanced Beginner 65
advisors 183
allergic asthma 209
ambiguities in *Trinity* models
eternal loops 146
mixing alternatives (xor) with combinations (and) 147
mixing levels of genericity 146
and (logical and) 136; 146; 147; 307
A_{pe}-type problems 68
appropriateness of discrete, qualitative paradigms for modelling schemes for improvement 292
arrow 135
arrow (causation in *Trinity*) 135
arrows, specification of (see also the footnote) 156
as is 18
Atomic perspectives 54
atomic subsystems 313
A-type problems 68
autonomous activity list 181
autonomous process 136
Trinity representation of 136
avoiding pain 23
avoiding pain behaviour 24

B

backbone 283
backbone relation 283
back-casting 259
when to use 266
back-casting mode 189
bad 39
balancing
between scope and level of support 3
balancing mechanism 112
balancing strategies 107
and multi-level *Trinity* models 161
balancing strategy 108
in summary 108
battery example 143
B_{cu}-type problems 69
B_{di}-type problems 69
Beginner 65
bi-directional model relation 35
bi-modal approach 265
bird's eye view strategy

example 169
in *Trinity* syntax 164
bird's eye strategy 102
bold notation 146
bootstrap phenomenon
in inviting participants in problem solving 281
bounded rationality 16
B_{pe}-type problems 69
brown lawn example 140
B-type problems 69
buckets
the filling buckets analogy for perspective construction 27
Building and demolition waste
overview of the problem context 199
building blocks modelling steps 87
building blocks strategies
in *Trinity* syntax 164
building blocks strategy 105
business process redesign 2

C

causal environmental chain 213
causal relation 134
causal relationship 135
necessary but insufficient 135
causal relationship in Trinity 135
C_{cu}-type problems 69
C_{di}-type problems 69
central concepts of this dissertation 3
central research question 3
chaining by means of a shared environment 41
change referents 103
chipboard 209
circular
definition of actor and action (in footnote) 19
definition of perspective and action (in footnote). 19
claims of intentionality
the presence of a perspective 27
c_{max} 110
cognitive mapping approach 190
cognitive threshold 108
a visual presentation 111
and its role in a balancing strategy 287
Common knowledge 67
common perspective 67
Communication 293
definition of 131
communication and control 293
communication domain 130
communication domain process 132
communication domain state 132
communication function of models 44
communication viewpoint 278
Competent Performer 65
completeness of the typology of modelling steps.... 113
complex modelling strategy 100
complexity
a qualitative interpretation 65
complexity family of modelling strategies 106
complexity of a point of view 51
concept
as used in the meaning triangle 81
conceptualisation 81
conscious
a counscious mode 26
consultant 129
continuous 289
control 293
critique on 293
co-ordination 70
co-ordinator 71
C_{pe}-type problems 69
critique 1
several notions are not included 21
critique 2
one generic model is to simple to cover all intentional activities 23
critique 3
stages 2a, 2b and 2c do not take place in sequential order 26
critique 4
the model assumes an axiomatic mind-body dualism 28
critique 5
the presumed rigidness of a stage-based approach31
critiques
with respect to our generic model of intentional activities 21
crossing the river
example to explain the exclusion of the behavioural interpretation of problem solving 21
C-type problems 69
cultural effects 71
Cultural knowledge 67
cultural perspective 67
curative
curative intentional activities 24
cycles in *Trinity* models introduce ambiguity 291

D

D_{cu}-type problems 69
D_{di}-type problems 69
deadlock situation
in D-type problem solving 76
decision making (in footnote) 289
decision-making paradigm
its incorporation in *Trinity* 289
degeneration problem 280
as a statement of inability and inadequacy 281

manifestations of....281
descriptive model
as the "as is" part of a model of a perspective....18
descriptive model relation....34
Disciplinary knowledge....67
disciplinary perspective....67
discrete....289
discrete event simulation....301
discrete world view....292
disturbance
in the routine flux of events....28
diversity
a qualitative interpretation....65
diversity of a body of knowledge....51
document scanning....181
domain-time continuum....291
double model relation....35; 43
D_{pe}-type problems....69
D-type problem ownership....299
recognition and awareness of....299
D-type problem solving as a profession....269; 302
D-type problems....69
dualism and relativism....33
dualistic and relativistic stance....30
dynamic parallel abstractions
in *Trinity* syntax....156
dynamic parallel restrictions
in *Trinity* syntax....150
dynamic parallel specifications
in *Trinity* syntax....156

E

efficiency function of models....44
ellipse
introduction of....133
emergence....293
emergence and hierarchy....293
emerging properties....31
endomorphic systems....140
environmental chain closure....250
environmental problems
as multi-actor problems....i
environments
perspectives and intentions....33
essential models....248
evaluation arrow
when allowed to omit (see rule 5)....144
evaluation stage....138
examples of possible relations between perspectives, intentions and environment....37
exclusive or....21; 24; 146; 179
exclusive or (logical xor)....147
expansion strategies....107
in *Trinity* syntax....163
experience-action cycle
of Checkland....17
experiental learning....22
Expert....65; 69
explanation function of models....44
explorative function of models....44
extension modelling steps....87
extension strategies
in *Trinity* syntax....164

F

field players....183
as interviewees in the VOC2000 experiment....231
final conclusion....299
find correct level of detail strategy....102
find correct level of generalisation strategy....103
first principles of *Trinity*....276
focus on details strategy....102
focus strategy
in *Trinity* syntax....165
formaldehyde....209
formalisation function of models....44
Fourier analysis (in footnote)....51
frame format....181
frame problem
in footnote....280
frames....20
perspective construction as a sequence of....46
functions of models....44
future research....300

G

general functions of models....44
generalisation....23
in terms of perspectives....23
generalisation function of models....44
generic....84
generic knowledge domain script....176
generic model....103; 172
examples in indoor environmental problems....213
generic model of intentional activities....18
generic model strategy....172
generic physical domain script....176
Generic scripts for the indoor environment....213
Generic Systems Space....313
genus....84
good....39
greenhouse-with-water-tap example....41
ground rule of *Trinity*....268
GSS (Generic Systems Space)....313
Lego example....314
guidelines for designing qualitative modelling languages....114

H

habitat ... 209
handbook ... 299
here
  of actors ... 22
heuristic function of models ... 44
heuristic function of *Trinity* models ... 129
hexagon
  correspondence with intentional activities ... 138
  introduction of ... 133
  six corner positions of ... 138
hidden use ... 188
  bi-modal approach ... 265
  when to use ... 264
hierarchy ... 293
  critique on ... 293
hierarchy of generic "as is" models ... 222
hierarchy of generic models ... 205
homonym ... 83
house dust mites ... 209
how ... 18

I

improvement ... 39
in line use of Trinity ... 260
Indoor environment as a D-type situation ... 203
Indoor environmental problems ... 198
  overview of the problem context ... 198
inductive learning ... 23
  in terms of perspectives ... 23
informer ... 129
intention list ... 181
intentional acting
  is all acting intentional acting? ... 27
intentional *action* ... 18
  Trinity representation of ... 136
intentional activities
  definition of ... 17
  proactive, preservative, preventive and curative ... 24
  routine and problematic activities ... 25
intentional activity list ... 181
intentions
  definition of ... 34
  perspectives and environments ... 33
invisible strategy ... 103
invitation paradox ... 281
  see also bootstrap phenomenon ... 281
iso-effort planes ... 60
isolated use ... 188
  when to use ... 264
iso-plane ... 59
  definition of ... 59
  equation of ... 59
  jumping towards another ... 63
  moving on a ... 62
  visual examples of ... 59
italics notation ... 147
iterative approaches ... 17

J

Jetztpunkt ... 22

K

KD ... 54
KDS ... 54
  overview of features ... 64
  quantitative definition, qualitative interpretation ... 75
key features
  1. definition of problem solving ... 277
  2. distinction of intentional and autonomous activities ... 278
  3. three domains, three viewpoints ... 278
  4. systemic nature ... 278
  5. pragmatic correctness ... 279
  6. genericity with respect to D-type problems ... 282
  7. one language for modelling all three parts of a perspective ... 282
key features of *Trinity* ... 277
knowledge
  definition of ... 22; 131
knowledge acquisition ... 125
  definition of ... 127
  methods for ... 127
knowledge acquisition plan (VOC2000) ... 232
knowledge acquisition process
  and its relation to the *Trinity* modelling process 129
knowledge broker ... 71
  characteristics of ... 72
knowledge distribution
  and its relation to problem solving ... 49
  formal definition ... 54
  informal introduction of ... 49
knowledge distribution attributes ... 51
  mathematical formulas of ... 53
Knowledge Distribution Space ... 54
knowledge domain ... 130
knowledge domain problem-solving strategies ... 193
knowledge infrastructure ... 218
knowledge management ... 2; 224; 278
knowledge management function of models ... 44
Knowledge processes ... 57; 131
knowledge state ... 56
Knowledge states ... 131
knowledge step ... 58
knowledge viewpoint ... 278
knowledge workers ... 252

L

languages 84
Lawn mower examples 311
lawn mower model 139
as an example of pragmatical correctness 139
layers of the methodological pyramid 4
learning
in terms of perspectives 22
learning cycle
of Kolb 17
learning mechanisms 23

M

meaning *is* action potential (in footnote) 190
meaning triangle 81; 126
and model relation 82
methodical framework 295
guidelines for use 263
Trinity as a 263
methodological pyramid 4
methodological research 5
methodology 5
methods layer 5
mind-body dualism 28
mind-body problem 29
pragmatic solution to 29
mind-matter problem 29
minimal environmental situation of concern
generic model 173
mis-correspondence
between environment and intention 35
between intentions and environments 28
mode of communication in the experiments 200
model
definition in terms of use 81
general functions of 44
model primitives 133
model relation 81
and meaning triangle 82
complex model relation 82
multi-referent model relation 82
multi-representation model relation 82
parallel model relation 82
simple model relation 82
three different types 82
when satisfactory 85
model relations and KDS (in footnote) 82
model-based support for problem solving 43
modelling heuristics 192
modelling process
definition of 86
mathematical formula of 86
modelling processes 85
modelling step
a typology of 86
abstraction 87
building blocks type 87
conservation principle (transformation type) 87
definition of 86
extension 87
restriction 87
semantic dimension 87
specification 87
transformation type 87
modelling strategies 163
definition of 100
families of 100
in *Trinity* syntax 163
why useful 100
modelling strategies visualised in KDS 285
movie analogon 46
multi-actor
informal introduction of i
multi-actor processes
examples of 2
multi-level models 107
multi-level *Trinity* model 161
multiplicity of points of view 266
multi-referent 83
multi-referent model relations
Trinity representation of 144
multi-referent restriction strategy
when to use 267
Multi-referent steps 92
multi-referent strategies 100
in *Trinity* syntax 172
multi-representation model relations
representation in *Trinity* 146
multi-representation relation 84
Multi-representation steps 94
multi-representation strategies 100
mutual benefit
as the rationale for co-operation 293

N

natural language 186
natural language and its flaws 186
non-monotonous process
problem solving as a 27
Non-primitive modelling steps 96
definition of 96
now
of actors 22

O

o (overall complexity) 107

object
as used in the meaning triangle....81
objective....28; 35
opinions of field players....231
orange example....155
overall complexity (**o**)
definition of....107
overall goal of the research....297
ozone....209

P

pain
avoiding pain behaviour....24
paradox
the invitation paradox....281
parallel abstraction....91
parallel abstractions
in *Trinity* syntax....155
parallel bird's eye strategy
visualised in KDS....287
parallel building blocks strategy
in the Building and demolition waste experiment259
when to use....267
parallel extension....90
parallel extension strategies
when to use....267
parallel extensions
in *Trinity* syntax....150
parallel model relation
Trinity representation of....134
Parallel modelling steps
in *Trinity* syntax....149
parallel modelling strategies....101
parallel relation....83
parallel restriction....90
parallel restrictions
in *Trinity* syntax....150
parallel specification....91
parallel specification strategy
when to use....267
parallel specifications
in *Trinity* syntax....155
Parallel steps....90
parallel strategies....100
in *Trinity* syntax....163
Parallel *Trinity* strategies examples....165
Parallel *Trinity* strategies overview....165
part of....134
participative modelling cycle
in policy analysis....17
participative use....188
bi-modal approach....265
when to use....265
parts of society....56
Personal knowledge....67
personal perspective....67
perspective
as a threefold conglomerate structure....18
atomic perspectives....51
definition of....18
recursive structure of....51
should be modelled as a whole....266
perspective construction
as a modelling process....43
as one complex stage....26
perspectives
and intentional actions....50
as a more specific alternative for the concept knowledge....50
environments and intentions....33
possessing perspectives....22
processing perspectives....22
re-introduction and elaboration of....50
perspectives embody action potentials....190
philosophy layer....5
photocopiers....209
Physical....131
physical domain....130
physical domain problem-solving strategies....193
physical processes....131
physical reality
another way of looking at....43
as a common sense notion....39
physical states....131
physical viewpoint....278
pleasure
seaking pleasure behaviour....24
policy cycle....17
policy programme....227
power....293
pragmatic conventions....305
pragmatic conventions of *Trinity*....140
pragmatic correctness....270; 282
as a driving and guiding force....280
as a hypothesis....308
discussion of....279
turns knowledge into action potentials....281
pragmatically correct....139; 170
a more intuitive description (in footnote)....279
an alternative formulation....282
and bi-directional model relations....36
indoor environmental models....213
predictive model
as the "to be" part of a model of a perspective....18
predictive model relation....34
premature closure....16; 178
prescriptive model
as the script part of a model of a perspective....18
prescriptive model relation....34
preservative
preservative intentional activities....24
preventive
preventive intentional activities....24
primary resources....245

principle of rationality.......16
proactive
proactive intentional activities.......24
problem.......19
definition of.......19
problem areas
in the VOC2000 experiment.......231
problem context
definition (in footnote).......125
informal introduction of.......i
intuitive vs. model-based understanding of.......269
problem owner....... 127; 198; 266
problem ownership
establishing of.......269
problem solving
a combinatorial paradigm.......71
and its relation to knowledge distribution.......49
as acting.......16
as decision making.......16
as re-establishing bi-directional coupling.......38
as the generation of alternatives for taking action 16
definition of.......20
different interpretations of.......15
exclusion of the behavioural interpretation.......21
our definition and its relation with the other three interpretations.......20
questioning its meaning.......15
refined definition of.......38
result of.......20
problem solving process
the design of (in footnote).......232
Problem typologies
examples of.......64
problem-solving processes
supporting the design by means of *Trinity*.......282
process.......131
process list.......181
Processes
definition of.......132
relativistic nature of.......132
Proficient Performer.......65
Pushing clouds.......238
pyramid.......5

Q

qualitative.......290
qualitative modelling
a generic theory of.......79
examples of paradigms.......79
qualitative models refer to qualities (in footnote).......290
qualities
in designing a modelling approach.......114
to be distinguished in a referent.......130
qualities and their symbols.......133
qualities of interest of the *Trinity* language.......131
qualities referred to in a *Trinity* model.......131
quantitative.......290

R

rational
limitations of.......282
rationalism.......28
rationalistic philosophical stance.......28
rationality
Newell's principle of.......16
reactions from users.......268
recommendations.......299
rectangle
introduction of.......133
reduction strategies.......107
in *Trinity* syntax.......163
reference model primitives.......133
rules for connecting.......136
referent.......81
referent abstraction.......93
referent extension.......93
referent generalisation strategy.......103
referent modelling steps
in *Trinity* syntax.......162
Referent modelling strategies.......103
referent restriction.......93
referent specialisation strategy.......103
referent specification.......93
reflection.......41
reflection in a *Trinity* model.......142
reflective architectures.......41
reflective chaining.......41
reflective function of models.......44
relation
between the three interpretations of problem solving.......19
relational qualities.......131
relativism
an extreme interpretation.......279
evaluation is the proof of the pudding.......279
relativistic and dualistic stance.......30
repetition effects.......71
representation.......81
representation abstraction.......94
representation extension.......94
Representation modelling steps
in *Trinity* syntax.......162
representation restriction.......94
representation specification.......94
Representation strategies.......105
representational function of *Trinity* models.......129
requirements for intentional action.......36
research approach.......4
research questions
relation between the.......3
restriction modelling steps.......87
restriction strategies

in *Trinity* syntax ... 164
result of problem solving
one (1) perspective ... 21
role
from role to actor ... 260
roles
deriving roles from actors ... 235
rounded box
introduction of ... 133
rules of thumb
for selecting a mode of communication ... 264
with respect to the modelling strategy to be used267
with respect to the order in which a perspective should develop ... 265

S

s (structural complexity) ... 107
scenario
contrasted with action potentials (perspectives) . 177
scenario strategy ... 177
scenario-like approach ... 190
schemas ... 20
scientific experiment ... 27
in terms of a perspective ... 27
scope of a Trinity model ... 163
scoping strategy ... 102
script ... 18
scripts branch off the "as is" situation ... 215
seeking pleasure ... 24
seeking pleasure behaviour ... 24
semantic conventions ... 305
semantic conventions of *Trinity* ... 140
semantic correspondence ... 133
semantic dimension of modelling steps ... 88
semantic family of modelling strategies ... 105
semantic transformation rule of *Trinity* ... 156
semantically correct ... 139; 305
seven why's (in footnote) ... 42
shift downward ... 41
shift upward ... 41
similarity ... 313
situated
perspectives situated in time and space ... 22
social change ... 57
societal change ... 57
specialist ... 71
specification modelling steps ... 87
specification strategies
in *Trinity* syntax ... 164
specifications on causations (on arrows)
in *Trinity* syntax ... 156
spiral model
Schön ... 17
state ... 131
States
definition of ... 132
relativistic nature of ... 132
static parallel abstractions
in *Trinity* syntax ... 156
static parallel restrictions
in *Trinity* syntax ... 150
static parallel specifications
in *Trinity* syntax ... 156
structural complexity (s)
definition of ... 107
structural family modelling strategies ... 100
structure-process-effort correspondence assumption. 60
sub-clinical complaints ... 203
subject of this dissertation ... 2
subjective ... 28; 35
sub-questions
derivation from the central research question ... 3
support of *Trinity* in the building and demolition waste experiment ... 262
sustainable society
the requirement of systemic applicability of our model of intentional activities ... 32
symbiotic ... 5
relation between the layers of the pyramid ... 5
symbiotic nature of systems levels ... 293
symbol
as used in the meaning triangle ... 81
synonyms ... 84
syntactic conventions ... 305
syntactical conventions of *Trinity* ... 140
syntactical transformation rules of *Trinity* ... 156
syntactically correct ... 305
syntax-preserving ... 149; 168; 307
syntax-preserving design of modelling steps ... 149
system ... 313
as an iso-plane ... 313
system boundary of a *Trinity* model ... 137
systemic
nature of *Trinity* ... 276
systemic construct ... 277
definition (in footnote) ... 276
systemic level
changes in ... 31
Trinity's independence of ... 32
systemic level of a *Trinity* model ... 163
systemic levels ... 31
systemic nature
of *Trinity* ... 278
systemic reflection ... 41
systems concept ... 313
Systems Dynamics
a minimal description of ... 290
and simulation ... 290
and *Trinity* ... 291
comparing *Trinity* with ... 289
qualitative or quantitative ... 290
Systems notion of *Trinity* ... 313
dimension 1
parts as a whole ... 313

dimension 2
all parts are different 313
dimension 3
all parts are the same 313
systems thinking 293

T

tetrahedron figure 97
textual attachment 193
theoretical features of KDS 56
theory layer 5
thinking
definition of 22
this
of actors 22
this-now-here 22; 292
threefold structure 18
of a perspective 18
time scale of the experiments 199
time-space continuum 291
tools layer 5
trajectories in KDS 57
trajectories in KDS as Trinity modelling processes .285
transformation modelling steps 87
transformation strategies
in *Trinity* syntax 164
transformation strategy 106
translation of actors to roles 235
translation of problem areas to actors 235
translation of roles to *Trinity* models 235
trick-exploiting mode 189
trigger 181
definition of 181
Trinity
added value 269
and Systems Dynamics 291
as a means to design D-type problem solving processes 282
explanation of name 315
first principles 276
from practice to theory 276
from theory to practice 276
groundrule of 268
introduction of 2
its position in the field of mainstream paradigms for dealing with complexity 283
key features of 277
middle-out design 276
parts of the methodical framework 298
review and discussion of its basis 275
the central issue in problem solving 280
three trinities 315
three-dimensional systems notion 279
tools 299
Trinity modelling process
and its relation to the knowledge acquisition process 129
Trinity modelling steps and knowledge processes ... 284
Trinity modelling steps and trajectories in KDS 284
Trinity models and knowledge distributions 284
Trinity models and position in KDS 284
Trinity principle 130; 277
Trinity process 184
trivalent model relation
between perspectives and environments 34
trouble-shooting 189
when to use 265
typing intentional activities 24

U

ureum-formaldehyde 209
use layer 5

V

value system 39
vanishing properties 31
viewpoints 84; 94; 162
VOC2000
overview of the problem context 198

W

waterfall approaches 17
water-tap examples 37
workflow management 2
worlds, as in "it worlds"
as a verb (Heidegger, in footnote) 37

X

xor
see exclusive or 21