Ahonen, Jarmo

Modeling Physical Domains for Knowledge Based Systems — Joensuu: University of Joensuu, 1995. 127 p. (Joensuun yliopiston luonnontieteellisiä julkaisuja 33 = University of Joensuu. Publications in Sciences, 33. ISSN 0781-0342).

ISBN 951-708-367-X

Keywords: knowledge engineering, expert systems, simulation, modeling

This is a collection of five articles which deal with domain modeling. The articles span from the usefulness of qualitative modeling to the algorithms which could be used to reduce the computational burden associated with simulating multiple biotic agents. The main theme of the articles is that we should pay more attention to the semantics of the modeled domain and use tools which suit the domain.

The first article discusses the usefulness of qualitative modeling and concludes that the philosophical foundations of qualitative modeling are questionable. The questionable nature of these foundations is mainly due to the fact that neither philophers nor artificial intelligence researchers have been able to establish any logical primitives. This makes the fundamental assumptions behind qualitative modeling unsuitable for the modeling of continuous systems. The use of causal processes for quantitative modeling is proposed as a tool for the creation of structural models.

The second article discusses the terms *deep knowledge* and *robustness*. After clarifying the terms, the nature of causality in domain models is discussed and the structure of quantitative causal models is presented. The robustness of the models is discussed and the creation of such models is tied to the classical simulation model creation steps.

The causal model structure is extended and experimentally used for simulation in the third article. The nature of the knowledge used in model creation and simulation is discussed further and the causal model structure is extended towards the creation of working simulation models. A simple programmin environment is developed alongside the creation of a causal process simulation language. An experimental simulation model of pine is created and experimental simulation carried out. The results reveal that the approach surprisingly is surprisingly robust.

The fourth article discusses the creation of formal explanations from the simulation knowledge. The Salmonian approach to the causal structure of the world is integrated to a causal calculus which has been created especially for causal explanation.

The last article outlines a method which can be used to reduce the computational burden associated with the simulation of multiple biotic agents. The same method can be used with any type of structural and hierarchical models.

Jarmo J. Ahonen, Department of Computer Science, University of Joensuu, P.O.Box 111, SF-80101 Joensuu, Finland.

I want to thank my wife, Hilja, for invaluable support over the years.

Contents

List of original publications included in this thesis:

Ahonen, J. J. (1994). On Qualitative Modelling. *AI & Society*, 8. 17-28. (Ch. 2)

Ahonen, J. J. (1995). Deep knowledge and domain models. *Informatica*, 19. 265-279. (Ch. 3)

Ahonen, J. J. (1995). Causal process modeling. Submitted for publication. (Ch. 4)

Ahonen, J. J. (1991). Domain Knowledge and Explanation. Presented in *Bar-Ilan Symposium on the Foundations of Artificial Intelligence*, June 1991. (Ch. 5)

Ahonen, J. J. and Saarenmaa, H. (1991). Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents. In: *Computer Science for Environmental Protection*, 6th Symposium, München, December 1991, Proceedings. Springer-Verlag, Berlin. 193-200. (Ch. 6)

Chapter 1

Foreword

Jarmo J. Ahonen

1.1 Introduction

This collection of five articles represents the results of a frustrating venture to the land of knowledge representation. The venture started in Summer 1988 when I started to work for the Faculty of Forestry, University of Joensuu. My original task was to participate in an expert system development project for strategic forest management, and as a simple subtask in that project was the implementation of the knowledge of the beginning of the life-cycle of Scotch pine. In addition to the implementation of the pine-knowledge we were to implement a simple decision support system for regeneration planning.

Unfortunately the easy looking subtask, the implementation of the pine specific knowledge into a simulation model, turned out to be a very difficult and frustrating experience. I and my colleague Sari Anttila tried to combine and implement several growth models and pieces of expert originated knowledge into a single model to be used in the expert system. That turned out to be very frustrating because the tools we used seemed always to lack one or more necessary features. Although we were blessed with the versatility and expressiveness of Common Lisp and KEE[1], we were unable to implement a sufficiently detailed model. This left us with several unanswered questions about both the knowledge representation techniques normally used in such projects and the actual usability of the pine models published in forestry and biological journals.

The other part of the preliminary steps of the expert system development project proceeded much better. Taneli Kolström and I implemented a simple expert system for forest regeneration decision making. The system worked reasonably but not brilliantly. As discussed in [17], the system was

[1] IntelliCorp's registered trademark

a fixed collection of `IF ...THEN ...` rules. Although a simple explanation facility was implemented, the system was not able

- to provide domain specific detailed knowledge for the user's questions, or
- to simulate actual growth after the execution of the chosen regeneration plan.

In other words, the system lacked the detailed model which I and Sari were unable to produce. In addition to that, it was not clear how the system would have utilized the model in explanations or simulations even if the model had existed.

One additional complication to the expert system development project came from the practical expectations of forest researchers. They expected the system to produce *numeric* answers which could be easily interpreted as money, man-hours, growth of bio-mass in kilograms etc. They were not at all happy with the symbolic approaches normally used in expert system development. Unfortunately the initial optimism turned into frustration and a number of unanswered questions. The most important ones were:

A. Is it possible that there are fundamental problems with the symbolic approach because it was so painful to implement forestry knowledge using the symbols?

B. How could we combine different pieces of knowledge and a variety of numeric models into a single model which could be used for simulation? Could there be a single approach to the phenomena or objects modeled? What would that approach be?

C. How could that model be developed in a way that makes it to work well in as many cases as possible?

D. Could we easily combine the symbols used in reasoning and rules with those numeric models?

E. How could we include the knowledge in the numeric models into the explanations generated by the system?

F. The use of the existing simulation models was often much too slow for multiple simulations to be done in a short time — could there be any methods to speed up the use of complicated models?

These questions made me turn from practical expert system development to the fundamental questions of domain modeling. The questions have been, at least partially, answered during the years I have been working with the fundamental aspects of domain modeling. My proposals for such answers are expressed in the articles included, some of which have been published or presented elsewhere and some appear the first time in this collection.

Before discussing my own papers more deeply I will briefly outline what has been done and what have been my answers to the questions. First I will outline how I started to develop a solution for the representational problem.

1.2 Proceeding from the questions

The first step in order to find a way out of the representational lock-up was to find out what had been done earlier in the field. The knowledge representation field has given rise to quite many articles and good books, but the essential question was about the usability of those approaches in the problem of implementing classical simulation knowledge into an expert system.

The first approach to knowledge representation was the classic rule-based approach which was not usable in this case. Unfortunately this outruled most of the excellent work done with systems like MYCIN (see e.g. [2]) and other early expert systems. Therefore the early knowledge representation schemes were not the perfect ones to be used.[2] Because rule-based approaches did not promise solutions, new alternatives were sought after.

The approaches based on the *semantic-networks* principle did not seem to be very promising either. Although semantic networks provide us with many interesting possibilities for reasoning, they do not offer a suitable methodology for representing changing things. Similarly the criticism that propositional approaches (semantic networks, predicate calculus, logic programming) to knowledge representation impose only a local organization on the world [1] did not make the propositional approaches more promising.

One of the most interesting knowledge representation techniques is *frames*. Frames, which have been proposed by Minsky [18], do not, however, promise very much for simulation of continuous systems. Although frames are one of the most powerful knowledge representation techniques, they do not provide ways to simulate rapidly changing dynamic relations and numeric values. Thus, we remain with traditional numeric methods and qualitative modeling.

Traditional numeric methods which are based on differential equations or other types of functions do not easily bend for explanations or representations of causality. This has been found to be problematic by Iwasaki and Salmon who express concern about the subjectivity of causal interpretations of models [16] and who have developed methodologies to achieve a causal ordering of differential equations [15].[3] I decided to follow a different path because I found out that even qualitative models were unsuitable for the modeling tasks I was concerned with (this answered question A), but I will return to the problems of qualitative modeling later because one of the articles explicitly discusses the problems of qualitative modeling.

[2] It is interesting to note, however, that a great deal about explanation and similar basic things have been fairly well covered those days. For example Clancey's articles [6] [7] and book [5] are very interesting.

[3] Although the causal-ordering approach proposed by Iwasaki and Salmon was not present when this research started, it seems the definition of problems is fairly similar. The main difference seems to be that I attempt to remove the problem of causal ordering altogether by making causal structure a fundamental part of models, and Iwasaki and Simon attempt to solve the problem within the existing modeling paradigms.

1.3 Reformulating the questions

After considering the suitability of different modeling approaches and knowledge representation techniques I realized that they did not help me very much. The problem did not seem to be so much a technical one, it was much more a problem of *semantics*. The knowledge representation schemes and modeling techniques should obviously pay very close attention to the *substance* of the modeled domains, and I was not able to find guidelines for the domain I was interested in.[4] This was very unfortunate because I wanted to express the substance of the domain, not to use fancy techniques. After considering this difficulty I realized that Cercone and McCalla [4] had hit the nail on the head by saying:

> 'No data-structure makes sense unless what it means is precisely specified.'

Unfortunately that rather obvious realization did not in any way help me to find a suitable method for creating useful models of trees. After a while I started to search for the *semantics* for modeling and simulation of physical domains.

The semantics of the modeling approach should not be so difficult because the final product should be a representation of the modeled phenomenon or object. The problem arises when we consider the way of achieving and implementing such a representation — the model. In order to find a way to include the domain-specific knowledge in the model in the best possible ways I started to search for suitable approaches to the domain knowledge.

The target domain was a biological system, about which there exists a fairly large amount of scientific knowledge. Therefore it seemed a promising starting point to consider the domain knowledge to be *scientific knowledge* in the sense of professional knowledge used in biology and forestry. The interesting thing seemed, therefore, to be the nature of scientific knowledge.

Many of the writers concerned with scientific knowledge seem to think that scientific knowledge consists mainly of explanations. For example Niiniluoto [19] holds this point of view. From that philosophical discussion and the practical explanation-requirements of possible systems I concluded that the best possible modeling approach should be able to combine the explanatory powers of scientific knowledge and still be able to provide methodology for the implementation of practically useful simulation models.

This returned me more or less to the starting point, i.e. to the basic principles of knowledge representation. Those principles are [4]:

I. Information of many different kinds must be able to be represented, including knowledge about the world, knowledge about the goals and (sometimes) intentions, knowledge of the context and so on.

[4] It is interesting to note that in the field of knowlede acquisition there have been articles which propose that the characteristics of the domain should have a quite sustantial role in the way in which knowledge is acquired. One of the most interesting papers is Nwana's article [20].

II. Knowledge representation is relativistic — the best kind of knowledge representation scheme is often dependent on the particular requirements of the given application.

III. Knowledge should be representable to all depths — there is no absolute level of primitive which cannot be “opened-up”.

IV. The processes which manipulate a knowledge representation scheme are important. Moreover, they should run in a *reasonable* space-time.

V. It is important that a knowledge representation scheme be precisely formulated — ad-hocness is no longer satifactory.

VI. Artificial intelligence approaches to knowledge representation do not have exclusive access to all the answers — other areas of inquiry are extremely useful and have been influential on AI knowledge representation research.

The above principles together with the original questions which surfaced during the forestry expert system development project provided a good approach for the search for possible solutions. In the next section I will briefly outline the answers I have developed.

1.4 Proposing answers

The principal questions of knowledge representation are quite large, and I think that it may not be possible to answer them at once. It is much more promising to divide possible domains into a few groups which have more similar characteristics. One possible group is physical domains.

Because the questions I have researched have oriented from the field of forest simulation, the choice of physical domains was obvious. In order to clarify and restrict the field I have restructured some of the problems and principles according to the approach chosen by me. Due to the restriction of the domain I was able to reformulate the principles in the following way:

I. Information of physical reality must be able to be represented. There is no need to consider other domains and their requirements.

II. The relative nature of knowledge representation is not as straightforward as with the concrete world. The relativisticity is not so remarkable because the prepresentation should strive to be as faithful to the original as possible.

III. The processes used to manipulate knowledge are important because models of physical reality are inherently complex.

IV. The theory behind the representation should be as good as possible.

V. Artificial intelligence approaches do not seem to be able to provide promising schemes, hence other approaches should be sought after.

Because the domain was restricted to physical reality, there were no need to have a knowledge representation scheme which could express anything other than knowledge of the physical reality. In that way the way to provide a solution to question B was much easier to find. If we remember that Hayes [14] required that the formalization used in a model of physical reality should have a common framework for the whole formalization, it is obvious that the structure of the knowledge of physical reality should provide the semantics according to which different models should be conceptualized. In that way the formalization would provide the solutions to question B in addition to being in accordance with the modified principles of knowledge representation. The knowledge to be used in models of physical reality seems to be quite near *scientific knowledge*. The modeling paradigms should not try to force the domain to be according to the paradigm; the domain should determine how we should model it. In other words, if our knowledge of a specific domain is scientific knowledge, then we should use a modeling approach which respects the nature of scientific knowledge.

One of the most promising approaches to the nature of scientific knowledge has been proposed by Salmon [22] [23]. Salmon's approach enables us to avoid the problems encountered with other approaches and allows us to create a quite straightforward implementation of our domain knowledge. The connection between so-called *deep knowledge*[5] and the Salmonian approach should enable the creation of models which are as robust as possible. The depth of knowledge used in a specific model depends on pragmatic aspects, and those aspects have been discussed in *Deep knowledge and domain models*. This answers question C.

Although the relative nature of knowledge representation is a fact, relativism is greatly reduced if the domain is restricted to physical reality only. In that way the reliability of the model is greatly improved. The robustness of models[6] could also be improved by using the best knowledge available, in this case scientific knowledge. Unfortunately the original question D has not, yet, been answered. It has been left for further research.

The connection between explanation and simulation models (question E) has been at least partly answered by incorporating a causal calculus, which was originally been developed in order to enable the formalization of causal explanations, with the concept of causal processes. That connection should enable the creation of explanations from the causal models and their behavior. Such explanations could be used for justifying the simulation results or for education.

The time-space requirements of the processes are important with any knowledge representation scheme and especially important in the case of large simulation models. In the last article of this collection one possible method (which is an answer to question F) of reducing the computational burden is presented.

[5] See the included article *Deep knowledge and domain models* for the definition of deep knowledge.

[6] A model is *robust* if it behaves like its real counterpart would behave in a similar new situation. In other words, a robust model produces right values to the variables used to represent the modeled real-world phenomenon or object.

1.5 The articles

In this section I will briefly outline what the actual articles contain. The order of the articles is according to their contents — chronological order was not a proper order because some articles are contextually successors of the theoretical articles, even if they are a few years older.

The chapters of this book are directly based on the original articles, and the article-format has been retained — only some technical errors have been corrected during the inclusion of the articles into this collection.

1.5.1 On qualitative modeling

In this article[7] I outline why I feel such discomfort with qualitative and symbolic methods when dealing with domains like forest modeling and simulation. This article is a fairly recent one although its contents clearly make it to be the first one.

Some parts of this article seem quite controversial when compared to the second and the third article of this collection. In this paper I propose the use of neural networks instead of symbolic methods, but that proposal is not used in any of the following articles. The main reason is that I have not yet implemented any explanation systems or decision making systems based on the Salmonian models.

This article is mainly based on the realization that qualitative modeling does not support the production of quantitative results at all. In addition to that, the philosophical approach behind qualitative modeling seems to be based on the assumption of the existence of a limited set of logical primitives by which the workings of the system can be more or less easily expressed. The assumption of the existence of such a set of logical primitives is questionable and that makes other features of qualitative models questionable also. If we remember that a qualitative model can be thought to be a simplified version of the differential model, it is easier to see why I do not think that qualitative modeling will provide the answers I have been looking for.

In qualitative modeling the differential equations are replaced by a collection of directions, change-markers and the like. Similarly numbers are simplified to signs, inequalities and orders of magnitude [13]. In other words, qualitative models look surprisingly like higher derivatives of the original differential equation models. Because my research started as an attempt to develop usable and explanatory models of trees, qualitative modeling did not enable the main goals to be fulfilled because forest researchers required *numeric* answers and *explanations* for those numeric values. In addition, the models should be robust. Qualitative modeling did not provide suitable methods because even the robustness of qualitative models can be questioned.

This article proceeds from the critique of qualitative modeling to possible alternatives. The main alternative proposed is the use of scientific knowledge, which is explanatory by nature, in models. The chosen approach to the scientific knowledge is the Salmonian approach which considers the world

[7] Ahonen, J. J. (1994). On Qualitative Modelling. *AI & Society*, 8. 17-28.

to be made of causal processes. This would enable the modeler to follow a uniform and concise view of knowledge throughout the modeling task.

1.5.2 Deep knowledge and domain models

This article[8] is, as surprising as it may sound, one of the oldest ones. The basic ideas proposed in this article originate from 1990. The most important idea in this paper is that I realized that if we use a concise and sensible approach to the structure of the physical world, we are able to develop a uniform approach to structural quantitative domain modeling and simulation.

One of the most important features of models is *robustness*, which could allow us to use a model in much more complicated cases and more freely than we could if the model is not robust. Normally authors concerned with robustness have been speaking about *deep knowledge* which should be the tool to allow us to create robust models. Unfortunately deep knowledge has not been very well defined — in this article I clarify the term. In addition to clarifying the term deep knowledge the connection between explanation and the depth of knowledge is considered. Because explanation has been considered to be a very important feature of domain models, the most sensible way to approach domain knowledge seems to be the explanatory capabilities of that knowledge. Because scientific knowledge is considered to consist of explanations, it is straightforward to think that scientific knowledge should be the foundation for both explanation and robustness.

Although scientific knowledge does provide a natural step for the explanation, its use for simulation-oriented modeling is not as clear. Fortunately Salmon [22] has proposed an approach which uniformly covers the physical reality around us. Although the Salmonian approach is not easy to be formalize exactly [12], it is coherent enough to be used as the semantic background on which models could be based.

The Salmonian approach to the structure of the physical world is conceptually tied to the steps of the modeling process. Those steps have been proposed by Zeigler [24], who divides model development into five conceptual steps. The Salmonian processes can be incorporated fairly easily into Zeigler's steps. In this paper that incorporation is proposed on a conceptual level.

1.5.3 Causal process modeling

This article[9] describes an experimental implementation of the modeling and simulation approach proposed in *Deep knowledge and domain modeling.* In this paper I have avoided the qualitative modeling problems outlined in *On qualitative modeling.*

The article starts with a short introduction to the domain and then proceeds to the definition of interesting types of knowledge. The new definitions differ a bit from the definitions presented in *Deep knowledge and*

[8] Ahonen, J. J. (1995). Deep knowledge and domain models. *Informatica*, 19. 265-279.

[9] This article has not yet been published. Submitted for publication.

domain modeling because Salmon changed his definitions (see [23] for the new definitions) due to criticism[10] presented by Dowe [8]. The redefined concepts are incorporated into implementable approaches to the domain modeling. The formalism is fairly simple and yet expressive enough in order to enable the modeling of complex systems.

A modeling language is derived from the developed modeling formalism and a simple experimental modeling environment is presented. The formalism and the modeling language are tested by developing a simple experimental model of Scotch pine. The experimental model reveals the expressiveness of the developed formalism and the modeling language. Although the experimental model is fairly simple, it does provide surprisingly good simulation results, which supports the claim that models based on Salmonian concepts are robust.

1.5.4 Domain knowledge and explanation

This article[11] is the oldest one. During the time when this article was written I was concerned with the explanation capability of simulation models. During those times it seemed that the representation scheme would be object-oriented and the interpretation would be according to the concept of causal processes. In order to make it possible to produce explanations *mechanically* from the simulation models some type of formalism is required. One possibility is to use logic as the formalism.

Because the forestry simulation models I tried to implement into expert systems did not include any straightforward possibilities for explanation generation I had to rethink the whole matter. The first step was to reconsider the nature of the knowledge used in simulation models. The most promising approach to rethinking knowledge is to consider simulation models to be representations of scientific knowledge. Because scientific knowledge is principally explanatory, it is not difficult to note the similarities between explanation in expert systems and explanation in science.

In forestry most of the simulation models I am familiar with consider biological low-level things to be *processes*, and this was in accordance with the Salmonian theory of causal processes as the structure of the world. Although the Salmonian theory is very useful in the conceptualization of the world, it does not itself provide any mechanism which could be automatized for explanation. Hence more formal approaches were required. In this article the causal calculus developed by Fetzer and Nute [9] [10] and used for explanation by Fetzer [11] has been integrated with the concept of causal processes in order to produce a formalism by which it is possible to produce explanations.

[10] The criticism will not be considered here because its foundations are outside the scope of this collection. I refer to the original papers and books.

[11] This article was refereed and accepted to *Bar-Ilan Symposium on the Foundations of Artificial Intelligence*, June 1991. The article was presented in that symposium. Unfortunately no official proceedings appeared.

1.5.5 Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents

This article[12] is the most practical of these papers. Although it is practically based on principles which are surprisingly near the theoretical principles which have been developed in other papers, it is the second oldest of these articles.

During the years 1990 and 1991 Professor Hannu Saarenmaa and I did some work together because our common interest was in the performance of complex simulation models of trees. During those days we used an object-oriented approach, mainly because Prof. Saarenmaa and I both believed that our approach could provide us with the means to specify complex and robust simulation models. Later our opinions diverged — I started to develop a new approach to representing and using domain knowledge and Prof. Saarenmaa started to use the approach used in this paper more rigorously. We agreed to disagree on my semantic concerns and the fundamental ideas behind the knowledge representation approach used for modeling and simulation. The disagreement was very understandable because Prof. Saarenmaa had much more pragmatic goals for the use of knowlege representation and simulation schemes. The direction he and his group has been following can be clearly seen from the article written by Salminen et al. [21]. I am not, however, content with those approaches because I feel that our original approach did not pay enough attention to the semantics of the representation, and semantics is one of the most important points in knowledge representation [4].

In this article we were concerned with the real-world performance of hierarchical simulation models. The obvious problem with such models is that they tend to expand very fast. Hence run-time requirements are very important and something must be done. One of the simplest ways to achieve performance improvements is to use previously computed values to approximate new simulation results. The discussed method uses previous simulation results in order to be able to drop simulation and use the approximation instead. According to our results, the approach really worked, but unfortunately we did not perform enough experiments with multiple domains in order to be able to present real statistics of the performance improvement achieved.

Unfortunately the future trends proposed in the paper did not materialize because the disagreement about semantic specifications broke our cooperation. This did not, however, produce any specific problems because Prof. Saarenmaa and his colleagues have followed the proposed approach and achieved interesting results.[13] I do not, however, agree with a modeling philosophy which uses different techniques without explicitly specifying the

[12] Ahonen, J. J. and Saarenmaa, H. (1991). Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents. In: *Computer Science for Environmental Protection*, 6th Symposium, München, December 1991, Proceedings. Springer-Verlag, Berlin. 193-200.

[13] Those results will not be discussed here. Interested readers should contact Prof. Saarenmaa directly.

semantics of the features of the approach and developed models.

References

[1] Bobrow, D. (1975). Dimensions of Understanding. In: Bobrow, D. and Collins, A. (ed). (1975). *Representation and Understanding.* Academic Press, New York.

[2] Buchanan, B. G. and Shortliffe, E. H. (1985). *Rule-Based Expert Systems.* Addison-Wesley, Reading.

[3] Cercone, N. and McCalla, G. (ed). (1987). *The Knowledge Frontier.* Springer-Verlag, London.

[4] Cercone, N. and McCalla, G. (1987). What is knowledge representation?. In: [3]. 1-43.

[5] Clancey, W.J. (1987). *Knowledge-Based Tutoring: The GUIDON Program.* The MIT Press, London.

[6] Clancey, W.J. (1983). The Epistemology of a Rule-Based Expert System – a Framework for Explanation, *Artificial Intelligence.* 20. 215-151.

[7] Clancey, W.J., Hashling, D.W. and Rennels, G. (1984). Strategic Explanation for a Diagnostic Consultation System, *International Journal of Man-Machine Studies*, 20. 3-13.

[8] Dowe, P. (1992). Wesley Salmon's Process Theory of Causality and the Conserved Quantity Theory. *Philosophy of Science*, 59. 195-216.

[9] Fetzer, J. H. & Nute, D. E. (1979). Syntax, Semantics, and Ontology: A Probabilistic Causal Calculus. *Synthese*, 40. 453-495.

[10] Fetzer, J. H. and Nute, D. E. (1980). A Probabilistic Causal Calculus: Conflicting Conceptions. *Synthese*, 44. 241-246.

[11] Fetzer, J. H. (1981). *Scientific Knowledge.* D Reidel Publishing Company, Boston.

[12] Fetzer, J. H. (1987). Critical Notice: Wesley Salmon's Scientific Explanation and the Causal Structure of the World, *Philosophy of Science*, 54. 597-610.

[13] Forbus, K. D. (1990). Qualitative Physics: Past, Present, and Future. In: Weld, D. S. and DeKleer, J. (1990). *Qualitative Reasoning about Physical Systems.* Morgan Kaufman, San Mateo. 11-39

[14] Hayes, P. J. (1979). The Second Naive Physics Manifesto. In Michie, D. (ed) *Expert Systems in the Micro-Electronic Age.* Edinburgh University Press. Edinburgh.

[15] Iwasaki, Y. and Simon, H. A. (1994). Causality and model abstraction. *Artificial Intelligence*, 67. 143-194.

[16] Iwasaki, Y. and Simon, H. A. (1993). Retrospective on “Causality in device Behavior”. *Artificial Intelligence*, 59. 141-146.

[17] Kolström, T. and Ahonen, J. (1991). Rule-based reasoning in the selection of forest regeneration methods. In: *Current Advances in the Use of Computers in Forest Research — Proc. of the IUFRO Workshop February 14, 1991*. Bulletins of the FFRI Vol. 395. 36-41.

[18] Minsky, M. A Framework for Representing Knowledge. In: Haugeland, J. (ed.) (1981). *Mind Design*. MIT Press. Cambridge.

[19] Niiniluoto, I. (1983). *Scientific Reasoning and Explanation* (in Finnish). Otava. Keuruu.

[20] Nwana, H. S. (1994). Domain-driven knowledge modeling for knowledge acquisition. *Knowledge Acquisition*, 6. 243-270.

[21] Salminen, H. et al. (1994). Modelling Trees using an Object-Oriented Scheme. *Mathematical and Computer Modeling*. To appear.

[22] Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton University Press. Princeton.

[23] Salmon, W. C. (1994). Causality Without Counterfactuals. *Philosophy of Science*, 61. 297-312.

[24] Zeigler, B. P. (1976). *Theory of Modeling and Simulation*. Wiley. New York.

Chapter 2

On Qualitative Modeling

Jarmo J. Ahonen
Lappeenranta University of Technology, P.O.Box 20, SF-53851 Lappeenranta, Finland

Abstract

Fundamental assumptions behind qualitative modeling are critically considered, and some inherent problems in that modeling approach are outlined. The problems outlined are due to the assumption that a sufficient set of symbols representing the fundamental features of the physical world exists. That assumption causes serious problems when modeling continuous systems. An alternative for intelligent system building for cases not suitable for qualitative modeling is proposed. The proposed alternative combines neural networks and quantitative modeling.

Keywords: expert systems, modeling, qualitative modeling
Running title: On qualitative modeling

2.1 Introduction

In early artificial intelligence (AI) research the approach was mainly based on the philosophical assumption that it is possible to reduce all necessary information into a set of basic elements which could be used by a computational mind. Those elements could be called logical primitives in the philosophical tradition, and probably their clearest definition was done by Wittgenstein in his *Tractatus Logico-Philosophicus* (1971; Finnish translation). In the AI tradition this approach has been explicitly expressed by Newell and Simon (1981) who stressed the idea that a physical symbol system has the necessary and sufficient means for intelligent action. That idea may be called *The Physical Symbol System Hypothesis*, from which a large part of the AI approach seems to originate.

Although the preference of a finite set of basic elements has not been very clear during recent years, it seems to be a fundamental part of the thinking behind AI techniques and approaches, and to be the cause of the introduction of similar thinking into other sciences influenced by AI. One such field is modeling and simulation into which AI originated techniques have been appearing lately and having more or less clear effects (There are several recent publications about the connection between simulation and AI, e.g. Widman, Loparo and Nielsen (1989), Miller et al (1992), and Fishwick (1992) — some of these papers have been published in a special issue of *ACM Transactions of Modeling and Computer Simulation*). In this paper I intend to show that an AI approach to modeling and simulation, qualitative modeling, which has aroused great interest in the last fifteen years, is, in fact, a descendant of the symbolic approach to artificial intelligence. This background is the cause of several problems associated with qualitative modeling.

The importance of qualitative techniques for AI research has been clearly expressed by Clancey (1992). He states:

> ...that *qualitative process modeling* is a good way of characterizing AI programming for scientists and engineers; providing a useful pedagogical answer to the question, "What constitutes an AI program?".

The importance of qualitative techniques in AI programming means we should not be surprised that the qualitative approach has accompanied other AI techniques to fields with which AI comes into contact.

The interaction between the simulation community and the AI community is not surprising because simulation and AI have common interests although that was not so clear in the early days of AI because at that time AI was most concerned with reasoning. It now seems to be an accepted paradigm in artificial intelligence research that in order to produce practical expert systems[1] we have to turn our attention to representations of the physical world. (Note that in this paper I will consider expert system research to be a part of AI research — a point of view to which a few AI or expert system practitioners may object, but which will help me to outline my point of view.) This has originally been proposed by Hayes who proposed the use of models of physical reality as the method of avoiding the problem of too simple and simplified problem domains (Hayes 1979). Hayes proposed that everyday knowledge of the physical world should be formalized. That means, of course, that a model of physical reality should be created. Even if we are not very optimistic about the possibility of a generally useful formalization of the world around us in the Hayesian sense, the thinking behind that approach is worth considering, especially because it seems to hide an approach which is different from the normal thinking used in modeling and simulation (as seen in fields other than AI). In addition, it is worth noting that Hayes proposes the use of knowledge engineering techniques used in expert system development as the method of model creation. This

[1] Expert systems are programs which are intended to replace human specialists in some fields. In principle such systems are collections of knowledge from the field in question and reasoning technique to enable the system to produce answers which should, in principle, be similar to answers which the human expert would have produced.

may produce several problems in the creation of those models, as will be seen later.

Note that according to Clancey (1992) all expert systems use models; all expert systems are model-based. It is especially interesting that this way of thinking seems to share the original philosophical basis of symbolic AI research. In the following section I will briefly outline the traditional modeling view in order to make the philosophical difference between AI modeling and traditional modeling clearer.

In order to make the following considerations clearer I will define several concepts, some of which have been already used in the meaning given. In this paper the term *physical reality* means the concrete world. Similarly, a *real-world phenomenon* or a *real-world object* is a phenomenon or an object which exists or can potentially exist in the physical world. A *model* is a representation of the physical reality. Such representations are simplifications of the physical reality, and unfortunately they will not produce reliable information on every aspect of the phenomenon being modeled (Lewis and Smith, 1979, 2). Our definition of models differs from the normal definition in its clear nature as a representation of the physical reality (see e.g. (Futo and Gergely 1990) for the usual approach). *Simulation* is the creation and execution of dynamic models employed for understanding system behavior (Fishwick, 1992). These definitions restrict the following considerations on the modeling and simulation of physical reality, in which case theories of artificial and pure theory building are left out.

Before considering qualitative modeling in greater detail I will briefly discuss a more common approach to modeling and simulation.

2.2 The traditional view of modeling

In order to emphasize the distinction between the modeling thinking in natural sciences like forestry and AI, I use the term *traditional* when I refer to the modeling philosophy existing in sciences like forestry. Although forestry includes many fields which cannot be considered to be natural sciences like chemistry or physics, the term forestry will be used to cover the field of more or less biologically-oriented research of trees and their behavior.[2] Such research normally provides models according to which the growth of forests (or individual trees) are simulated in order to forecast actual forest growth. That research includes the behavior of trees, the effect of pests on trees etc. Since that part of forestry is close to biology it is reasonable to say that it is a natural science.

One possibility of examining the thinking behind the traditional approach to modeling is to consider that by approaching the model building process by outlining the course of such modeling projects. One way of outlining the traditional approach to the modeling task is presented by Widman and Loparo (1989). They summarize Zeigler's work (1976) to the following levels:

[2] The considerations presented in this article have originally been developed during various modeling and simulation projects in the field of forestry. Hence the use of forestry as an example.

- the “real system”, a source of potentially observable data;
- the “experimental frames”, a set of limited observation or manipulation cases for the real system;
- the “base model”, a comprehensive model of the system in every experimental frame;
- the “lumped model”, a simplified version of the base model, which is simplified in a way that still provides the reliability of the original base model in interesting cases.
- the “computer model”, an implementation of the lumped model in a computer programming language on a machine.

The above steps of traditional modeling include the assumption that there is no model which is *complete*, i.e. so thorough a replicate of the modeled object or phenomenon that it could not be improved due to changes in knowledge about original or experienced defects in the developed model. Although traditional modeling intends to develop robust[3] models, it acknowledges the fact that due to the incompleteness of human knowledge new models may be more robust than earlier models, but real robustness can never be achieved. This is clearly expressed by Zeigler (1976, 31) who concludes that:

> In any realistic modeling and simulation area, *the base model description can never be fully known.* (original emphasis)

The lumped model then includes the chosen features of that imperfect base model which will be used for understanding the modeling system in the most interesting cases.

In other words, the traditional modeling approach hopes that the developed model is a more or less good enough approximation of the reality (see e.g. Thompson (1989) and Lewis and Smith (1979) for practically oriented examples of the traditional modeling). This type of approach requires greater and greater complexity of the model in order to overcome the inherent shortcomings of any model, namely their incompleteness resulting from the imperfect knowledge of the real system in question. In some cases it seems to be possible to produce robust enough models which include so much knowledge that the incompleteness of that knowledge has been, at least partially, overcome.

2.3 Knowledge acquisition and domain modeling in expert systems

The origins of qualitative modeling are in conventional AI research and partly in expert system research. In order to understand the thinking behind

[3] A model is *robust* if it behaves like its real counterpart would behave in a similar new situation. In other words, a robust model produces right values to the variables used to represent the modeled real-world phenomenon or object.

qualitative modeling and simulation in expert systems we have to briefly consider expert system development and how domain specific knowledge is normally introduced into the expert system program (or into any other AI program). This is especially important because Hayes (1979) proposed the use of knowledge engineering as the method for the creation of models of the physical world. Knowledge engineering is essentially the same process as the traditional expert system development process.

The traditional expert system development involves at least two persons, namely the domain expert and the knowledge engineer. Traditionally the expert system development could have been expressed as the following cycle (note that there may be more than one engineer and more than one expert):

- The knowledge engineer interviews the domain expert in order to get useful knowledge.
- The knowledge engineer writes the knowledge into the expert system.
- The expert and the engineer evaluate the system, and if they decide that the system is good enough, then the process stops, and if they do not feel that the system is good enough, then the process is iterated to the first step.

The inherent problem of traditional knowledge acquisition is clear from the expert system development cycle. From the cycle we should note that the original knowledge provided by the expert to the expert system goes through the knowledge engineer and from that we realize how important the role of knowledge engineer is to the development of the system. If the expert sees an error in the system, he/she has to tell the engineer that the system has to be changed. Hence one part of the problem does not lie in the expert system technology, but in the communication between the expert and the engineer. This obviously has practical consequences.

When a human expert is interviewed by a knowledge engineer, the interview is in fact a two-way process. At the same time as the expert tells the engineer what he/she considers to be the most important features of the object or phenomenon being modeled, the knowledge engineer can be argued to transfer his/her way of viewing knowledge engineering and model building to the expert. In other words, it is only natural that the domain expert starts to modify his/her answers to the questions asked by the engineer in a way that is gradually more similar to the engineer's view of domain knowledge. There is no way to be sure that the modified way of composing answers to the engineer's questions is in accordance with the way in which the expert would explain the same domain to other experts in his/her field. This type of problem is especially apparent in cases in which the knowledge engineer introduces modeling concepts that are alien to the domain expert. Such alien concepts may well be frames, object oriented programming, logic as used in AI, a reasoning based approach to knowledge usage etc. One of the most significant things the engineer may introduce to the domain expert is the AI research philosophy, and when we remember that the original AI research (and obviously a great part of current AI research also) was based on the assumption that a finite set of logical primitives (or other types of symbolic elements) exists, we have a clue to the birth of qualitative modeling.

Considering the adaptability of the domain expert's methods to express his/her knowledge of the domain and the AI research tradition, it is not surprising that AI researchers have combined the AI tradition and simplified domain knowledge. It may be said that qualitative modeling is an obvious combination of the AI philosophy and simplified domain knowledge originating from domain experts or textbooks (by textbooks I refer to simplified books used to teach e.g. physics to novices, i.e. books in which knowledge is used in a simplified way in order to make it comprehensible to non-experts).

Since I believe that qualitative modeling may be a fundamentally restricted or flawed approach to domain modeling (at least in some cases), I will outline those features which make me consider it inadequate for at least some modeling tasks.

2.4 Qualitative modeling and qualitative simulation

Despite the fact that qualitative modeling and qualitative simulation have been amongst the main interests in AI research during recent years, it has been very difficult to develop a concise picture of the field. This has, however, changed because nowadays there are good collections of papers on qualitative research, see e.g. Weld and de Kleer (1990). Unfortunately qualitative modeling and qualitative simulation articles very rarely make their under-currents clear. In order to understand the thinking behind technical considerations we have to briefly consider technical aspects as well.

One of the most illustrative perspectives to qualitative modeling and simulation has been written by Kuipers, who is one of the rare writers who has made the connection between qualitative models and differential equations explicit. In his paper (Kuipers 1985) he outlines a system for qualitative simulation, which he claims is able to produce every actual behavior of the modeled system.[4] Although the QSIM system outlined by Kuipers is, obviously, fairly efficient in modeling a *mechanism*, it is quite important to realize that QSIM leaves some questions open.

In the QSIM examples, and in other qualitative simulation systems, the actual behavior of the modeled mechanism is considered to be modeled by a very simplified artificial system representing the physical system. In that artificial system the functions representing the changes of the values of different parameters of the system are reduced into changing directions and landmarks. Landmarks are used to represent significant values of those parameters and the borders of the legal values which those parameters can attain. This is both the strength and the weakness of qualitative simulation. For those cases in which there is a limited, clearly distinctive set of *possible states* of the system, qualitative modeling can fairly reasonably provide a tool by which those systems may be modeled without being forced to compute exact values for every parameter used to represent the system in question. Such qualitative models can be seen as extremely simplified

[4] It could be argued that a model which is able to produce *every* actual behavior of the modeled system is, in fact, robust. This is a very different concept than the realistic assumption of traditional modeling philosophy.

traditional models of the actual system. In a qualitative model the collection of numeric parameters and the differential equations representing the relations between the values of those parameters are replaced by a similar collection of parameters, which do not have exact values now but only landmark values (which could be *full* and *empty* for a bucket), and the differential equations are replaced by some kind of derived forms which will then be used to determine the changes, i.e. the direction of change, and possible reached landmark values of the parameters. The new equations derived from the original differential equations do not represent the method how the value for a parameter is computed, they represent the direction of change which could occur to the parameter.

Lately there have been some interesting considerations presented in the connection of qualitative models, and those considerations require more attention. For example Clancey (1992) says that a qualitative model definition is, in fact, a qualitative graph definition and that possible defects in the qualitative domain theory can be found by doing a complete search for every possible combination. The claim is very interesting: obviously this means, according to Clancey, that the checking of the domain theory of any system using qualitative models can be done by producing a state-transition graph of all possible states for the model. The problem with that kind of system is that processes continuously interacting with each other may themselves be continuous. A complete, or even a reasonably covering, search of different combinations could easily turn out to be a practical impossibility.

The concept of a qualitative graph draws attention to a feature of qualitative models which seems not to be considered with enough seriousness. From Clancey's discovery that qualitative models are, in a sense, graphs which represent the modeled object or phenomenon, it is possible to reconsider the philosophical foundation which makes the loss of accuracy acceptable. The graph-likeness reveals the underlying assumption of a finite set of basic elements. Because the sufficient set of elementary symbols is finite, there is no reason to worry about the lost accuracy. From a finite set of basic elements it is, of course, fairly easy to generate an ideal graph from which every possible state of the system can be derived. Note that qualitative simulation (i.e. simulation done by using qualitative models) produces similarly finite graphs with clearly distinctive nodes. This should enable the system to derive numerical answers fairly easily from those states.. The change from qualitative representations to quantitative ones should be relatively easy but that change has turned out to be much more difficult than one would expect. This is clearly expressed by Kuipers (1993b) who lists the derivation of quantitative problems from qualitative ones as one of the most important unanswered questions. This is, however, very surprising if we assume, as Kuipers (1985) does, that qualitative models cover *every* actual behavior of the modeled system.

Although qualitative models and quantitative models are both models in every sense of the term, I feel that in addition to the mathematical differences there is a very fundamental difference in thinking behind the approaches. That difference may not exhibit itself in many cases, but in other cases it may have a very important role. Examples of cases where this difference has practical meaning will be discussed later, at this point I shall concentrate on briefly describing the difference.

Consider that a qualitative model is provided as a large piece of paper on which the qualitative graph of the model has been printed. Clearly it is the case that with enough patience it is possible to track down the *exact* state of the modeled system after a certain amount of time. But with quantitative models which include probabilistic aspects that may not be possible. Quantitative models of continuous physical systems provide an infinitely large number of different states (of which there can be no state-transition graph). In the following section I will outline a few reasons which cause some concern.

2.5 Problems with qualitative modeling

Normally the qualitative modeling techniques have been used in connection with various mechanical devices. Although it could be fairly comfortably argued that those devices have too often been everything but seriously complex (as an example of the simplest cases I refer to the string example used by Kuipers (1985)). Despite the fact that qualitative systems large enough to be practically useful seem to be very rare, there is no reason to claim that qualitative modeling may not be a useful technique in technical and other established fields. But the problematic fields seem to be those which introduce more uncertainty to the domain.

In qualitative reasoning literature, like the collection edited by Weld and de Kleer (1990), there is *not even one* example of qualitative simulation applied to natural natural systems, like trees, which are continuous and in which processes may be in continuous interaction and form constant feedback loops, for example, the flow of nutrients from roots to the upper parts of the tree, and sugar etc produced in leaves to the roots (or processes which result from external stimuli, e.g. a moose may eat a part of the crown of a young tree and cause a feedback loop to occur). Similarly even some qualitative modeling researchers have expressed serious reservations on the general applicability of qualitative modeling, see e.g. Sachs (1987) who considers her own technique to be useful for man-made devices only. This is not comforting, it raises questions about the usefulness of qualitative simulation in nontechnical fields. Or could it be that qualitative models cannot be built in those cases because such basic elements (or good imitations of them) as required by the philosophical foundations of qualitative modeling cannot be found?

Because qualitative modeling requires, according to Clancey, the possibility of producing a complete graph of the different states of the modeled system (qualitative simulation can, also, be thought to be a graph with a finite set of nodes representing *every* possible starting state and *every* possible simulation result can be achieved simply by traveling through the graph), I claim that qualitative simulation provides, in fact, *analytical* solutions to problems[5], not simulation solutions in the same sense as simulation is con-

[5] It is worth noting that Kuipers (1993a) indirectly admits the analytical nature of qualitative modeling by stating that a good general purpose algebraic manipulation utility may be

sidered in the traditional simulation and even in early artificial intelligence literature (see e.g. Sayre (1965)) in which a clear philosophical distinction between solutions produced by analytical means and solutions provided by simulation is made. *Simulation* inherently includes *probabilistic* and *continuous* parts, which makes it impossible to produce similar "state-space" graphs for simulation as is for qualitative simulations.

As an example of a modeling task in which qualitative modeling encounters serious problems we may consider a real project from the field of forestry. The problem with normal AI and expert system methodologies is clearly outlined by a discussion of the usability of an expert system developed by using logic programming and qualitative modeling techniques. That system, developed by Saarenmaa et al (1991) was designed to cope with operational forest management, but the system did not succeed in giving any practically useful results. As an example of the embarrassing results of the application of AI methodologies and implicit philosophies to a fundamentally different domain we may consider a very annoying feature of that system. Saarenmaa et al state:

> '...This led to such surprising rules as "artificial regeneration never fails". This is not simplistic at all, but only a way to re-organize knowledge so that it can be made functional.'

in an attempt justify a feature which has made the authors themselves unsure. The problem with the rule is that even artificial regeneration can fail with varying degrees and varying probabilities, and the future behavior of the modeled system after such failure is composed of continuously varying combinations of different actions, and that it turned out to be impossible to identify different variations as single symbolic names and unambiguous directions (Saarenmaa, in personal communication).

The surprising rule of the never failing artificial regeneration did not, however, encourage the authors to try to develop the system further, they seem to have, in fact, dropped the method and turned their attention from the qualitative techniques to the incorporation of quantitative models in the decision support systems. They even say, in that same article, that

> '...it is very simplistic to base model integration on an inference method only'

which represents their view according to which the qualitative techniques are purely reasoning techniques, not simulation techniques. An especially problematic feature of the qualitative techniques was their philosophical basis, which would have required a philosophical change and which was felt alien. It is interesting to note that de Kleer (1993) complains about a very similar case, namely that physicists did not see any point in qualitative modeling, but that should not be surprising. According to Hayes (1979), he was very shocked at the age of eleven when he was taught Newtonian mechanics which was not in accordance with his previous conceptualization of the physical world. From this background it is obvious that physicists,

able to evaluate models and produce new equations to be used for simulation. This clearly implies the analytical nature of qualitative models.

who are experts, would not use the naive conceptualization of the physical world then talking to each other, they would use their field-specific working knowledge e.g. Newtonian physics. Because physicists and other experts would use their field specific concepts when talking to each other and simplified or even wrong (in the sense of the specific knowledge which makes them experts) conceptualizations when explaining the same things to novices like knowledge engineers, it really is not a surprise that such experts (like the physicists in de Kleer's example) would see no sensible reason for qualitative models which are like simplified versions of their own domain knowledge.

2.6 An alternative to qualitative techniques

Although the qualitative modeling approach and the physical symbol system hypothesis have been dominant in domain modeling (in expert system research), it is not clear that qualitative techniques really are the only way to produce practically useful expert systems. I believe that it is possible to develop expert systems which include quantitative models and provide an alternative to the qualitative modeling based reasoning approach.

An alternative could be to develop quantitative models in a way which uses a uniform way of thought and connect such quantitative models to properly taught neural networks. The method for the uniform thinking for domain models could be something like the causal process approach to the structure of the reality proposed by Salmon (1984). This method could be able to cover the original requirements proposed by Hayes, who said that the formalization of the physical world should have the following characteristics (1979):

- Thoroughness, i.e. it should be able to cover the whole range of everyday phenomena (the formalization will not, of course, be perfectly thorough).

- Fidelity, i.e. it should be reasonably detailed.

- Density, i.e. the ratio of facts to concepts should be high.

- Uniformity, i.e. there should be a common framework for the whole formalization.

I believe it possible to incorporate causal and structural knowledge into quantitative models. In the development of quantitative models it should be concentrated on achieving as much accuracy as possible in both the results and the structure of the model. Unfortunately it often seems to be the case that the use of accurate models does require so much computing power that those models may not be practically useful. This problem can, however, be avoided (at least partially) as discussed later.

If we intend to use a model for multiple simulations, it is a safe assumption that in many cases the results may vary only a little. In such cases it might be possible to combine quantitative models, which include structural and causal knowledge of the domain, with statistical methods in order to improve the overall performance of the system. In such combinations the quantitative model would be used when a novel situation is encountered (i.e. a situation which has not yet been simulated by using the current model) and cases which have a very near resemblance with previously simulated cases could be approximated by using statistical methods based on previously generated simulation results. In that way it would be possible to make actual simulations by using quantitative models, and let the performance improvements be done by numeric methods based on previous simulation results. One possibility for such a method has been proposed by Ahonen and Saarenmaa (1991).

In order to be able to provide the required intelligence for, for example, expert systems, an alternative to qualitative models must be found. I think that there is no reason why the output of quantitative models (which should be developed according to the structural considerations) could not be used for providing the simulation results to pre-taught neural networks. The neural network would provide the users with the required “intelligent” answers.

In order to avoid arguments saying that neural networks alone are enough, I will outline why I disagree with such arguments. Since one of the main uses of modeling and simulation is to help us to understand the modeled system and make certain that we really understand it, it would be much more probable that we would identify the proper parameters used as the input (for a neural network) if they are produced by well-designed models. Without using models to provide the neural network with important information, there would be no guarantee that the information would be essential for decision making. Hence I believe that the best results could be achieved by combining quantitative models and neural networks.

2.7 Discussion

In AI research it has been customary to bury methodological and philosophical aspects in various technical considerations and the development of simple working systems (It is interesting to note that Hayes complained about this trend more than a decade ago. Has nothing changed?). That habit may have a problematic side-effect, namely that it may introduce qualitative modeling philosophy to fields for which it is not suitable and/or alienate potential users from AI techniques. For example, in forest simulation there is often an explicit need to produce numeric estimates, and the symbolic nature of qualitative techniques makes it doubtful in the eyes of forestry researchers. This type of doubt is very unfortunate because in many fields of traditional numeric simulations (e.g. forestry) there is a clear need for at least some paradigms originating from expert system research — object-oriented programming and conceptual modeling, just to mention some examples.

If it is true that there are systems which cannot be adequately modeled by qualitative modeling, the philosophical justification of the qualitative simulation paradigm is questionable. It even may be possible that qualitative simulation has been used as an escape route from the threatening computational requirements of real-world models — an escape route which provides a convenient excuse to return back to symbolic representations under the misconception that everything can be reduced into fundamental, atomic structures which can fairly easily be implemented as a computer program. This is very discomforting because the philosophers have not been able to provide such fundamental notions thus far.[6] Could it really be possible that the AI community will succeed in something in which others have not succeeded?

The success of AI research in providing the basic elements required for accurate symbolic systems is not very probable considering the efforts of philosophers who have attempted to define even one logical primitive. The lack of success of the qualitative approach has even been disappointedly pointed out by de Kleer (1993) who complains that the advances required have not been made and that research should still strive to fulfill the central point of the more than a decade old naive physics manifesto of Hayes (1979). This disappointment should not, however, be a great surprise because qualitative techniques still suppose the existence of the set of basic elements to be used as the parts of the qualitative models. It may even be said that the problems encountered by qualitative modeling could be considered to be a constructive proof against the philosophical assumption about the existence of such basic elements.

The existence of possible fundamental problems in the qualitative approach and cases which are not suitable for the qualitative approach does not mean that the qualitative approach is not a useful approach for many cases. In technical fields (and probably in many other fields also) qualitative techniques have proved their practical usability (see e.g. the articles in Weld and de Kleer (1990)). But for the cases which are not suitable for the qualitative approach new alternatives are required. I believe that one alternative is to incorporate causal and structural knowledge into quantitative models and feed the quantitative simulation results into neural networks. This should enable us to avoid the problems of the restricted nature of the quantitative approach and still be able to build systems which could be used as expert systems.

References

Ahonen, J. J. and Saarenmaa, H. (1991). Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents. In *Proc. 6th Symposium on Computer Science for Environmental Protection.* 193-200.

Clancey, W. J. (1992). Model Construction Operators. *Artificial Intelligence*, 53. 1-115.

de Kleer, J. (1993). A view on qualitative physics. *Artificial Intelligence*, 59. 105-114.

[6] I will not, however, discuss this in great detail because I think that it is better if I leave that discussion to those who are better educated in these matters.

Fishwick, P. A. (1992). An Integrated Approach to System Modeling Using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. *ACM Transactions on Modeling and Computer Simulation*, 2. 307-330.

Futo, I. and Gergely, T. (1990). *Artificial Intelligence in Simulation*. Ellis Horwood. Chishester.

Hayes, P. J. (1979). The Naive Physics Manifesto. In Michie, D. (ed) *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press. Edinburgh.

Kuipers, B. J. (1993a). Reasoning with qualitative models. *Artificial Intelligence*, 59. 125-132.

Kuipers, B. J. (1993b). Qualitative simulation then and now. *Artificial Intelligence*, 59. 133-140.

Kuipers, B. J. (1985). The Limits of Qualitative Simulation. In: *Proceedings of International Joint Conference on Artificial Intelligence*. 128-136.

Lewis, T. G. and Smith, B. J. (1979). *Computer Principles of Modeling and Simulation*. Houghton Mifflin. Boston.

Miller, D. P., Firby, R. J., Fishwick, P. A., Franke, D. W. and Rothenberg, J. (1992). AI: What Simulationists Really Need To Know. *ACM Transactions on Modeling and Computer Simulation*, 2. 269-284.

Newell, A. and Simon, H. (1981). Computer Science as Empirical Inquiry: Symbols and Search. Reprinted in: Haugeland J. (ed) (1981). *Mind Design*. MIT Press. Cambridge.

Saarenmaa, H., Kaila, E., Nuutinen, T. and Kolström, T. (1991). Operational forest management planning with logic programming. In: *Current Advances in the Use of Computers in Forest Research — Proc. of the IUFRO Workshop February 14, 1991*. Bulletins of the FFRI Vol. 395. 61-68.

Sachs, E. (1987). Piecewise Linear Reasoning. In: *Proceedings of AAAI-87*. Seattle (WA). 655-659.

Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton University Press. Princeton.

Sayre, K. M. (1965). *Recognition: a Study in the Philosophy of Artificial Intelligence*. University of Notre Dame Press. Notre Dame (Indiana).

Thompson, J. R. (1989). *Empirical Model Building*. John Wiley & Sons. New York.

Weld, D. S. and de Kleer, J. (eds) (1990). *Qualitative Reasoning about Physical Systems*. Morgan Kaufmann Publishers, San Mateo, California.

Widman, L. E. and Loparo, K. A. (1989). Artificial Intelligence, Simulation and Modeling: A Critical survey. In Widman, Loparo and Nielsen (1989). 1-44.

Widman, L. E., Loparo, K. A. and Nielsen, N. R. (1989). *Artificial Intelligence, Simulation & Modeling.* John Wiley & Sons. New York.

Wittgenstein, L. (1971). *Tractatus Logico-Philosophicus* (Finnish translation). WSOY. Keuruu.

Zeigler, B. P. (1976). *Theory of Modeling and Simulation.* Wiley. New York.

Chapter 3

Deep Knowledge and Domain Models

Jarmo J. Ahonen
Lappeenranta University of Technology, P.O.Box 20, 53851 Lappeenranta, Finland

Abstract

An approach to the concept of deep knowledge is outlined. The approach is based on the assumption that the depth of knowledge results from its explanatory powers. After considering some examples of deep and shallow knowledge and defining deep knowledge and robustness, an approach to the development of domain models based on deep knowledge is proposed. The proposed approach is based on the Salmonian concept of causal processes and it provides a uniform point of view to knowledge of physical domains and domain modeling. The approach is developed in order to incorporate structural and causal knowledge directly into numeric models because qualitative approaches seem to have philosophical problems.

Keywords: expert systems, modeling, qualitative modeling

3.1 Introduction

Since the Chandrasekaran and Mittal article (Chandrasekaran and Mittal, 1983) most writers concerned with so-called domain models have been using the term *deep knowledge*. Unfortunately, the term has not been defined very well and its use has depended on vague notations of its nature. The vagueness of the meaning of the term invites us to try to define it more clearly, and one approach to the concept of deep knowledge and its nature and use in domain modeling will be discussed later in this paper. The approach is developed because qualitative techniques seem to have some philosophical problems (Ahonen, 1994) and there obviously is a need for the incorporation of the best features of quantitative and qualitative approaches. In addition to this some of the most obvious consequences of the use of deep knowledge for the overall usefulness and structure of domain models will be discussed.

In fault diagnosis and simulation there is one obvious requirement for the domain model, namely that it should be able to function like the modeled phenomenon in every case. Unfortunately, how this can be achieved is not entirely clear, despite some results and authors speaking in favor of deep knowledge. The main problem with deep knowledge is that we do not know what it is although we know what it should do, i.e. provide better robustness for domain models.

The use of the term deep knowledge seems to be closely related to the idea that we cannot create practical AI systems without turning our attention to physical reality around us. Hayes proposed the use of models of the physical reality as a method of avoiding the problem of too simplistic problem domains (Hayes, 1979). The formalization of the physical world and models created from the physical world should, according to Hayes, have the following characteristics (Hayes, 1979):

- Thoroughness, i.e. it should be able to cover the whole range of everyday phenomena (the formalization will not, of course, be perfectly thorough).

- Fidelity, i.e. it should be reasonably detailed.

- Density, i.e. the ratio of facts to concepts should be high.

- Uniformity, i.e. there should be a common framework for the whole formalization.

In addition to the interest in physical domains, Hayes said that reasoning alone is not enough, from which it may be concluded that an intelligent system would include both reasoning and knowledge of physical reality. This dual view leads us to adopt the idea of *model based reasoning*, according to which the descriptions of domain and reasoning are kept separate.[1] The model based reasoning approach is more a methodology than a technique, because it does not determine the implementation (Saarenmaa, 1988). The model based approach seems to be especially useful in fault diagnosis and other fields that require the use of real-world models (see, e.g., de Kleer (1987), Koton (1985), Nardi and Simons (1986), Xiang and Srihari (1986), Rich and Venkatasubramanian (1987), Cross (1984), and Adams (1986)).

One interesting possible approach to deep knowledge is to consider the depth of knowledge possessed by human experts. The human performance is especially interesting because human experts seem to be able to estimate the functioning of a physical system even in cases that have not been encountered earlier and because studies of cognitive processes of human experts seem to suggest that they, i.e. human experts, use some kind of domain models in their reasoning (e.g. Sweller's (1988) results support this belief).

[1] Note that according to Clancey (1992) *all* expert systems are model based.

In this paper we will consider the nature of deep knowledge from the perspective of scientific knowledge. Scientific knowledge was chosen because there are human experts succesfully applying scientific knowledge to tasks which require deep knowledge and robustness. Possible examples vary from engineers designing new equipment to doctors diagnosing patients. Hence scientific knowledge provides a promising starting-point for our considerations.

At this point it must be stressed that the considerations presented in this paper are applicable only to domain modeling. All other aspects like knowledge used by natural robust creatures are left out. In addition to that restriction, the discussion is limited to physical domains only. The domain we are interested in is physical reality, i.e. the concrete world, in which case pure theory building and abstract entities are omitted. The main reason to consider physical domains only is that there obviously is a need for a method which could be used to incorporate causal and structural knowledge into numeric simulation models because qualitative models do not work wery well with all physical domains (Ahonen, 1994).[2]

In the next section we will define some of the key terms and concepts.

3.2 Models

In order to make the following discussion clearer we will define several concepts, some of which have been already used. In this paper the term *physical reality* means the concrete world. Similarly, a *real-world phenomenon* or a *real-world object* is a phenomenon or an object which exists or can potentially exist in the physical world. A *model* is a representation of physical reality. Such representations are simplifications of physical reality, and unfortunately they do not produce reliable information on every aspect of the phenomenon being modeled (Lewis and Smith, 1979, 2). Our definition of models differs from the normal definition in its clear nature as a representation of physical reality (see, e.g., (Futo and Gergely, 1990) for the usual approach). *Simulation* is the creation and execution of dynamic models employed for understanding system behavior (Fishwick, 1992). These definitions restrict the applicability of the following discussion.

Sometimes it may be the case that the model is used to simulate a situation that has not been considered during the development of the model. Such a *new situation* is a case in which the model is intended to be used without prior knowledge of the possibility that such a case may be encountered during the use of the model. A model is *robust* if it behaves like its real counterpart would behave in a similar new situation. In other words, a robust model produces right values to the variables used to represent the modeled real-world phenomenon or object.

[2] One of the fundamental points to remember when dealing with physical domains is that scientific knowledge differs from everyday knowledge (or intuitive knowledge) (Tuchanska, 1992), and it would seem fairly strange to consider physical causality and intuitive mythical causality, which has been discussed by de Kleer and Brown (1984), to be the same. In addition to that, the philosophical discussion presented in (Ahonen, 1994) suggests that qualitative approaches are not very good for modeling some types of continuous physical systems.

Models should be designed in a way that they do not lose their modeling ability even in new situations. The obvious way to achieve this goal is to develop models by using knowledge that is required to determine the behavior of the model in as many new situations as possible and do it despite the fact that models are simplifications (note that the models of human experts are also simplifications). We will call such knowledge deep knowledge because it can be said to go beyond the surface of the knowledge on which our models are based. *Deep knowledge* is knowledge of the features that define the structure and behavior of the phenomenon or the object considered. Deep knowledge provides the basis on which robustness may be built.

This definition of deep knowledge is usable, however it leaves two important questions open. The first question concerns the actual nature of deep knowledge, and the other question the structure of models that are based on deep knowledge.

Despite the superficial differences both AI systems and simulation systems include models, simulation systems by definition, and according to Clancey (1992) all AI systems include models. Hence the distinction between knowledge representation in the traditional AI sense and knowledge representation in the traditional simulation sense is not very clear, although some rough divisions may be made on the basis of the intended usage of the models (Miller et al, 1992). Because the connection between both types of models is very strong, we can proceed as if they were the same.[3] The sameness will be assumed because the approach we will outline later in this paper will be the same for both AI and simulation modeling. The approach outlined in this paper is based on the philosophical discussion and proposal presented in (Ahonen, 1994). In order to approach modeling tasks from a clearer point of view it is necessary to outline how modeling tasks proceed. One way to look at such tasks is to divide them into different steps or levels. For example Zeigler (1976) divides modeling tasks into the following levels:

- the “real system”, a source of potentially observable data;
- the “experimental frames”, a set of limited observation or manipulation cases for the real system;
- the “base model”, a comprehensive model of the system in every experimental frame;
- the “lumped model”, a simplified version of the base model, which is simplified in a way that still provides the reliability of the original base model in interesting cases;
- the “computer model”, an implementation of the lumped model in a computer programming language on a machine.

The above steps of traditional modeling include the assumption that there is no model which is *complete*, i.e. so thorough a replica of the modeled object or phenomenon that it could not be improved as a result of changes

[3] There is clearly a need for a representation which could really connect both approaches (Kuipers, 1993).

in knowledge about original or experienced defects in the developed model. This is clearly expressed by Zeigler (1976, 31) who concludes that:

> In any realistic modeling and simulation area, *the base model description can never be fully known.* (original emphasis)

The definition of robustness and the inherent incompleteness of models mean that no really robust models exist. But for pragmatic reasons it is a compelling and useful aim to make models more robust.

In the next section we will briefly look at the connection between shallowness, robustness and the explanatory powers of the knowledge used in the creation of models.

3.3 Shallow models, explanation, and robustness

In the field of the philosophy of science, one of the most active topics of interest is the explanative nature of scientific knowledge. For our purposes we will consider the explanatory aspect of scientific knowledge, and suppose that scientific knowledge consists mainly of explanations. (This assumption seems to be a safe one because the explanatory nature of scientific knowledge tends to be so widely accepted a proposition, see e.g. (Fetzer, 1981), (Niiniluoto, 1983), (Salmon, 1971), and (Salmon, 1984), that it is not seriously questioned at all.) In order to avoid a multitude of arguments we will assume that concepts are considered to be defined in such explanations.

To clarify the connection between robustness and the explanatory powers of models we will briefly consider so-called shallow models, which are often thought to be the opposite of deep models (see e.g. Chandrasekaran and Mittal (1983)). Although we will not define shallow models explicitly, we will briefly consider some of their features. In this paper the term *shallow-model* is interpreted in a way that makes clear that shallow models primarily include shallow knowledge acquired by, for example, empirical observations, which are then written into rules or equations that define the values of variables by relations between those values.[4] If we consider shallow models to include only empirical knowledge acquired by observing the surface behavior of an object or a phenomenon, then we can assume that the features of such models differ from the features of the respective deep-knowledge models. In order to emphasize the distinction we will briefly consider some simple examples of models that provide desired robustness or do not provide it.

Consider, as an example, a situation where a human being cannot get enough vitamin A. It is well known that too little vitamin A causes illness, and it is similarly well known that by giving more vitamin A to the person in question, his/her health can be improved. From a limited amount of knowledge it can be concluded that by giving more and more vitamin A the person's health will improve infinitely. It is, however, known that this

[4] In other words, a shallow model is like a "black-box" which responds to external stimuli and produces values to interesting variables according to predefined mappings from the stimuli to the values. The inside of the box cannot be seen in the case of shallow models.

is not the case. Knowledge of the biological low-level effects of vitamin A would have enabled the avoidance of the wrong conclusion. This example is, however, fairly weak because a fully covering statistical example could be created reasonably easily. Although a covering model could be created, that statistical model would suffer from the same shortcomings as the next example.

As an example of shallow models which completely lack explanatory power we can consider a model of the growth of pines. Having gathered a great number of observations, it is possible to generate a function, and say that

> 'According to our empirical observations we can give the statistical relationship between the height growth and the age of pines as
>
> $$h(t) = \frac{H}{1 + h_1 e^{-h_2 t}} \tag{3.1}$$
>
> where H (max. height), h_1 and h_2 are 21 m, 20.4 m and 0.064, respectively'

which is true. But by saying (as Oker-Blom et al (1988)) that

> 'The height growth was assumed to be independent of stand density and was modeled as a logistic curve (3.1). The values of H (maximum height), h_1 and h_2 were chosen as 21 m, 20.4 m and 0.064, giving a height development in accordance with existing growth and yield tables.'

and implicitly supposing that the equation explains something, we say something that is not true. If we say that the equation states the statistical relationship between h and t, we do not say anything that is untruthful but we do not provide any explanation, either. In addition to not being able to provide any explanations, the model is not robust. The lack of robustness can be easily pointed out by asking "What will happen to the growth if the climate really changes as they say?". The presented statistical model cannot answer that question.

Obviously Oker-Blom's model does not have great explanatory power in the sense required by the question:

'Why does the equation (3.1) hold for
the height growth and the age?' (3.2)

The question (3.2) to which a human expert could easily provide an answer, cannot be answered by the model — although that question is one of the most obvious to be asked.

Although knowledge present in Oker-Blom's model can explicitly determine a model, it does not even implicitly answer the question (3.2). The fact that models of this type do not include the knowledge required for adequate explanations does not mean that explanations do not exist at all. The problem of non-existence of explanations in statistical relations has been noted earlier (Fetzer, 1981, 87; Salmon, 1984).

It is actually easy to find examples of cases in which obvious statistical relations have nothing to do with explanations or actual mechanisms of the case. We can, for instance, take Salmon's (1984, 268) example:

> '...there is a strict positive correlation between the amount of time required for clothes hung out on a line to dry and the distance required to get an airplane off the ground at the nearby airport. I take it as given that the higher the relative humidity, the longer the clothes will take to dry. Thus the phenomenon that requires explanation is the fact that increased relative humidity tends to make for greater takeoff distance, other things — such as the temperature, the altitude of the airport, the type of the plane and the load it is carrying — being equal...'

In the example chosen by Salmon there is no direct explanatory connection between the observed phenomena. The actual explanation gives the common reason for the phenomena. The interesting thing is that in addition to providing the necessary knowledge, the explanation hints at the possibility of creating a model of the concepts used in the explanation. Such a model could be used in a variety of cases which involve physical phenomena and objects modeled — in other words, such a model could be fairly robust. Note that in Salmon's example there is no way of creating a covering statistical model which could offer any insight into the correlation.

Although statistical analysis of shallow knowledge is often very useful, the unfortunate fact is that sometimes even strict correlations or conditional probabilities of 1 or 0 do not mean that there is an explanatory relation. It can, at first, seem strange to say that $\mathrm{P}(A|B) = 1$ or $\mathrm{P}(A|B) = 0$ does not mean that there is any connection between A and B, but it is fairly easy to find examples which show that the claim is true. A great deal of this is implied by the fact that statistical correlation is a symmetric relation while causality is not (Irtzik and Meyer, 1987). Obviously an alternative is requied in order to create robust domain models.

Considering the vitamin A example and Oker-Blom's equation and Salmon's airport-example and proper explanations for those cases[5], it is interesting to note that such explanations have to explain the observed phenomena by using concepts and entities which are not parts of the original description of the case. The same can be said of any physical phenomena which is explained using Newtonian mechanics because Newtonian mechanics is not present in the description of e.g. the famous apple often said to be connected with the late Newton. It seems to be the case that those explanations use deep knowledge, i.e. knowledge that goes beyond the observed surface features (i.e. the values of the actually interesting variables) of the case and the original description of the situation.

Problems with shallow models lead us to look for other approaches. As was said, one of the most important aspects of knowledge is to answer why-questions, questions which often imply some kind of causality. Unfortunately the concept of causality are very difficult to define, but we will, how-

[5] The explanations are left out because such case-specific aspects are outside the scope of this paper.

ever, briefly outline causality and explanation. We will continue on to the assumption that causal relations exist.

Thus far we have briefly considered the connection between explanatory power and robustness. In the following section we will outine one possible approach to the conceptualization of physical reality. The approach differs somewhat from the usual object-phenomena -models.

3.4 Causality and explanation

The use of causality in different forms is actually very tempting. The concept of causal relations is acceptable at face value an and cognitively plausible. Actually, different levels of explanatory powers and probability in causal explanations seem to exist (see e.g. Sober (1984)), which implies the versatile nature of causal explanation.

Probably the most useful approach to define the features of the physical reality spoken about is proposed by Salmon (1984), who proposes the concept of causal processes instead of causal events. Of causal processes he says:

> 'Causal processes propagate the structure of the physical world and provide the connections among the happenings in the various parts of space time. Causal interactions produce the structure and modifications of structure that we find in the patterns exhibited by the physical world. Causal laws govern the causal processes and causal interactions, providing the regularities that characterize the evolution of causal processes and the modifications that result from causal interactions' (Salmon, 1984, 132)

The concept of causal processes is not, however, very clear; many of their features have to be defined with more precision. Those features include mark transmission, structure transmission, the principle of causal influence, and causal interaction.

The most important criterion for a causal process is its ability to transmit a mark. By mark transmission we mean that a causal process can transmit a mark from point A to point B (and every point between them) without further interactions. The *mark transmission* (MT) is defined, in a more explicit way (Salmon, 1984, 148) , as:

> Let P be a process that, in the absence of interactions with other processes, would remain uniform with respect to a characteristic Q, which it would manifest consistently over an interval that includes both the space-time points A and B $(A \neq B)$. Then, a mark (consisting of a modification of Q into Q'), which has been introduced into process P by means of a single interaction at point A, is transmitted to point B if P manifests the modification Q' at B and all stages of the process between A and B without additional interventions.

Note that marks in a process are, actually, changes in the process itself. Therefore we can say that a transmission of a mark is a transmission of the changed structure of the process transmitting the mark, and a process can always be said to transmit its own structure, changed by a mark or not. The principle of *structure transmission* (ST) can be formulated as follows (Salmon, 1984, 154):

> If a process is capable of transmitting changes in structure due to marking interactions, then that process can be said to transmit its own structure.

The fact that a process does not transmit a particular type of mark does not mean that it is not a causal process. Consider, as an example, the processes of a hard rubber ball and a particular beam of light (caused by a lamp and colored white). It is possible to paint a green mark on the surface of the ball but it is not possible to do the same to the beam of light, although it is possible to change the color of the beam to green by using a green filter. Marks must be consistent with the structure and properties of causal processes — a causal process cannot be marked by every method.

In accordance with the principle of structure transmission there must be a way to define how a causal process propagates causality from one space-time locale to another. The *principle of causal influence* (PCI) can be defined as (Salmon, 1984, 155):

> A process that transmits its own structure is capable of propagating a causal influence from one space-time locale to another.

Combined together the concepts MT, ST and PCI define what a causal process is. Although a causal process can be very effectively defined by them, no interactions between processes have been defined. Obviously processes interact and interactions constitute the actual structure of causal relations.

As can be deduced from the definitions of MT, ST and PCI, there exist, in addition to causal processes, a great number of non-causal processes. The existence of non-causal processes makes the definition of causal interaction (CI) between processes a quite difficult task, and one proposition is made by Salmon (1984, 171), and is as follows:

> Let P_1 and P_2 be two processes that intersect with one another at the space-time point S, which belongs to the histories of both. Let Q be a characteristic that process P_1 would exhibit throughout an interval (which includes subintervals on both sides of S in the history of P_1) if the intersection with P_2 did not occur; let R be a characteristic that process P_2 would exhibit throughout an interval (which includes subintervals on both sides of S in the history of P_2) if the intersection with P_1 did not occur. Then the intersection of P_1 and P_2 at S constitutes a causal interaction if:
>
> 1. P_1 exhibits the characteristic Q before S, but it exhibits a modified characteristic Q' throughout an interval immediately following S; and
> 2. P_2 exhibits the characteristic R before S, but it exhibits a modified characteristic R' throughout an interval immediately following S.

Salmon proposed that scientific explanations should use the concept of causal processes as the basis, and that actual explanations are linguistic descriptions of chains of causal interactions and mark transmissions. The discussion of how explanations are actually created by using the causal processes -concept is not presented in this paper — for that discussion we refer to (Salmon, 1984). In the following section we will consider a modeling approach that is based on the concept of causal processes and briefly discuss the intuitively tempting claim that models able to provide explanatory knowledge are robust.

3.5 Causal process models

In this section we will describe a domain modeling approach based on the Salmonian concepts of causality. Generally we will consider models that may be called explanatory models. An *explanatory model* is constructed according to the concept of causal processes and the model produces quantitative results, i.e. values to variables, based on processing the descriptions of the modeled causal processes. Note that we are considering models which are not qualitative in the meaning discussed in (Weld and de Kleer, 1990). Because qualitative models of the physical reality suffer from philosophical problems (Ahonen, 1994), we have to develop an alternative to normal qualitative and quantitative approaches. Our approach incorporates causal[6] and structural knowledge directly into numeric simulation models.

In order to prevent any confusion between explanatory inferences (i.e. the mechanism by which explanations are generated) and explanatory models (i.e. models of the causal structure of the reality) we can consider the sentence ‘I see that there is snow outside and therefore I know that the temperature outside is below zero degrees celsius’. The sentence is not an explanatory model, and even the sentence ‘I see that there is snow outside and therefore I know that because water can be snow only if the temperature is below zero, I am able to deduce that the temperature outside is below zero’ is not an explanatory model. Writing that ‘When the temperature is below zero° C, water molecules arrange into a combination in which they do not constitute a liquid’ we are much nearer to having an explanatory model. Note that a transmission of energy causes water molecules to rearrange – an obvious case of causal interaction and mark transmission. In the next subsection we will outline a simple formalization of explanatory models.

[6] At this point it must be stressed that in this paper we are discussing physical domains and physical causality only. This is partly because the approach presented is developed to overcome the problems encountered by qualitative modeling of physical reality, and the knowledge considered is our best knowledge about the physical world, i.e. scientific knowledge, not naive knowledge used in qualitative modeling approaches like the mythical causality discussed by de Kleer and Brown (1984).

3.5.1 The structure of explanatory models

If we consider any practically useful model, it is very probable that such a model would include several processes which may interact with each others. In addition to interacting with other processes from time to time, it is interesting to note that causal processes P_1 and P_2 can be in continuous interaction. Consider, as an example of continuous causal interactions, a car. The motor of the car always has several different causal interactions with other parts of the car, and there are, in the motor, parts which can be thought to be causal processes continuously interacting with each others. An explanatory model M (without the interactions) could now be said to be a set of causal processes, i.e.

$$M = \{P_1, P_2, \dots, P_n\}$$

in which P_i denotes a causal process included into the model. In order to have interactions, as defined in the previous section, between various processes an explanatory model must be redefined as

$$M = (P, I)$$

in which P is the set of the modeled processes and I is the set of the possible interactions between them. Every member of I is a CI.

Another interesting feature of causal processes is that they are divisible. Consider, again, a car. For a pedestrian injured by the car, the car constitutes one process, and for a mechanic the car consists of a multitude of different causal processes. The example of the car shows that we can divide a process into several processes depending on what we know of it. A causal process can sometimes be divided into several subprocesses which have causal interactions and causal structures of their own.

Because some processes can be divided into subprocesses, a process should be clearly distinguishable from its subprocesses by means other than just having different characteristics. One possible method for this separation is the introduction of the concept of levels; on lower levels there would be the subprocesses of process P_j and P_j itself would be on the upper level. Now the process P_j can be formalized as

$$P_j = \{P_{j,1}, P_{j,2}, \dots, P_{j,n}\}$$

in which $P_{j,i}$ is a *subprocess* of P_j.

In principle the number of levels of an explanatory model is not restricted. Different models require different numbers of levels, and the number of levels of a model is restricted by the limitations dictated by the computer on which the model is used or by the fact that sometimes the required accuracy can be achieved by using fewer levels, or by the fact that human knowledge of the problem may not be complete enough.

It is interesting to note that the concept of levels seems to a be natural solution for the modularity problem. Different processes existing on different levels fulfill the modularity requirements presented by Cota and Sargent (1992).

In order to define the characteristics of a processes there must be a way to identify that particular process. Therefore it is reasonable to consider the characteristics of a process to be identified by an intensional name (the name represents the intension of the intended real counterpart existing or potentially existing in the physical reality). The name can be considered to identify the variables and the values of the variables and the characteristic space-time functions which constitute the features according to which the process is defined and characterized. Now a process is formalized as

$$P_j = < N, S >$$

in which S is the set of subprocesses and N is the name of P_j.

A process is characterized by the values of the variables and the variables themselves and the space-time functions which are used to represent the process in question. The name of the process is used to clarify the recognizable features of the process in order to make the characterization more comprehensible. In order to include the actual process characteristics in the model, we can define a process as a tuple

$$P_j = < N, V, F, S >$$

in which N and S are as above, V is the set of the variables used to represent the characteristics of P_j, and F is the set of space-time functions which define the characteristics of P_j together with V. Note that a model of a causal process does not require any names in principle, because an actual process is characterized by the variables and the values of the variables and the space-time functions which may change the values of the variables (the space-time functions are required in order to enable specific time-related characteristics like Polonium^{218}'s tendency to lose mass over time). Processes are named in order to make the model more understandable and easier to use and construct. Now an explanatory model can be defined as a tuple

$$M = < P, I >$$

as above.

In this solution the variables in V and the functions in F are the method by which the characteristics of a given process are represented. These characteristics include the structure of the process and possible marks, i.e. changes to the structure of the process, and transmit the structure and the marks over space-time. If the process definition includes a variable for the color of a rubber ball, that ball will be *marked* by painting it and changing the value of that variable. That change would then be transmitted over the simulated space-time and the representation would fulfil the definition of MT. Similarly the "variables with functions" representation obeys the principle of structure transmission, and by being able to transmit their own structure the representations are able to propagate causal influence from one space-time locale to another and hence fulfil the definition of PCI. Naturally every $I_k \in I$ must fulfil the definition of CI.

Obviously the movement of a car, which is a causal process, has an effect on the doors of the car — at least the effect of movement. Or consider a stone hitting one of the doors. Obviously causal processes on different levels can have causal interactions with each other. It seems actually to be the case that there is no limit for the level number difference between processes which have causal interactions with each others.

In some cases the causal interaction is constant, but sometimes it is only a possibility. The possibility of the causal interaction exists between any cup and any table, but it actually exists only if the causal processes in question fulfill certain characteristics (one example of that kind of characteristics is the space-time locations of the processes). This makes I_k more a definition of possible causal interactions than the actual interaction.

It can be said that the definitions of interactions are intentions of interactions and actual interactions are their extensions. The general definition of an interaction could be thought to be a demon which causes a real interaction to happen when certain requirements are satisfied.

3.5.2 Remarks on explanatory models

In the subsection above we defined the structure of explanatory models and some other concepts. In this subsection we will consider some special semantic and syntactic features of our explanatory model structure, and try to clarify its intended interpretation.

In the explanatory model structure a causal process is a set of variables and functions. The set is named by a term, and the naming relation defines the characteristics of the process connected to the term. The mapping from the name to the set defines the characteristics of the process. The mapping represents the intension of the name, and the set of the variables and functions to which the mapping maps the name represents the extension of the name. This means that a certain process, i.e. the set of variables and space-time functions, can be named by several terms at a time and that the set of terms can change over time. As the consequence of the possibility of multiple names, N must be defined as a *set of names*.

One result of this characterization is that causal interactions and characterizations themselves can make the process fall under different characterizations at different times. Consider, as an example, a steel plate. If the plate is crushed into another form, it is not a steel plate any more although it is still steel and still the same process. Before crushing, the steel plate was named by “steel” and “plate”, but after crushing that process is no longer named by “plate” although it is still the same process. Actual processes do not disappear although they can change into forms which may be very different when compared to the original forms. Physical relations are not static. A part of a machine can be removed, or a physical entity can be broken into several pieces. Causal interactions can change the physical relationships between processes. This requires great flexibility from the simulation control program which should allow such changes and maintain the reliability of the system despite those changes.

Two models of two different chairs have very much in common despite the fact that the chairs can be very different in some respects. A model of every physical entity can include knowledge of atoms and particles, and knowledge of atoms and particles is very similar in any case. The relation between knowledge used in two models could be said to mean that causal processes on deeper levels are more and more alike conceptually, although physically different. Two processes may have very similar structure and definition. If models T_1 and T_2 are both models of pines, then they have very similar knowledge on their lower levels. In other words, knowledge used in a model is more common to other models on lower levels than it is on upper levels.

In order not to give the impression that the only useful type of explanatory models includes knowledge of even subatomic relations, we have to point out a quite obvious way of determining where to stop modeling. In every case we start modeling by using a scientific theory according to which we model. During the deepening of the model we will always end with a level which cannot be defined without using a scientific theory that is different from the theory according to which we started modeling. Such a change of theories provides a suitable point in which we may stop modeling and still be able to provide required robustness. If we are modeling a mechanical device, e.g. a clock, we may be able to stop modeling when we have reached a level on which we have exhausted the means of Newtonian mechanics. Explaining that level would require entirely another kind of theory, and in many cases such theories are not necessary in order to achieve the required robustness or explanatory power. Note, however, that the number of scientific theories used in a model depends on the pragmatic aspects of the modeling project, and those aspects depend on the case considered.

There is often no need to compute the actual causal interaction on every level of the model. The actual number of causal interactions required to be computed depends on the model and the use of the model. Consider, for example, the familiar fact that a metal table will support a coffee cup. It is clear that the explanation of this fact involves the details of atomic structure and the appeal to concepts as unfamiliar to everyday experience as the Pauli exclusion principle (the example is taken from (Salmon, 1984)). An explanatory model of the situation can be created, but a precise model would include the atomic structure, and the workings of the atomic structure would appear to be extremely difficult to be computed.

The computational problem can be avoided by hiding the atomic structure of the model from normal computations. Such hiding could be done by using the method described in (Ahonen and Saarenmaa, 1991). If computations are necessary, the causal processes and interactions needed could be created when required and disposed after the need has been met. The hiding of the more precise structure of a model in order to avoid computational problems seems to promise a solution to the question of the computational effectiveness of causal models, see e.g. (D'Ambrosio et al, 1985).

3.5.3 The robustness of explanatory models

Although we would like to show that the structure of explanatory models does provide robustness, we have to admit that it is not easy. The best we are able to achieve is to discuss some features of explanatory models and the intuitively robust nature of those features.

A model should be able to function properly in new situations. From the nature of models it is clear that robustness is actually provided by the knowledge already present in the model, and the same seems to be the case with the robustness present in estimations made by human experts. This makes us to adopt a view very similar to Hacking's (Hacking, 1967, 319). According to Hacking you do not actually know something if you have not performed the necessary inferences required to deduce it.

If we assume that the robustness of models is due to knowledge which is used in a new way, we can understand the role of deep knowledge better. If our assumption of the growing similarity of deep knowledge is true, then it is tempting to conclude that new combinations of such basic knowledge, i.e. deep knowledge, provide the robustness. Since very deep knowledge of one model is very alike to similarly deep knowledge of another model, it may be the case that such knowledge can be easily arranged in combinations that provide the required robustness at the upper levels of the model.

We have considered the reasons for the robustness of models that use deep knowledge, but we have not paid any attention to the robustness of our explanatory models. We claim that explanatory models may provide a uniform approach to domain knowledge, explanatory power and robustness, because explanatory models employ only one concept — the concept of causal processes. The concept of causal processes enables the modeler to use uniform representations and a uniform way of thought throughout the development of the model and maintain the explanatory power of scientific knowledge. Causal processes provide a method by which more or less deep knowledge may have similar conceptualization and representation notwithstanding its depth.

The concept of causal processes has originally been developed to provide a uniform approach to the study of scientific knowledge and scientific explanation, and there seems to be no reason to assume that the same approach could not be used in domain modeling. The causal process approach with a suitable formalization may provide a very promising alternative by defining what kind of knowledge should be used and what kind of interactions models include. Considering the nature of causal processes it is reasonable to believe that representations of processes enable very different and unanticipated causal interactions and mark transmissions to occur.

In the next section we will briefly discuss the creation of explanatory models.

3.6 Building an explanatory model

In this section we will briefly outline how an explanatory model could be built. Although the actual process of building such models is not covered here, we will outline some of the major issues and pitfalls.

3.6.1 Real systems and experimental frames

The choice of the domain to be modeled may seem to be an easy task. Unfortunately it is not — especially if we consider the requirements for the domain.

Earlier we restricted our interest to the physical domains, i.e. the concrete world. If we consider physical domains only, we have to drop the modeling of purely theoretical things like mathematics, arts and other non-concrete artifacts. In other words, the domain must be one you can take a piece off (probably not literally, but in the same meaning).

In addition to being limited to the physical world as his/her domain, the modeler has additional limitations to his/her view of the structure of the world. As discussed above, the modeler will consider *processes* in space-time continuum — and those processes must be *causal processes* in the Salmonian sense.[7] The most significant feature of causal processes is their different nature from normal conceptualization of processes in modeling, especially in AI modeling (for a more usual approach, see e.g. (Drabble, 1993)) in which processes are considered to be series of events occuring in the world of objects. This is not the case when the world is conceptualized as causal processes.

The causal process conceptualization of the physical world has no objects and *no* events in the traditional sense.[8] Every physical object or phenomenon is thought to be a *causal process* or a *causal interaction* between such processes. Processes are *not* series of events, they are the basic *structure* of the world on which everything else is based. Processes have characteristics which identify them and which make them behave in their specific ways, but they are not objects.

The fundamental difference between objects and causal processes is in their connection to the physical reality. It is much easier to decide that objects may be almost anything, not only something which really exists in the physical world. One of the main reasons for using causal processes when considering the world is that the definition of causal processes strictly defines what there is in the physical world. Normal object-oriented approaches do not impose such restrictions on the suitable domain. In order to make the nature of the modeled reality and the usage of knowledge unambiguous such restrictions are, in our opinion, required. Such restrictions are needed especially if we want to maintain a coherent structure and uniformly interpretable representation in our models.

The strict definition of the domain to be modeled and thorough thinking

[7] Note that *all* qualitative approaches to continuous causal processes suffer from the problems of qualitative modeling discussed in (Ahonen, 1994). All qualitative approaches handle continuity in the same way (Bobrow, 1984), and hence approaches like the "Qualitative Process Theory" of Forbus (1984) suffer from the same problems. Therefore Qualitative Process Theory does not provide satisfactory means to achieve our goal, i.e. to be able to represent causal and structural knowledge in models which produce reliable numeric simulation results.

[8] This makes causal ordering and other considerations presented by Iwasaki and Simon (1994) unnecessary when dealing with physical systems. In this paper we do not, however, attempt to say anything about the usability of those considerations when modeling other domains.

prior to commencing the actual modeling project are even more necessary when modeling by using causal processes than other approaches. The exclusion of all features and concepts which do not exist in the strictly defined world of causal processes is a necessary step in order to make the creation of explanatory models successful. This makes it necessary to focus the possible experiments on the definition of the characteristics of causal processes existing or potentially existing in the modeled domain and the possible causal interactions between them. Such experiments may not be easily planned or carried out. We do, however, leave the discussion of the identification of causal processes existing in a specific domain out of this paper, for such considerations we refer to (Salmon, 1984).

In the next subsection we will briefly consider what specific features the usage of the causal process concept and explanatory models will require from the base model created from the chosen domain and the experiments done to it.

3.6.2 Base model

The definition of the domain and the possible experiments carried out in order to find out more about the domain are themselves the first step in the creation of the base model. It is very difficult and probably impossible to make a distinction between the choice of the domain, the planning of the experiments and the creation of the base model.

The definition of the domain to be modeled and the creation of the base model seems to be a circular process. The first step is to decide to model, for example, a tree and the second step is to start the creation of the base model of the tree. During the creation of the base model the modeler has to identify the causal processes and decide to model a specific subdomain of the original domain (a needle is a good example of such subdomains if the original domain is a pine).

The close relation between theory building and the choice of what experiments to conduct has been discussed in the literature of the philosophy of science, and it seems to be the case that it is very difficult to identify the order in which those phases appear. Hence we will leave that discussion out of this paper and only refer to the discussion in the field of the philosophy of science.

In the next subsection we will briefly discuss the creation of the lumped model from the previously defined base model. In addition to that, we briefly consider the implementation of explanatory models.

3.6.3 Lumped model and computer implementation

Unfortunately the base model of the domain cannot ever be fully known, and that makes it very difficult to produce robust models by simplifying theories which are already incomplete. Fortunately it seems to be the case that scientific knowledge is, in many cases, fairly robust.

Many scientific theories are fairly robust,[9] and in our opinion they are

[9] This is because they are themselves conceptually or empirically testable, either directly

robust because they attempt to offer a testable explanation to the questions regarding the structure and behavior of reality[10]. Such theories are, of course, members of one type of models. The apparent use of theories in explanation may help us to reduce the possibly very large set of domain theories into an implementable lumped model.

As discussed above, the most viable point to stop modeling is when the modeler should change the theory of the domain. If an explanatory model of a mechanical device has been defined according to Newtonian theory and in order to deepen the model the modeler would need to use quantum mechanics, then that level would be an obvious level to stop modeling. In that way several layers of scientific theories of the domain could be dropped from the lumped model, although those theories are important parts of the base model. Note that the base model could, in our opinion, be created by using multiple levels of specification and theoretical accuracy — of course all the theories should be able to be used to model the structure and behavior of the domain according to the concept of causal processes.

The actual computer implementation of an explanatory model would not be an obvious task to be completed using a conventional language like FORTRAN 77. The most interesting alternatives seem to be various object-oriented languages, but even with them there are some conceptual problems. The most difficult problem with object oriented languages is that they are mainly event-oriented and causal processes are, by definition, continuous and the simulation of causal processes should proceed over constantly flowing time, not as series of events.[11]

In order to make the task of implementing an explanatory model easier it might be a good idea to develop a specified programming environment and a specific language for the task. The language should directly support the concept of levels and other features of explanatory models. In addition to supporting the features of causal processes, the language should include methodologies to connect the implemented model structure and behavior to an explanation generation system of some type.

3.7 Discussion

It is plausible that the cognitive models used by humans are very robust because of the human ability to connect different items of knowledge in new ways. If we assume this, it is easy to conclude that robustness of domain models may be based on the same method. The obvious way to introduce robustness to domain models seems, in that case, to depend on the conceptualization and representation of domain knowledge. Such conceptual-

or indirectly (Bunge, 1973, 27-43).

[10] This discussion is outside the scope of this paper and its fundamental aspects and problems are left to people who have specialized in those problems. In this paper we attempt only to use one specific point of view of the scientific knowledge, namely that scientific knowledge consists of *explanations*.

[11] A direct application of the methods used in qualitative simulations of continuous processes, e.g. QSIM (Kuipers, 1985), is not very promising because those methods are more or less directly connected with the philosophical assumptions behind qualitative modeling.

ization should provide a versatile and uniform enough approach to domain knowledge and its representation. In addition to that, the conceptualization should be able to ease the discomfort that Iwasaki and Simon (1993) express about the subjective and arbitrary representation of causality.

The concept of causal processes provides a uniform view to domain knowledge, and it is not very difficult to develop a framework for the modeling of causal processes and their features. This should enable us to follow a clear path in our modeling activities, and it is reasonable to assume that models with enough levels and accurate descriptions of processes and the features of processes are robust. Unfortunately this is not obvious, although most of the writers that have discussed deep knowledge have adapted a similar point of view to modeling with deep knowledge.

The uncertainty concerning the robustness provided is due to the fact that we do not actually know how human experts generate new combinations of knowledge. If it is done by methods that work like normal logic or straightforward simulation, then there should not be any problems. But if the only method is induction, our attempts to develop models robust enough to be comparable to the cognitive models used by human experts may be void.

Another hazard with the use of the concept of causal processes may be that the concept of causal processes seems to be extremely difficult or even impossible to formalize fully with descriptive methods. It may be possible that the nonlogical modalities and counterfactuals present in the definition of causal processes prevent exact formalization (Fetzer, 1987). This problem does not, however, make the use of the concept of causal processes impossible, it may only complicate it by giving more responsibility to the modeler.

The modeler's difficulties should, however, be manageable because the structure of explanatory models is surprising simple. By using that simple structure and uniform conceptualization of physical reality, the creation of robust domain models should be possible and resulting models would fulfil the requirements defined by Hayes (1979). Thoroughness is an inherent feature of causal processes because they are the basic structure of the physical reality, and fidelity and density are directly supported by the definition of the explanatory model structure. In order to make such models work, the amount of detail has to be fairly high and a working model would include relatively many facts when comparing their number to the number of concepts. Uniformity is, also, an inherent feature of explanatory models because everything existing in the physical reality is modeled by using the causal processes conceptualization.

One additional benefit of the concept of causal processes is that if quantitative models were developed according the approach, it could be possible to provide a natural connection between numeric simulation models and the representation of physical causality and structural knowledge. If there were a method to naturally include causal and structural knowledge into numeric simulation models, it should be much easier to provide the connection requested by Clancey (1992) and Kuipers (1993). That connection could enable us to create numeric simulation models which would include the best features of both qualitative and quantitative approaches and which would not suffer from the shortcomings of qualitative approaches discussed in (Ahonen, 1994).

References

Adams, T. L. (1986). Model-Based Reasoning for Automated Fault Diagnosis and Recovery Planning in Space Power Systems. In *Proc. 21st Intersociety Energy Conversion Engineering Conference*. San Diego.

Ahonen, J. J. (1994). On Qualitative Modelling. *AI & Society*, 8. 17-28.

Ahonen, J. J. and Saarenmaa, H. (1991). Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents. In *Proc. 6th Symposium on Computer Science for Environmental Protection*. 193-200.

Bobrow, D. G. (1984). Qualitative Reasoning about Physical Systems: An Introduction. *Artificial Intelligence*, 24. 1-5.

Bunge, M. (1973). *Method, Model and Matter.* D Reidel Publishing Company, Dordrecht.

Chandrasekaran, B. and Mittal, S. (1983). Deep Versus Compiled Knowledge Approaches to Diagnostic Problem Solving, *International Journal of Man-Machine Studies*, 22. 425-436.

Clancey, W. J. (1992). Model Construction Operators. *Artificial Intelligence*, 53. 1-115.

Cota, B. A. and Sargent, R. G. (1992). A Modification of the Process World View. *ACM Transactions on Modeling and Computer Simulation*, 2. 109-129.

Cross, S. E. (1984). Model-based Reasoning in Expert Systems: An application to enroute air traffic control. In *Proc. AIAA/IEEE 6th Digital Avionics Systems Conference*. Baltimore.

D'Ambrosio, B., Fehling, M. and Adams, T. (1985). Levels of Abstraction in Model Based Reasoning. In *Proc. ROBEXS '85 (Robotics and Expert Systems)*. NASA. Johnson Space Center.

de Kleer, J. and Williams, B. C. (1987) Diagnosing Multiple Faults, *Artificial Intelligence*, 32. 97-130.

de Kleer, J. and Brown, J. S. (1984). A Qualitative Physics Based on Confluences. *Artificial Intelligence*, 24. 7-83.

Drabble, B. (1993). EXCALIBUR: a program for planning and reasoning with processes. *Artificial Intelligence*, 62. 1-40.

Fetzer, J. H. (1987). Critical Notice: Wesley Salmon's Scientific Explanation and the Causal Structure of the World, *Philosophy of Science*, 54. 597-610.

Fetzer, J. H. (1981). *Scientific Knowledge.* D Reidel Publishing Company. Boston.

Fishwick, P. A. (1992). An Integrated Approach to System Modeling Using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. *ACM Transactions on Modeling and Computer Simulation*, 2. 307-330.

Forbus, K. D. (1984). Qualitative Process Theory. *Artificial Intelligence*, 24. 85-168.

Futo, I. and Gergely, T. (1990). *Artificial Intelligence in Simulation.* Ellis Horwood. Chishester.

Hacking, I. (1967). Slightly More Realistic Personal Probability. *Philosophy of Science*, 34. 311-325.

Hayes, P. J. (1979). The Second Naive Physics Manifesto. In: Michie, D. (ed) *Expert Systems in the Micro-Electronic Age.* Edinburgh University Press. Edinburgh.

Irtzik, G. and Meyer, E. (1987). Causal modeling: New Directions for Statistical Explanation, *Philosophy of Science*, 54. 495-514.

Iwasaki, Y. and Simon, H. A. (1994). Causality and model abstraction. *Artificial Intelligence*, 67. 143-194.

Iwasaki, Y. and Simon, H. A. (1993). Retrospective on "Causality in device Behavior". *Artificial Intelligence*, 59. 141-146.

Koton, P. A. (1985). Empirical and Model-Based Reasoning in Expert Systems. In *IJCAI-85 Proceedings.* Los Angeles.

Kuipers, B. J. (1993). Qualitative simulation then and now. *Artificial Intelligence*, 59. 133-140.

Kuipers, B. J. (1985). The Limits of Qualitative Simulation. In: *Proceedings of International Joint Conference on Artificial Intelligence.* 128-136.

Lewis, T. G. and Smith, B. J. (1979). *Computer Principles of modeling and Simulation.* Houghton Mifflin. Boston.

Miller, D. P., Firby, R. J., Fishwick, P. A., Franke, D. W. and Rothenberg, J. (1992). AI: What Simulationists Really Need To Know. *ACM Transactions on Modeling and Computer Simulation*, 2. 269-284.

Nardi, B. A. and Simons, R. K. (1986). *Model-Based Reasoning and AI Problem Solving.* IntelliCorp Technical Article Reprint. IntelliCorp.

Niiniluoto, I. (1983). *Scientific Reasoning and Explanation* (in Finnish). Otava. Keuruu.

Oker-Blom, P., Kellomäki, S., Valtonen, E. and Väisänen, H. (1988). Structural development of Pinus sylvestris Stands with Varying Initial Density: a Simulation Model, *Scandinavian Journal of Forest Research*, 3. 185-200.

Rich, S. H. and Venkatasubramanian, V. (1987). Model-Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants, *Computers and Chemical Engineering*, 11. 111-122.

Saarenmaa, H. (1988). Model-Based Reasoning in Ecology and Natural Resource Management. In *Proc. Resource Technology 88: International Symposium on Advanced Technology in Natural Resource Management.* Fort Collins (Colorado).

Salmon, W. C. (ed) (1971). *Statistical Explanation and Statistical Relevance.* University of Pittsburg Press. Pittsburg.

Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World.* Princeton University Press. Princeton.

Sober, E. (1984). Common Cause Explanation, *Philosophy of Science*, 51. 212-241.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning, *Cognitive Science* , 12. 257-285.

Tuchanska, B. (1992). What is explained in Science. *Philosophy of Science*, 59. 102-119.

Weld, D. S. and de Kleer, J. (eds) (1990). *Qualitative Reasoning about Physical Systems.* Morgan Kaufmann Publishers, San Mateo, California.

Xiang, Z. and Srihari, S. N. (1986). Diagnosis Using Multi-level Reasoning. In *Proc. Expert Systems in Government Symposium*, McLean (VA USA).

Zeigler, B. P. (1976). *Theory of Modeling and Simulation.* Wiley. New York.

Chapter 4

Causal Process Modeling

Jarmo J. Ahonen
Lappeenranta University of Technology, P.O.Box 20, 53851 Lappeenranta, Finland

Abstract

One of the most important benefits of the newest modeling paradigms used mainly in AI research is their ability to enable the use of structural and causal knowledge in models. Structural and causal knowledge has, however, been mainly used in qualitative modeling, and the use of those types of knowledge has not been common in quantitative, more traditional modeling and simulation. In this paper a structural and causal approach to the development of quantitative models is developed from Salmon's approach [39] [40] to causal knowledge. The approach is tested by an experimental model of Scotch pine which is implemented by using the developed implementation language and modeling environment. The developed approach is suitable for both discrete and continuous modeling and it enables modular modeling.

4.1 Introduction

In simulation and fault diagnosis in which models of the physical reality are used there is one obvious requirement for the domain model, namely that the model should be able to function as the modeled phenomenon in every case. Unfortunately the method by which this can be achieved is not clear, although some results and authors from the AI field speak in favor of so-called *deep knowledge* (one of the first articles in which that term has been used was [6]). Unfortunately deep knowledge has not been defined very well, and from that vagueness it naturally follows that one of the problems with deep knowledge is that we do not know what it is although we know what it should do, i.e. make models behave more like the modeled phenomenon or object. The most plausible approaches to robustness and deep knowledge use are *causal* and *structural* knowledge of the domain. Causal and structural knowledge has been successfully used to strengthen the robustness of models, see e.g. the articles in [44].

The use of the term deep knowledge seems to be closely related to the idea that we cannot create practical AI systems without turning our attention to the physical reality around us. Hayes proposed the use of models of the physical reality as a method of avoiding the problem of too simple and simplified problem domains [18]. In addition to the interest in physical domains Hayes said that reasoning is not enough, from which it may be concluded that an intelligent system would include both reasoning and knowledge of the physical reality. The distinction between reasoning knowledge and domain knowledge is sometimes very vague because a great deal of the reasoning knowledge can be, according to Clancey [7], considered to be qualitative domain knowledge. If we, however, make the division, it leads us to adopt the idea of *model based reasoning*, according to which the descriptions of domain and reasoning are kept separate. The model based reasoning approach does not determine the implementation [36], which seems to be especially useful in fault diagnosis and other fields that require the use of real-world models (see e.g. [11], [24], [32], [47], [35], [9], and [1]). Our interest is in the modeling part of such systems. It is, however, worth stressing that our main interest will be in *quantitative* domain models, i.e. models which produce numeric results, and not so much in *qualitative* models. See e.g. [44] and [27] for definitions of qualitative models. The definition of qualitative models and modeling has been left out of this paper because qualitative modeling can be thought to be a clearly different field than our main interest, which is to make traditional quantitative modeling more expressive. This distinction is very important because now we can leave qualitative considerations of our modeling approach, e.g. Clancey's [7] operators of qualitative modeling can be left out because Clancey left quantitative models out of his considerations. One thing to notice with quantitative and qualitative models is that qualitative models, which are weak in producing exact quantitative answers [15], can be thought to represent the reasoning oriented side of modeling and simulation, and quantitative models are the traditional simulation models which are good with exact answers. If qualitative models may be considered to be on the reasoning side of the model-based reasoning approach, quantitative models may be thought to represent the modeled phenomenon or object *only*, not the reasoning associated with the use of the model.

In order to enable quantitative models to be robust, we have to approach domain knowledge and its usage from a point of view which allows us to use the quantitative knowledge in a new way. This principally means that we have to approach the task from a knowledge-oriented point of view, and forget the normal technically oriented approaches which seem to cause the simulation approaches to pay less attention to the structure of domain knowledge.

In the following sections we will develop a modeling approach to *quantitative* modeling. The approach is developed to be able to incorporate both structural and causal knowledge into quantitative models. We believe that with a suitable approach to the quantitative modeling and simulation it is possible to avoid most of the problems associated with traditional simulation codes and to achieve a methodology which fulfills the modularity requirements presented in [8].

4.2 An approach to modeling knowledge

We think that it is necessary to develop also *numerical* approaches to the knowledge representation, because purely symbolic methods are not sufficient to represent the workings of the physical world.[1] Despite all superficial differences both AI systems and simulation systems include models, simulation systems by definition and all AI systems according to [7]. Hence the distinction between knowledge representation in the traditional AI sense and knowledge representation in the traditional simulation sense is not very clear, although some rough division can be made on the basis of the intended usage of the models [30]. There is clearly a need for a representation which could connect both approaches [26]. In this section we will, however, consider one approach to the scientific knowledge. The approach considered seems to connect the structural and causal aspects of qualitative modeling and allow quantitative aspects to be derived from the function of such structural aspects.

In our consideration the term *physical reality* means the concrete world. Similarly a *real-world phenomenon* or a *real-world object* is a phenomenon or an object which exists or can potentially exist in the physical world. A *model* is a representation of the physical reality. Such representations are simplifications of the physical reality, and as simplifications they will not produce reliable information on every aspect of the phenomenon being modeled [29, p. 2]. Our definition of models differs from the normal definition in its clear nature as a representation of the physical reality (see e.g. [16] for the normal approach).

Models should be designed in a way that they do not lose their modeling ability even in new situations. The obvious way to achieve this goal is to develop models by using the knowledge that is required to determine the behavior of the model in as many new situations as possible and do it despite the fact that models are simplifications (note that the models of human experts are that also). We will call such knowledge deep knowledge because it can be said to go beyond the surface of the knowledge on which our models are based. *Deep knowledge* is knowledge that defines the structure and behavior of the phenomenon or the object considered and provides the basis on which robustness may be founded.[2] In this paper we have incorporated deep knowledge into quantitative models, as will be seen later.

[1] See [5] for a discussion about possible problems present in the qualitative modeling approach.

[2] For a discussion on the nature of deep knowledge, robustness and modeling knowledge and their definition we refer to [4].

4.2.1 What modeling knowledge should express?

One of the most important features of modeling in expert systems and other systems is the ability to include some type of capability to express causality. Note that the ability to represent causality has not often been considered in the connection of quantitative i.e. numeric models. It is, however, very probable that in order to use and represent causality throughout the system (e.g. an expert system), the quantitative part of the system should also be able to express causality. We should realize that causality is not an easily definable feature of the reality.[3] Most of the knowledge representation practitioners should agree with at least some of the following assertions (these assertions have been listed by Dowe [12]):

1. causality is an *objective* feature of the world;
2. causality is a *contingent* feature of the world;
3. a theory of causality must be consistent with the possibility of *indeterminism*;
4. the theory should be (in principle) *time-independent* so that it is consistent with a causal theory of time;
5. the theory should not violate Hume's strictures concerning "hidden powers".

Assertion 1 reflects the discomfort that Iwasaki and Simon [22] express about the *subjectivity* of the representation techniques used to express causality in physical systems. The subjectivity of the causal interpretation is, however, a two edged sword.

Although the existence of the objective causality in the *physical world* could be considered to be a fact (at least in the light of our philosophical standpoint), some fundamental aspects of the nature of *any* model may impose problems for the representation of objective causality. First we have to remember that models (especially the computer models which we are interested in) are *representations* of the physical world, not pieces of the world. As representations those models are limited by the representation technique and the philosophy behind that technique.

The normal techniques used to represent causality often seem to be intended to capture the layman's concept of causality.[4] This approach to causality severely suffers, however, from its concentration on the layman-aspects of causality because there is not even one real guarantee about the usefulness of such an approach in the light of the five assertions about causality. It is actually possible that the layman's causality does not agree with even one of those assertions.[5]

[3] For this it is sufficient to point out the long-running philosophical discussion about ontology and causality.

[4] To us the discussion about the origins of the qualitative modeling in[5] shows that beyond any doubt.

[5] Hume's famous criticism of causality seems to be directed against the layman's causality, i.e. the type of causality used in qualitative modeling. We will not, however, discuss Hume's

Layman's causality is not objective, and some other approaches are required in order to produce a large set of useful models that are properly designed according to the *scientific knowledge* of the modeled object or phenomenon. This is because layman's causality is normally event oriented, and the causal ordering of such events may not be the same in common sense assumptions of the reality and the scientific conceptualization of the same reality. We believe that the common sense modeling of the world is not sufficient for many purposes, and that in order to achieve as much robustness as possible we should use a properly defined conceptualization of scientific knowledge.

The contingent nature of causality does not require clarification. The requirement of *indeterminism* is, however, more problematic. This requirement is very important, although it has not been considered with the seriousness it requires. It is interesting to note that in normal AI modeling approaches indeterminism has not received very much consideration, it has often thought to be covered by probabilistic considerations. This may be because most of the domains used in AI system development have not often included any indeterministic features. The possibility of strong indeterminism is, however, a very important requirement because there are fields in which most of the features are indeterministic.

The time-independence of causality is also an important aspect. As said in the assertion, the time independence allows the use of different approaches to time, although such time related theories may not be relevant for expert system building.

Although Hume's criticism against causality is important beyond doubt, it and its implications will not be considered here. Discussions about the Humean arguments are left to those who are better educated in these matters.

In the next subsection we will briefly consider an approach to domain knowledge, in this case scientific knowledge. We believe that the approach presented in the next subsection fulfills the five assertions, although some doubts of that have been expressed in the philosophical literature, see e.g. [12] and [13]. From our philosophical point of view the following approach is, however, a valid one.

4.2.2 The Salmonian Approach to Modeling Knowledge

The use of causality in different forms is actually very tempting. The concept of causal relations may be very explanation and cognitively plausible. Actually there seem to exist different levels of explanatory and probability in causal explanations (see e.g. [41]), which implies the versatile nature of causal explanation.

Probably the most useful approach to define the features of the physical reality spoken about is proposed by Salmon [39], who proposes the concept of causal processes instead of causal events. Of causal processes he says that

considerations longer, we are satisfied by referring to the literature of the philosophy of science.

> ‘Causal processes propagate the structure of the physical world and provide the connections among the happenings in the various parts of space time. Causal interactions produce the structure and modifications of structure that we find in the patterns exhibited by the physical world. Causal laws govern the causal processes and causal interactions, providing the regularities that characterize the evolution of causal processes and the modifications that result from causal interactions’ [39, p. 132]

The concept of causal processes is not, however, very clear; many of their features have to be defined with more precision. Those features include mark transmission, structure transmission, the principle of causal influence, and causal interaction. The original Salmonian approach has been considered in the light of deep knowledge, robustness and domain modeling in [5] and[4]. In those articles the reasons for structural quantitative modeling and the use of the Salmonian approach are discussed, and for that discussion we refer to those articles.

Due to Dowe’s [12] criticism Salmon [40] changed his theory in a way that makes Dowe’s criticism void. In that paper Salmon outlined the following propositions for causality [40]:

1. A *process* is something that displays consistency of characteristics.

2. A *mark* is an *alteration to a characteristic* that occurs in a single local intersection.

3. A mark is *transmitted* over an interval when it appears at each spacetime point of that interval, in the absence of intersections.

4. A *causal interaction* is an intersection[6] in which both processes are transmitted beyond the locus of the intersection.

5. In a causal interaction a mark is introduced into each of the intersecting processes.[7]

6. A causal process is a process that can transmit a mark.

From the above propositions Salmon did rephrase the definitions. Some of the rephrasing was done according to Dowe’s criticism [12] and other parts had been presented in [40]. The first new definition is:

Definition 1 A *causal interaction* is an intersection of world-lines which involves exchange of conversed quantity.

Definition 1 is a substitute for Salmons original definition of causal interaction (*CI*) used in [4].

[6] In Salmon’s vocabulary an “interaction” is always a causal one, but “intersections” can be causal or non-causal.

[7] This can be construed as a definition of “introduction of a mark”.

Definition 2 A *causal process* is a world-line of an object that transmits a nonzero amount of an invariant quantity at each moment of its history (each spece-time point of its trajectory).

Definition 3 A process *transmits* an invariant (or conserved) quantity from A to B ($A \neq B$) if it possess this quantity at A and at B and at every stage of the process between A and B without any interactions in the half-open interval $(A, B]$ that involve an exchange of that particular invariant (or conserved) quantity.

Salmon proposed that scientific explanations should use the concept of causal processes as the basis, and that actual explanations are linguistic descriptions of chains of causal interactions and mark transmissions. Note that the Salmonian approach to causality pays no attention to the distinction between discrete and continuous systems. In the following section we will consider how the Salmonian concepts may be used as the theoretical foundation on which causal models could be based.

4.2.3 From the Salmonian approach to implementable models

The existence of a theory is not enough when the purpose is to develop an approach to building computer models for simulation. The next step is to develop a way to represent knowledge, which is constructed according to the theory in such a manner that it can be expressed in a computer program. In the case of the Salmonian theory of the causal structure of the world it is tempting to adopt the most obvious way to represent causal processes. Such an approach may use variables to represent the characteristics of processes and procedural or functional structures of representing the interactions between different processes. In this subsection we will briefly outline how we will build models based on Salmonian concepts, for a more philosophical discussion we refer to [4].

In order to allow processes to be divided into subprocesses we introduce the concept of *levels* to our models. The subprocesses of a particular process are on lower levels than the process itself. Because causal processes have to be *named* (if not for any other reason, then to make the life of the modeler easier), we can consider the names of the processes to form a network in which pointed arcs start from the processes on upper levels and stop to the processes on lower levels. Note, however, that a particular name can represent a subprocess common to two or more processes on upper levels, like the atoms in different models of chairs. This can be formalized as

$$M = (P, A)$$

in which P is the collection of the processes and A is a collection of ordered pairs of those processes. The 'arc' from name P_i to P_j can go through more than one level. Since in a model of e.g. a forest there very probably is more than one tree of a type, we have to consider the names of the processes as discussed above to denote *types* of processes.

In the definition of causal processes Salmon used the term *characteristics* of processes. In a computer model such characteristics are represented by a set of variables and their values. If the values of the variables change due to a change in the space-time coordinates of the process, then the characteristic of a process includes the function of that change in addition to the variables themselves. Now a process P_i can be defined as a tuple

$$P_i =< N_i, V_i, F_i, S_i >$$

in which V_i is a set of variables used to represent the process, N_i is the name of the process, F_i is a set of functions used to determine the space-time related changes, and S_i is the set of the subprocesses of the process P_i.

Between processes there are causal interactions. Such interactions can be defined between the process types. Now we can define a model as a pair

$$M = (P, I)$$

in which I denotes the set of causal interactions. The set of causal interactions can be defined as

$$I = \{< p_i, p_j, c_{i,j} > | p_i \in N; p_j \in N\}$$

in which $c_{i,j}$ denotes the effects of the causal interaction between processes p_i and p_j on both processes, and N is the set of the names of processes, i.e. $N_k \in N$.

At this point we have defined a kind of "skeleton" for representing Salmonian models. In the following section we will briefly consider the traditional approach to modeling and connect that approach to our skeleton in order to produce usable models.

4.3 Modeling with causal processes

From the normal knowledge representation literature it is possible to get the impression that the greatest part of interest is in different technical aspects of the developed approach and not so much in the philosophical approach to domain knowledge and the combined benefits of that approach and its technical incarnation. Unfortunately the same problems seem to plague traditional simulation literature, see e.g. [43], [20], and [29]. It is interesting to note that according to Kiviat [23] there have lately been serious discussions about the relative benefits of the use of C or Pascal as a modeling language. This is quite astonishing. We will, however, concentrate on the theoretical part of the issue.

Now we will approach the creation of our causal process modeling approach by starting from the conventional approach to the modeling project. Normally the modeling project has been considered to be defined as the following steps [48]:

1. the "real system", a source of potentially observable data;

2. the "experimental frames", a set of limited observation or manipulation cases for the real system;

3. the “base model”, a comprehensive model of the system in every experimental frame;

4. the “lumped model”, a simplified version of the base model, which is simplified in a way that still provides the robustness of the original base model in interesting cases.

5. the “computer model” which implements the base model in a computer programming language on a machine.

We will not consider the origins of the above steps. For an interested reader we recommend the original book [48].

After comparing the modeling project steps to our formalization of the Salmonian models, it is obvious that our formalization represents the base-model step. Although the formalization can be used to represent the base-model, it does it in a theory-like way. In such a theory-like model the real system(s) and experimental frames are combined into a Salmonian theory of the causal structure represented as the base model. In that sense our third step is modified to a theory representation step during which a causal theory of the real system is represented according to the Salmonian formalism.

Obviously the computer model is even more clearly an abstraction than the base model mentioned above. The interesting thing in the abstraction-related nature of computer models is that very often the abstractive nature has been taken so for granted that traditional modeling literature has not paid enough attention to the possibility of developing methods to shorten the gap between the real world system (or phenomenon) and the computer model of the system, although AI research has been interested in implementing causal and structural knowledge into models in order to do just that. In order to shorten the gap between normal modeling and AI modeling the traditional modeling steps should be changed in a way which reflects the changed approach to domain knowledge. Hence the transition from the base model to the lumped model should not be done at the expense of the accuracy of the model.

Because our third step is modified into a theory representation step, the previous two steps have to be reconsidered. The first step does not require any modifications, but the second one is now transformed into a *domain theory formation* step. Note, however, that in this approach we do not pay any attention to the role of theories in the choice of the real system to be modeled, the construction of the “experimental frames”, the choice of the data to be observed etc. Discussion of those roles of theories are beyond the scope of this paper and that discussion is left to thosee who are better educated in these matters. Although the theory formation step is very important, we will not propose any method for theory formation, we will only recommend philosophical literature of the formation of scientific theories for interested readers.[8]

The next step, the fourth step above, is now changed into a *model formation* step in which the general *theory* of the domain (developed in the

[8] Lakatos has written very good articles about research programmes etc, and those articles also include important hints for theory formation. See [28].

previous step) is used to generate a working model of the system. This step involves the change from a generalized domain theory (like Newtonian mechanics) to a case-oriented representation of the modeled system. In such a representation the modeler has to decide how he/she will use the domain theory in order to produce the actual simulation of the modeled case.

In order to make that transition more obvious, we will consider the following example (which is, in fact, used through this article). Let us assume that we intend to simulate a pine. (Note that the considerations of the actual structure and workings of pines presented in this article are, due to the lack of decisive biological knowledge, very probably erroneous and wrong to a biologist or a forest researcher, but the model of a pine is used because it provides a complex enough example to outline the method even with a very naive pine-model.) After some consideration it is possible to find out that in a pine there are at least three different types of causal processes, namely:

- needles,
- shoots, and
- buds

which could be used to represent *any* pine. Note that the reproduction mechanism and parts below the ground surface have been left out for simplicity's sake.

Generally, in any pine, the growth of the tree proceeds according to the following cycle:

1. a shoot grows a bit, and new needles grow to the growing shoot;
2. after a shoot has grown a while, buds grow to the top of the shoot, and no new needles appear to the shoot and the shoot does not grow length any more after its buds have grown (the shoot may still grow thicker, however), and a shoot dies if it has no needles and no younger shoots connected to it;
3. a bud changes into a tiny shoot, and the cycle starts again.

After looking at pines it must be noted that generally step three precedes our first step, and that the climate seems to dictate when the steps occur. From this we can identify at least one causal interaction — the interaction between some aspect of the "climate" and the pine (such an interaction is obvious from our knowledge of plants: a plant requires heat and light in order to grow and go on living).

The ground can be considered to provide growth potential to the tree which reduces that potential near itself. Now we have identified a new interaction between our tree and "ground".

The needles get heat and light from "climate" and the potential from the shoot the particular needle is connected to. Similarly the needle provides the shoot with growth energy. Shoots transfer energy and potential to each others (potential goes upwards and energy downwards) and provide growth energy to the bud. Now we have identified several simple interactions and processes for our model.

According to the formalism of Salmonian models our theory of pines could be expressed as:

$$M_{\text{pine}} = (P, I)$$

in which

$$\begin{array}{rcll} P_{\text{pine}} & = & \{ & < \text{shoot}, V_{\text{shoot}}, F_{\text{shoot}}, \{\} > \quad , \\ & & & < \text{needle}, V_{\text{needle}}, F_{\text{needle}}, \{\} > \quad , \\ & & & < \text{bud}, V_{\text{bud}}, F_{\text{bud}}, \{\} > \quad \} \end{array}$$

and

$$\begin{array}{rcll} I_{\text{pine}} & = & \{ & < \text{shoot}, \text{needle}, \text{potential+energy transfer} > \quad , \\ & & & < \text{shoot}, \text{bud}, \text{energy transfer} > \quad \}. \end{array}$$

From the pine model it should be noted that there really is no interaction between *all* the subprocesses of the pine. Similarly the above model pays no attention to the interactions between the climate and the needle or between the ground and the pine.

In order to model the interaction between the ground and the pine we may introduce the model of “stand” in which

$$\begin{array}{rcll} P_{\text{stand}} & = & \{ & < \text{pine}, V_{\text{pine}}, F_{\text{pine}}, \{\} > \quad , \\ & & & < \text{ground}, V_{\text{ground}}, F_{\text{ground}}, \{\} > \quad \} \end{array}$$

and the potential-transfer occurs between the processes “pine” and “ground” instead of occuring between a “shoot” and the “ground”.

Because we consider the causal interaction between “ground” and “pine” to appear between a more or less homogeneous “ground” and “pine”, we will consider our “needles” etc to be *subprocesses* of “pine”. Now the processes “needle”, “shoot”, and “bud” are on a lower level than “pine” and “ground”. In other words, the processes in P_{pine} constitute the process “pine” in the process “stand”, i.e.

$$\begin{array}{rcll} P_{\text{stand}} & = & \{ & < \text{pine}, V_{\text{pine}}, F_{\text{pine}}, P_{\text{pine}} > \quad , \\ & & & < \text{ground}, V_{\text{ground}}, F_{\text{ground}}, \{\} > \quad \} \end{array}$$

and similarly

$$\begin{array}{rcll} I_{\text{stand}} & = & \{ & < \text{shoot}, \text{needle}, \text{potential+energy transfer} > \quad , \\ & & & < \text{shoot}, \text{bud}, \text{energy transfer} > \quad , \\ & & & < \text{pine}, \text{ground}, \text{the transfer of potential} > \quad \}. \end{array}$$

Although the relation between M_{pine} and M_{stand} is that the model of the stand *includes* the model of the pine (this can, obviously, be used to provide modularity), i.e.

$$M_{\text{stand}} = (P_{\text{stand}}, I_{\text{stand}}).$$

when P_{stand} is as above. The transfer of potentiality from “ground” to an individual “needle” is not clear yet, as no interaction between processes which belong to P_{pine} and P_{stand}, i.e. interactions between levels have been defined. In order to allow that transfer to happen we have to remember that an individual process has a set of variables which are used to represent its characteristics. Due to this, we may consider the potentiality to be transfered to the “pine” and from the “pine” to the “shoot”. Now we have, however, to add one interaction to I_{stand} and now

$$\begin{array}{rclc} I_{\text{stand}} & = & \{ & < \text{shoot}, \text{needle}, \text{potential+energy transfer} > & , \\ & & & < \text{shoot}, \text{bud}, \text{energy transfer} > & , \\ & & & < \text{pine}, \text{ground}, \text{the transfer of potential} > & , \\ & & & < \text{pine}, \text{shoot}, \text{the transfer of potential} > & \}. \end{array}$$

It is much more problematic to introduce “climate” to the model. For the sake of consistency we should have “climate” on the same level as “ground”, but that might present new problems because there is no interaction (at least according to our thinking as seen later on) between “climate” and “pine”. Ignoring that we now have

$$\begin{array}{rclc} P_{\text{stand}} & = & \{ & < \text{pine}, V_{\text{pine}}, F_{\text{pine}}, P_{\text{pine}} > & , \\ & & & < \text{climate}, V_{\text{climate}}, F_{\text{climate}}, \{\} > & , \\ & & & < \text{ground}, V_{\text{ground}}, F_{\text{ground}}, \{\} > & \} \end{array}$$

and

$$\begin{array}{rclc} I_{\text{stand}} & = & \{ & < \text{shoot}, \text{needle}, \text{potential+energy transfer} > & , \\ & & & < \text{shoot}, \text{shoot}, \text{potential+energy transfer} > & , \\ & & & < \text{shoot}, \text{bud}, \text{energy transfer} > & , \\ & & & < \text{needle}, \text{climate}, \text{energy transfer} > & , \\ & & & < \text{pine}, \text{ground}, \text{the transfer of potential} > & , \\ & & & < \text{pine}, \text{shoot}, \text{the transfer of potential} > & \} \end{array}$$

in which the energy transfer between “climate” and “needle” means both light and heat. For modularity reasons the interactions “shoot-bud”, and “shoot-needle” could be included in M_{pine} and not in M_{stand}, but for simplicity’s sake they have been defined in M_{stand}.

At this point we will not specify when a “bud” decides to grow a new “shoot” during the spring. Being a naive modeler we will consider it to have at least two different alternatives:

1. there is an interaction between “climate” and “bud”; or
2. growth starts when the energy transfer between “bud” and “shoot” can happen, i.e. after “needle” has been able to produce energy.

The second alternative is intuitively much more tempting, although the first alternative is the one used. We will, however, return to this later on.

At this point we have developed the lumped model of our “pine”. (Note, however, that all the actual variables used to represent a process and functions associated with the characteristics of processes have been left out of our considerations. We believe it reasonable to leave such details out of the example model because at this point the example is not intended to produce a real model which could be used in forest research, but rather as an illustrative example of our modeling approach.) Currently we have completed the act of theory representation related to the third of the model building steps, we should proceed to the fourth step, the model formation step.

In the third step we fitted the domain theory to the modeling approach. Now we should consider how an actual model could be expressed according to the modeling approach. In addition to the domain theory, there should be a method to simulate multiple “pines” at a time by using the same domain theory. In order to achieve that, we introduce the concept of *working model* to our approach. A working model WM is a pair

$$WM = (M, A)$$

in which M is a model as above and A is a set of *actualizations* or instances[9] of the processes modeled in M.

In order to enable the creations of those actualizations the modeling system has to provide some operators to create and destroy those actualizations. In other words, there should be a *create-process* operator to create processes. In addition to the process creation operator there should be an operator for process destruction.

The operators create and delete are required for setting up the simulation and for doing the destruction or creation of certain types of processes prior to or during the simulation. Such operators (and other operators) could be included as

1. a separate control section to coherently handle the creation, deletion, and other things done to the actualizations; and
2. integrated parts of the descriptions of the processes.

The inclusion of a separate control to the working model should be an easy task, it would require only one additional part to the definition of WM, which now is

$$WM = (M, A, C)$$

in which C stands for a *control block*. The inclusion of the operators in the definitions of the processes in M can be done by allowing the F of the process definition to include such operators.

At this point we may collate our definition of the model structure to a single representation. A *working model*, WM, is a model which is ready to be implemented. Such a working model is a triple $WM = (M, A, C)$, as above, in which M denotes the model of the domain theory, A denotes the set of possible actualizations of processes modeled in the domain theory, and C denotes a sequence of control operations performed in order to have the working model behave as intended. The *domain theory model* is a pair $M = (P, I)$ in which P is the set of definitions of individual processes of the model, and I is the set of definitions of the causal interactions modeled. Now a process P_i can be defined as a tuple

$$P_i =< N_i, V_i, F_i, S_i >$$

in which V_i is a set of variables used to represent the process, N_i is the name of the process, F_i is a set of functions used to determine the space-time related changes, and S_i is the set of the subprocesses of P_i. The set of

[9] Note that the term *instance* is not used in exactly the same manner as it is used in object-oriented programming.

causal interactions can be defined as

$$I = \{< p_i, p_j, c_{i,j} > | p_i \in N; p_j \in N\}$$

in which $c_{i,j}$ denotes the effects of the causal interaction between processes p_i and p_j on both processes, and N is the set of the names of processes, i.e. $N_i \in N$. Note that p_i can be of the same process type as p_j. Note, also, that the causal process modeling approach makes no distinction between discrete and continuous simulation. This enables a wider usage of the approach.

After the above modifications to the original modeling process steps we have got the following steps instead the originals:

1. the “real system”, a source of potentially observable data, the real system is chosen according to existing scientific theories;

2. the “domain theory formation”, during which the theory of the domain is created either by using pre-existing knowledge or by experiments and observations;

3. the “domain theory representation”, during which a comprehensive model of the system according to the theory is formalized following the Salmonian approach to the scientific knowledge;

4. the “working model definition”, the domain theory representation added with the considerations of how the actual system works in the accuracy of actualizations and the chosen time-frame;

5. the “computer model” which implements the working model in a computer programming language on a machine. For us this means the implementation of the working model in a special causal process programming language.

Above the modeling steps have been redefined according to the causal process modeling approach. Although the actual implementation technique has not yet been defined, the developed approach to the model definition seems to be able to produce the benefits of modularity by providing a way in which a process can include subprocesses i.e. construct modules (see [8] about the modularity in process modeling), in addition to a conceptually coherent approach to domain knowledge. In the next section we will consider how to make the theory practically usable.

4.4 Programming with causal processes

In the earlier section we considered a simple way of representing Salmonian causal processes. In this section we will develop a system for representing causal processes, implementing them as programs, and we will consider a very simple example of the adaptability of the modeling system by implementing a simple model of Scotch pine. Note, however, that the model of the pine is not intended to be an *exact* model of a pine as an exact model would be defined in forest research — the model is intended to be used as an example of the power of the modeling approach.

Of the generally known knowledge representation techniques, two seem to have some promise of being useful for the implementation of Salmonian models. Those techniques are frames and object-oriented programming.

The first technique, *frames*, proposed by Minsky [31] may be one of the most powerful knowledge representation techniques developed. A frame is thought to represent a set of stereotyped situations represented by a network of nodes and relations. This is, at first glance, very promising, but our concepts of setting up the simulation and instances of models do not seem to be very well suited for frames because simulations are *dynamic* and *changing* over the simulated period of time. That leaves the object-oriented programming approach.

From the object-oriented point of view we may consider our theory-level models to be the definitions of classes and runtime models to be instances. This is not, however, completely satisfactory because objects are *event driven* and causal processes are *time driven*. In the world of objects (like in an object-oriented program) an *event* causes something to happen in the object in question, and the object may send more messages (events) to other objects which will react to those events. But in the world of causal processes the driving factor is *time*, not a set of events. Hence we have to define a new approach to the implementation of causal process models. In the following subsection we will outline one approach to the implementation of our domain models.

4.4.1 Implementing causal process models

The conceptual structure of causal process models has been defined above, and in order to be able to implement such models we have to develop either an approach to the implementation using conventional programming languages like FORTRAN, C, or Pascal, or develop a new programming language to be used for the implementation of such models. We chose to develop a simple hybrid language for implementation.[10]

The working model included three parts: control block, process models, and actualizations. If we consider the model to be defined as one program text, the model implementation program is like:

- one or more process definitions;
- a control block definition.

As a syntactical definition this could be expressed as

```
WORKING_MODEL := {PROCESS_DESCRIPTION}+
                 CONTROL_DESCRIPTION
```

in which `{...}+` denotes one or more `PROCESS_DESCRIPTION`. Because the original working model definition proceeded from the domain theory defi-

[10] By a hybrid language we mean that the language includes features of the host language on which the implementation language has been built.

```
PROCESS_DESCRIPTION :=
        'A PROCESS' NAME
        'VARIABLES' NAMELIST
        'CHARACTERISTICS'
        'INITIAL'
        '(' INITIAL_CHARACTERISTICS_DESCRIPTION ')'
        'CHANGING'
        '(' CHANGING_CHARACTERISTICS_DESCRIPTION ')'
```

Figure 4.1: Syntax for process implementation

nition to the control block definition, we will consider the implementation of processes first.

As above, a process is principally a collection of variables with functions to define the characteristics of the process. In other words, $P_i =< N_i, V_i, F_i, S_i >$ requires a simple syntactical structure as in Figure 4.1 in which the words between single quotes denote reserved words which are parts of the implementation language. After 'A PROCESS' comes the name of the process, i.e. NAME is N_i, the set of the names of the variables comes after 'VARIABLES', and the definition of characteristics comes after 'CHARACTERISTICS'.

When a process is created it has to have initial characteristics. The characteristics implementation has, therefore, to be divided into two parts: the initial characteristics implementation, and the implementation of the definition of the characteristical changes of the variables of the process. Although the characteristics have originally been developed as a set of functions, it is much more convenient to allow modelers to implement those characteristics as program code affecting the values of the variables. No implementation language for that code is defined because currently the host language is used to implement such codes, as seen in the following subsection.

In order to make the work of the model implementer easier, the description of causal interactions should be included in the textual structure used to define the process on which they have their effects. For this we add the structure

```
'INTERACTIONS' INTERACTIONLIST
'INTERACTION DEFINITIONS' INTERACTION_DEFINITION_LIST
'INTERACTION EFFECTS' EFFECTS_LIST
```

to the process implementation syntax in Figure 4.1. The term 'INTERACTIONS' is followed by a list of the names of the causal interactions which have their effects on the process being implemented and the other process. Note that a causal interaction is defined only *once*, i.e. the interaction between processes P_i and P_j is implemented in the description of either P_i or P_j, but *not* in the description of both processes. The method by which the description of a causal interaction in the description of the process P_i has its effect on the process P_j is an implementation dependent choice which is not relevant to the actual approach in general. This is an implementation oriented difference to the original definition of working models. The names are necessary for the identification of the processes and for the internal workings of the programming environment. The interaction definition list is a list defining with which processes the process interacts. This is also for the internal use of the simulation environment.

In addition to the above structures the relation between different processes should somehow be expressed. This is achieved by using the notation

```
'CONSISTS OF' '(' NAMELIST ')'
```

in which the term 'CONSISTS OF' is followed by the names of the subprocesses. For computational reasons the process description syntax defined in Figure 4.3 includes the definition

```
'EVALUATION TYPE' EVTYPE
```

which is used to enable the hiding of computations in order to achieve better simulation speed. The alternative of choosing between the use of the speeded-up method and exact computation is given to the developer because sometimes it may be necessary to compute exact values for *every case*. The reserved word 'IMPLICIT' means that the simulation will not be done if the result can be estimated. The word 'EXPLICIT' means that the simulation will be performed *every time*. A method to achieve such speed-up is proposed in [2].

The description of the control block is a fairly easy task. In our simple process implementation language CONTROL_DESCRIPTION is defined as

```
CONTROL_DESCRIPTION := 'CONTROL'
                        TARGET_LANGUAGE_EXPRESSION
                       'ENDCONTROL'
```

in which TARGET_LANGUAGE_EXPRESSION denotes a lawful set of sentences of the host language, i.e. the language which is used as the intermediate phase between the executable simulation model and the process implementation description, including predefined simulation and controlling commands which depend on the chosen intermediate language. This is, however, only a temporarily phase before a complete causal process implementation language is defined. Such a complete language should include enough syntactical and semantical power to express all functions and control commands required for the implementation of simulation models, but currently the approach has been tested by using a hybrid between Smalltalk/V and the implementation language.

Although the actual syntax (and actual commands) used in the control block depend on the intermediate language, at least the following operations should be implemented:

- simulation initialization;
- start time, end time, time step etc, — time related settings;
- report generator initialization;
- process creation and deletion;

```
CONTROL
SimulationInstance initializeSimulation: [
        SimulationInstance endTime: 96.
        FC stepsInYear: 12.
        SimulationInstance startTime: 1.
        SimulationInstance timeStep: 1.
        SimulationInstance time: 1.
        Definer initializeDumpStream.
        SimulationInstance addProcess: 'climate'.
        SimulationInstance addProcess: 'ground'.
        (SimulationInstance addProcess: 'pine')
                placex: 1000; placey: 1000; energy: 7000].

SimulationInstance startSimulationExecution: [
        SimulationInstance dump: 'pine' to:
        (Definer provideDumpStream) when: [(TIME rem: 6) = 0].
        SimulationInstance dump: 'shoot' to:
                (Definer provideDumpStream) when: [(TIME rem: 12) = 0].
        SimulationInstance dump: 'bud' to:
        (Definer provideDumpStream) when: [(TIME rem: 12) = 0].
        ]        runCheck: [(TIME rem: 1) = 0]
        ifNot: [:proc | (((((FC computeClimate: TIME) at: 1) < 0)
                    and: [(proc = 'Bud') or: [proc = 'Needle']]))].
ENDCONTROL
```

Figure 4.2: A control block example

- process variable valuation;
- reporting commands; and
- possible simulation controlling commands (which are, of course, implemented only in interactive modeling environments).

Several examples of the usage of these commands can be seen in Figure 4.2 in which the control block of the working model implementation of the pine model is listed.

The control block in Figure 4.2 is the control block used for simulation control for our pine example. Because Smalltalk/V was used as the implementation language for our programming environment (briefly described in the following subsection), some knowledge of Smalltalk may be required in order to make the understanding of the example implementation easier. The first part of the control-block in Figure 4.2 performs the initialization of the simulation. This is done by a special command defined for that purpose. The initialization is done by sending a program block to the global variable `SimulationInstance`, which performs the actual simulation management. In the simulation initialization the timing settings are set and the necessary processes created with suitable values for their variables.

In the second block (a Smalltalk block) the actual simulation is started with several parameters. The first parameter, a program block, includes the reporting-oriented code, the second is a timing function used to handle the connection between direct user control and the simulation, and the third block is used to determine in which cases it is computationally useful to skip the simulation execution. This is an implementation dependent choice before a completely featured modeling language has been defined.

```
WORKING_MODEL := {PROCESS_DESCRIPTION}+ CONTROL_DESCRIPTION

PROCESS_DESCRIPTION :=
                'A PROCESS' NAME
                'CONSISTS OF' '(' NAMELIST ')'
                'EVALUATION TYPE' EVTYPE
                'VARIABLES' NAMELIST
                'CHARACTERISTICS'
                'INITIAL'
                '(' INITIAL_CHARACTERISTICS_DESCRIPTION ')'
                'CHANGING'
                '(' CHANGING_CHARACTERISTICS_DESCRIPTION ')'
                'INTERACTIONS' INTERACTIONLIST
                'INTERACTION DEFINITIONS' INTERACTION_DEFINITION_LIST
                'INTERACTION EFFECTS' EFFECTS_LIST

NAME :=         <a character string of alphanumeric characters>

NAMELIST :=     NAME | ' ' | NAME NAMELIST

EVTYPE :=       'EXPLICIT' | 'IMPLICIT'

INITIAL_CHARACTERISTICS_DESCRIPTION :=
                TARGET_LANGUAGE_EXPRESSION

CHANGING_CHARACTERISTICS_DESCRIPTION :=
                TARGET_LANGUAGE_EXPRESSION


TARGET_LANGUAGE_EXPRESSION := {TARGETEXPRESSION}*

TARGETEXPRESSION :=
                 <a lawfull expression in the target language>

INTERACTIONLIST :=
                 NAME NAMELIST

INTERACTION_DEFINITION_LIST :=
                 (NAME NAMELIST '(' NAME ')')+

EFFECTS_LIST :=  (NAME '(' 'IF ' '(' TARGETEXPRESSION ')'
                           'DO' '(' TARGET_LANGUAGE_EXPRESSION ')' ')')+

CONTROL_DESCRIPTION := 'CONTROL' TARGET_LANGUAGE_EXPRESSION 'ENDCONTROL'
```

Figure 4.3: Implementation language syntax

```
A PROCESS  climate
        CONSISTS OF
                none
        EVALUATION TYPE EXPLICIT
        VARIABLES
                heat light a1
        CHARACTERISTICS
                INITIAL
                (
                heat := 0. light := 0. a1 := 0.
                )
                CHANGING
                (
                a1 := FC computeClimate: TIME.
                heat := a1 at: 1.
                light := a1 at: 2.
                )
        INTERACTIONS
                none
        INTERACTION DEFINITIONS
                none ( nil )
        INTERACTION EFFECTS
                none ( IF   ( true )
                       DO ( nil )  )
```

Figure 4.4: Implementation of a process without interactions

The complete syntax of the simple working model implementation language is in Figure 4.3. From the complete syntax it is obvious that there are some problems with processes to which the modeler may not want to implement the interactions. An example of such a process is in Figure 4.4, in which `none` is a reserved word used to represent a nonexistent process and `nil` is a similarly reserved word used to denote nothingness.

The problem of denoting nothingness will, amongst other semantical issues, be covered in greater detail in the subsection *An example: a causal model of pine*. In the next subsection the experimental programming environment developed for the implementation of causal models will be briefly described.

4.4.2 Programming Environment for Causal Process Modeling

Now we have defined how to implement causal process models as computer programs. In order to make the implementation easier we have developed a programming environment for model development and simulation. Currently the system is implemented using Smalltalk/V under the OS/2 2.x operating system.

The programming system consists of roughly three parts which are: the user interface, the process representation compiler, and the simulation manager. Some parts of the programming environment may not be completely according to the causal process approach to modeling, but those differences are due to the practical consequences of the environment which has been used for the programming environment implementation.

The user interface part of the programming environment is in principle very simple. The user interface is currently text-oriented, the inclusion of graphics in the system is on the responsibility of the modeler. Graphics have been purposely left out because it is not yet clear what type of graphics would be the most productive tool in the model implementation. Due to the lack of knowledge concerning the most beneficial graphical tools no such tools or methodologies were implemented in the programming environment.

Principally the user-interface consists of three different windows which are the normal Smalltalk Transcript window which allows full access to the Smalltalk development environment, a message window used to display error messages, and a programming window which allows users to enter either process descriptions or control blocks. The Smalltalk Transcript was left as-is because it was decided during the design phase of the programming environment that it would be necessary to allow users to implement their own classes and develop their own interfaces and graphics in order to manipulate or visualize processes or their simulation.

The error window does not have any special functionality in addition to a few file oriented choices. The file oriented choices allow the user to save the contents of the window to a file. The error window is mainly used for displaying error messages and user-defined runtime or reporting messages. For the purpose of this paper there is no need to consider possible error messages and methods to add user-defined functions to the system.

The programming window constitutes the actual interface between the programming environment and the user. By using the choices present in the “Control” pull-down menu users can perform a variety of tasks. Some of those tasks are related to the actual simulations, like “Pause simulation”, “Begin simulation”, “Stop simulation”, and others are related to actual implementation of control blocks and processes. The compilation from the causal model implementation language to runnable Smalltalk code is done by the choice “Evaluate code” which starts the compilation of the implementation language expression. The actual implementation of the compiler is not described here because the implementation would be different in different implementations. In addition to the causal-process programming choices the programming windows provide the user with the normal Smalltalk/V Transcript features.

The process representation compiler is invisible to the user if the process implementation written by the user is correct. If there are errors, the error messages will be displayed on the error window. The implementation language compiler compiles the process type descriptions into Smalltalk classes which inherit methods used in simulation by the environment. The compiled representation is then compiled by using the internal compiling abilities of the Smalltalk environment.

Because working models are *time-driven* the actual simulation is done by incrementing the time counter by the chosen time step and then performing a simulation step. The simulation step is as follows:

```
FOR every process actualization DO
    perform the changing characteristics block
ENDDO
FOR every process actualization p DO
    FOR every actualization of the type T with which
        processes of the type in which p belongs can
        have interactions DO
        IF the condition part of the interaction is true THEN
            perform the interaction
        ENDIF
    ENDDO
ENDDO
```

Note that there has to be some parameter passing etc between actualizations during an interaction. This is achieved by introducing a programming environment implementation language dependent technique. That technique will be more closely explained from the process implementation side in the following subsections in which the implementation of the example model and some simulation results are presented.

4.4.3 An example: a causal model of pine

In this subsection the implementation of the example model, a simple pine model, will be described. Note that the model is *not* claimed to be an accurate model of a pine, it is used to briefly describe the causal process modeling technique. Although the model is not a very good model in the biological sense, the results from its use are surprisingly good as will be seen later on.

One of the processes defined in the example model is "ground", which may not *exactly* be a causal process in the Salmonian sense, although any material thing with substance constitutes a causal process. In that sense "ground" is a causal process. In the example implementation of the "stand" model the "ground" process has been implemented as shown in Figure 4.5.

In the implementation there are a few interesting implementation choices. Although the normal potentiality has been defined to be 10000, it is not changed in any way later on. This implementation shows how different parts of the model can be partially implemented during the modeling process. Although "ground" does not have all the defined interactions and characteristics, the behavior of the implemented model is surprisingly promising even without those features, as will be seen later on.

The implementation of "climate" is in Figure 4.4. The most notable feature of "climate" is that its characteristic features are used via causal interactions by other processes, as can be seen from the implementation of the "needle" type of processes. That implementation has been done as in Figure 4.6.

The "needle" implementation includes actual usage of interactions. There is an energy transfer from "climate" to "needle", and that interaction has been named `light_and_heat`. Parameter passing required in interactions between "needle" and "climate" is done by sending the message `heat` to a variable `climate` local to "needle" and receiving the value of the correspond-

```
A PROCESS  ground
        CONSISTS OF none
        EVALUATION TYPE EXPLICIT
        VARIABLES
                normal treeeffects
        CHARACTERISTICS
                INITIAL
                (
                normal := 10000.
                treeeffects := Dictionary new.
                 )
                CHANGING
                (
                true. "Do nothing"
                 )
        INTERACTIONS
                competitioneffect
        INTERACTION DEFINITIONS
                competitioneffect ( pine )
        INTERACTION EFFECTS
                competitioneffect ( IF   ( true )
                                    DO (
                                        true. "Do nothing in this version."
                                        )  )
```

Figure 4.5: Implementation of ground-process

ing variable `heat` from the “climate” process interacted with. This is enabled by the simulation system which automaticly creates required local variables for processes (one variable for each of the process types with which the current type may have interactions) and setting the process possibly interacted with as the value of the variable. Similarly the compiler automaticly creates methods for getting variable values of processes and setting them. Although causal interactions have, according to Salmon, effects on *both* causal processes which participate in the interaction, we see only single sided effects in the `light_and_heat` interaction. Although a causal interaction is, in any case, described in either of the processes, there may be no interactions defined even in the descriptions of processes which participate in interactions, but the two-sided effects of the interaction are implemented in the interaction description of the process types interacted with. But in the case of “climate” there are effects only on “needle”, not on “climate”. This seems like a violation to the definition of *CI* and requires closer consideration.

The violation is, however, in the implementation, not in the original domain theory. Obviously the transfer of heat and light from “climate” (light coming from sun and heat also, heat being in the air and light being photons and their energy) causes a similar part of the energy to be *removed* from “climate”. Because “climate” has an unlimited supply of energy (at least considering the naive model used as an illustrative example), it is unnecessary to *implement* such changes in the model. This is, in other words, a practical solution by which the implementation is made simpler, not a violation of the principles.

The original “stand” model consisted of three processes: “ground”, “climate”, and “pine”. Obviously the most complicated of those processes is “pine” which has to include code to create its subprocesses etc. That “pine”

```
A PROCESS  needle
    CONSISTS OF none
    EVALUATION TYPE EXPLICIT
    VARIABLES
        length x1 y1 z1 x2 y2 z2 consumption birthtime length
        age production light heat productionfactor
        overproduction
    CHARACTERISTICS
        INITIAL
        (
          age := 0. x1 := 0. y1 := 0. z1 := 0.
          x2 := 0. y2 := 0. z2 := 0. birthtime := TIME.
          production := 0. heat := 0.
          productionfactor := 1.
          length := 0.04. light := 0.
          consumption := 0. overproduction := 0.
        )
        CHANGING
        (
        (heat > 0)  ifTrue: [
                light := light - (upper shadowing).
                production :=  (FC  computeNeedleProduction:
                        productionfactor
                        heat: heat light: light
                        length: length).
                consumption := FC  computeNeedleConsumption: length.
                overproduction :=  (production - consumption) max: 0.
                (productionfactor < 0.15) ifTrue: [
                        self removeItself.
                        ^true].
                productionfactor := productionfactor *
                        (FC  randomNumberFrom: 0.75 to: 0.98).
                ].
        )
    INTERACTIONS
        light_and_heat
    INTERACTION DEFINITIONS
        light_and_heat (climate)
    INTERACTION EFFECTS
        light_and_heat ( IF   ( true )
                           DO ( heat := climate heat.
                                light := climate light. )  )
```

Figure 4.6: Implementation of needle-process

type has been implemented as in Figure 4.7. Note that in the implementations of the longer processes the actual code is replaced by English language sentences describing what has been done. Those sentences are marked by *.

The implementation of "pine" is clearly in an incomplete state. Currently it does not effectively implement the "fromground" interaction which should represent the transfer of potential (i.e. water and nutrients) from "ground" to "pine". Similarly it does not implement any reasonable competition computations for the competition between actualizations of "pine". Those interactions have not yet been fully implemented.

Inside the definition of the changing characteristics of "pine" there are some implementation oriented issues which are worth a closer look. If the tree has not yet created any shoots, then a new shoot is created. This is done by sending the message `delayedAddProcess: 'shoot'` to the `SimulationInstance`[11], which is a global variable used to enable the usage of required operands. The message is sent in order to create an actualization of "shoot". That message is used as the way in which actualizations of processes can cause other actualizations to appear. Note that the checking for the existence of previously created shoots has to be done by using an `ifTrue:` form. If there has been no previous actualization of "shoot", then one is created. It is problematic that the first "shoot" is created when the *changing* characteristics code is executed and not when the *initial* characteristics code is executed. That implies that a "pine" can exist before it has any "shoots".

The second subprocess of "pine" is "bud". The implementation of the "bud" type processes has been done as in Figure 4.8, in which there is only one interaction. That interaction is used only to cause the actualization of a "shoot" which replaces the "bud", i.e. the "bud" disappears after the actualization of the "shoot". The interaction between "climate" and "bud" denotes an energy transfer which causes the "bud" to grow and produce a "shoot". The actual energy transfer is not implemented due to reasons similar to the ones why the energy transfer between "needle" and "climate" has been implemented from one side only.

The "shoot" created by a "bud" (or a "pine") is implemented as in the figure 4.9. The implementation of the "shoot" type processes is obviously the most complicated in the example model. The complicated implementation is not so much due to the original definition of "shoot" but much more because it turned out that some parts of the definition had to be redefined in order to get any performance out of the testing environment. The functions etc required to compute all possible new values are not presented here because they are not very relevant to the actual usage of the causal process modeling approach.

[11] The `delayedAddProcess:` message causes a process actualization to be created in a way which makes it appear in the actual simulation cycle one time-step later. This is done to make it sure that the new actualization will not participate in causal interactions before all previously existing actualizations have been checked.

```
A PROCESS  pine
    CONSISTS OF needle bud shoot
    EVALUATION TYPE EXPLICIT
    VARIABLES
        age length competition volume mass initialenergy
        coordx coordy competitioneffect competition1 potential
        children
    CHARACTERISTICS
        INITIAL
        (
        competition := 0. initialenergy := 0.
        volume := 2*0.001*(Float pi).
        age := 0. length := 0. mass := 0.
        children := lower.
         )
        CHANGING
        (
        competition := competitioneffect.
        age := FC computeAgeChange: age time: TIME.
        competitioneffect := 0. competition1 := 0.

*       Compute the new maximun length of the tree. Can be done
*       by checking the biggest z2 from ``shoot’’s. This is a
*       violation to the original process approach, done for execution speed.
*       During that it is possible to get the masses etc from shoots and
*       needles.

*       If the tree has not been yet created any shoots, then create one.

        (lower isEmpty) ifTrue: [
            self deleteItself. ``No shoots left, the whole tree dies’’
            ^false].
        age :=  FC computeAgeChange: age time: TIME.
        mass := 0.
    )
    INTERACTIONS
        fromground competition
    INTERACTION DEFINITIONS
        fromground ( ground )
        competition ( pine )
    INTERACTION EFFECTS
        competition
            ( IF   ( (self = pine) not )
              DO (
*                Compute the competition related values. Not yet implemented.
            )  )
        fromground
            ( IF   ( true )
              DO (
                true. "DO nothing in this version."
                )  )
```

Figure 4.7: Implementation of pine-process

```
A PROCESS  bud
    CONSISTS OF
        none
    EVALUATION TYPE EXPLICIT
    VARIABLES
        x1 y1 z1 xk yk zk energyreserve thefirst
        p2 a a1 year summer budnumber
        alpha beta buds
    CHARACTERISTICS
        INITIAL
        (
        summer := 0.
        )
        CHANGING
        (
        (((FC computeClimate: TIME) at: 1) < 0.0001) ifTrue: [^false].
        ((upper ika) < 1) ifTrue: [^false].
        x1 := upper x2. y1 := upper y2. z1 := upper z2.

        alpha := upper alpha. beta := upper beta.
        thefirst ifFalse: [
*           Compute the new angles for the new shoot to be created.
            ].
        a := FC computeShootsLength: energyreserve.
        upper upper lower add:
                ((a1 := SimulationInstance delayedAddProcess: ’shoot’)
            length: a;
            x1: x1; y1: y1; z1: z1;
            beta: beta; alpha: alpha;
            x2: (x1 + (a * (beta sin)));
            y2: (y1 + (a * (alpha sin)));
            z2: (z1 + (a * (beta cos)));
            thickness: (FC computeShootThickness: a);
            runko: thefirst;  parent: upper;
            gives: (energyreserve - (FC computeShootConsumption: a1));
            upper: (self upper upper)); yourself.
        upper children add: a1.
        self deleteItself.
         )
    INTERACTIONS
        issummer
    INTERACTION DEFINITIONS
        issummer ( climate )
    INTERACTION EFFECTS
        issummer ( IF   ( climate heat > 0 )
          DO ( summer := 1. )  )
```

Figure 4.8: Implementation of bud-process

```
A PROCESS  shoot
    CONSISTS OF none
    EVALUATION TYPE EXPLICIT
    VARIABLES shadowing length x1 x2 y1 y2 z1 z2 energyreserve consumption
        thickness gives age dying needlenum i mass  noBudsYet alpha pillar
        a buds beta parent children birthtime
    CHARACTERISTICS
        INITIAL
        (birthtime := TIME. alpha := 0. beta := 0. shadowing := 0.
         x1 := 0. x2 := 0. y1 := 0. z1 := 0. z2 := 0. energyreserve := 0.
         consumption := 0. parent := nil. children := Set new.
         thickness := 0. age := 0. gives := 0. length := 0. dying := false.
         mass := 0. noBudsYet := true. pillar := false. needlenum := 0.
         a := 0. buds := 0.  )
        CHANGING
        (age := FC computeAgeChange: age time: TIME.
        dying ifTrue: [
*           If this shoot does not have needles or shoots growing from it,
*           then this shoot will be removed, i.e. it dies.
            ].
        ((FC isItBudGenerationTime: TIME) and: [noBudsYet])  ifTrue: [
                noBudsYet := false.
                buds := FC computeBudNumber: (gives / 2).
*               Create the new buds which will be given the starting
*               value for future growth. One of the buds will be marked as
*               the first bud from which a shoot will grow in the same
*               direction as the shoot from which it will grow.
                ].
        noBudsYet ifTrue: [ (needlenum < 1)
            ifTrue: [
*               If there is no new needles i.e. the shoot is a new one,
*               then create all those needles which belongs to it.
                ]
            ifFalse: [
*               Grow the shoot a bit more (if it can grow), compute new
*               coordinates for it and create the new needles.
            ].
*           Grow more thickness and mass for the shoot
            consumption := FC computeShootConsumption: self.
            lower do: [:each |
                ((each class) name = 'Needle') ifTrue:
                                    [gives := gives + (each gives)]].
            gives := gives - consumption. dying := (gives < 0.0).   )
     INTERACTIONS
        getshadowing energytransfer
     INTERACTION DEFINITIONS
        getshadowing ( shoot )
        energytransfer ( shoot )
     INTERACTION EFFECTS
        getshadowing
                  ( IF   ((self = shoot) not and: [z2 <= (shoot z2)] )
                  DO ( shadowing := shadowing +
                         (FC computeShadowing: self from: shoot). )  ))
        energytransfer
                ( IF ( (shoot = self) not and: [ (self parent =  shoot) or:
                        [shoot parent = self]])
                  DO (energyreserve := energyreserve + (shoot gives / 2).
                      shoot gives: (shoot gives / 2). ) )
```

Figure 4.9: Implementation of shoot-process

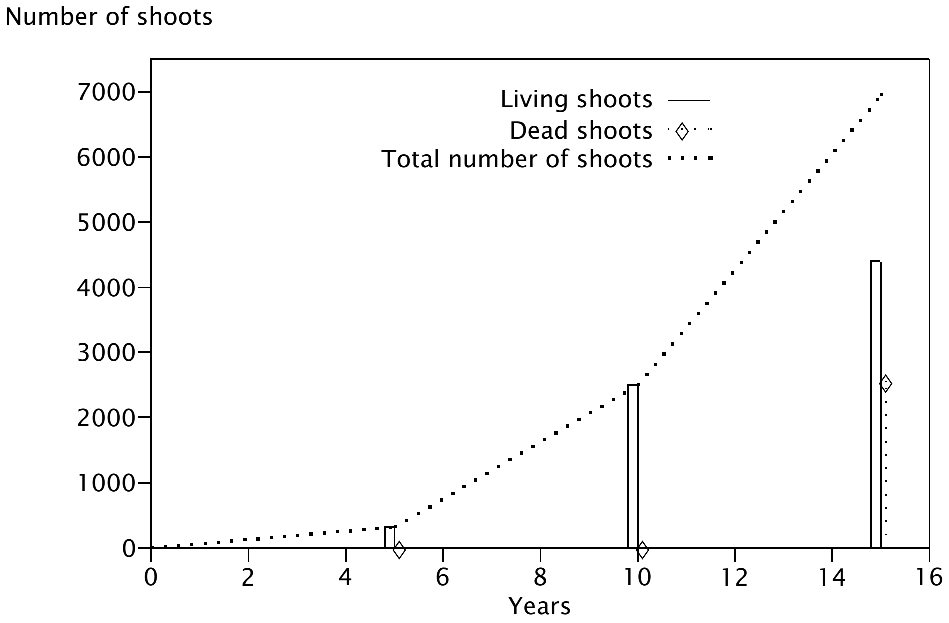

Figure 4.10: Field experiment numbers

4.4.4 A brief evaluation of the example model

In this subsection the workings of the example model implementation is briefly discussed. The most interesting feature of the implementation is that the simulation results are surprisingly good considering the simplicity and inaccuracy of the model. Although all the functions used in the model will not be discussed, some considerations of the behavior of the simple pine model will be presented. Because the model is not exact enough to provide complete values for all possible variables, only the number of generated "shoot"s and the length of the generated "pine" will be considered. First we will consider the number of the generated "shoot"s.

The analysis of the number of the generated "shoot"s shows that although the actual number of the generated "shoot"s is not the same, the overall shape of the diagrams in Figures 4.10[12] and 4.11 is very similar, although several differences should be noted. The first difference is that although the actual number of shoots in a tree is not the same, the shape of the curve representing the number of shoots is surprisingly similar.

In the implemented model, the dying ages of shoots have a very interesting pattern. From Figure 4.12 it is possible to see that in the experimental mode a great deal of the shoots have died very soon after their birth. This may imply that in a real pine such shoots are not created at all. If those shoots were not created in the model, then the growth rate of the number of the shoots might be much nearer the field-numbers.

[12] These numbers are from Professor Seppo Kellomäki, Faculty of Forestry, University of Joensuu.

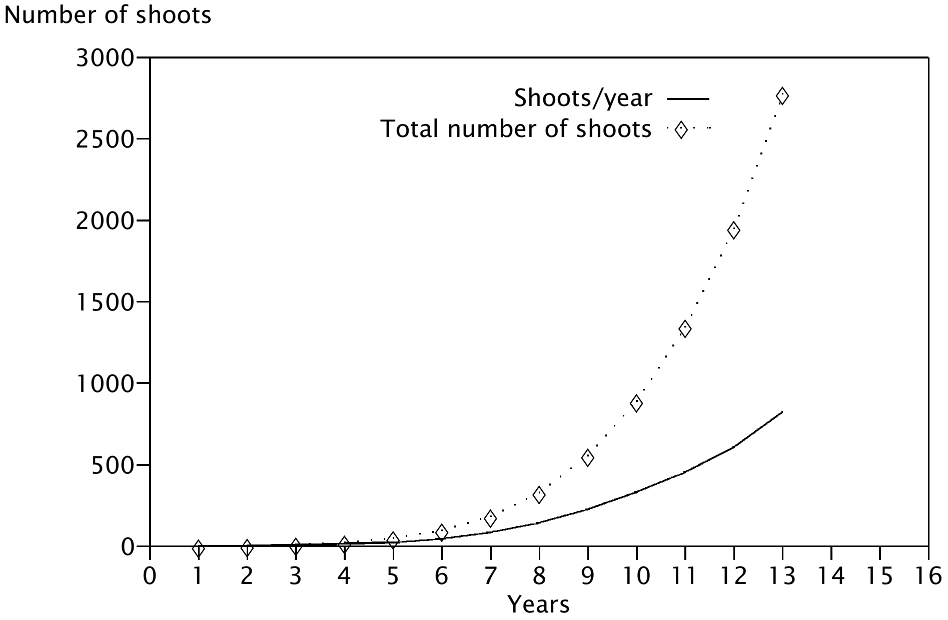

Figure 4.11: Diagram of the number of generated shoots

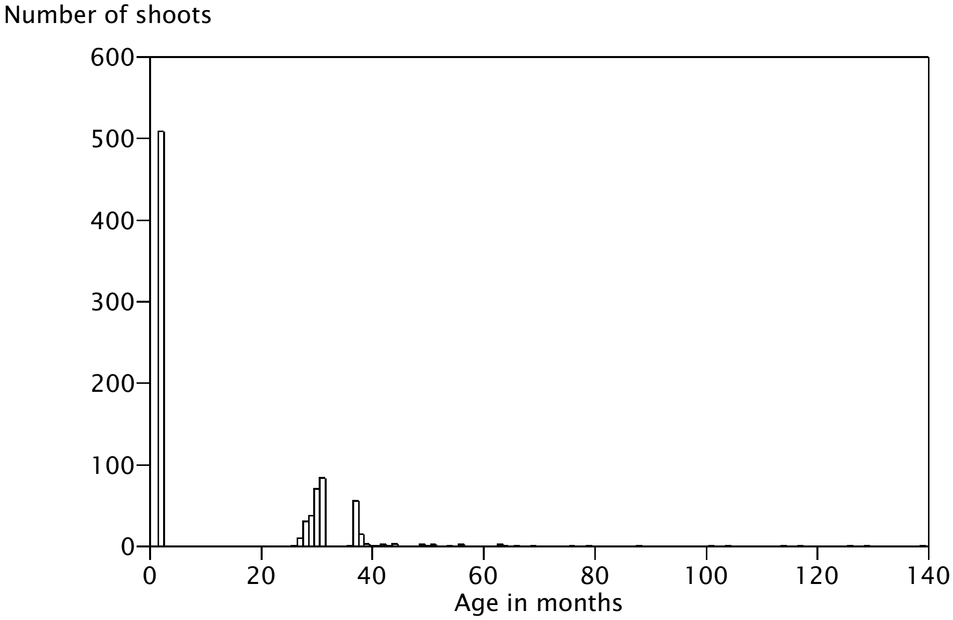

Figure 4.12: Dying ages of generated shoots

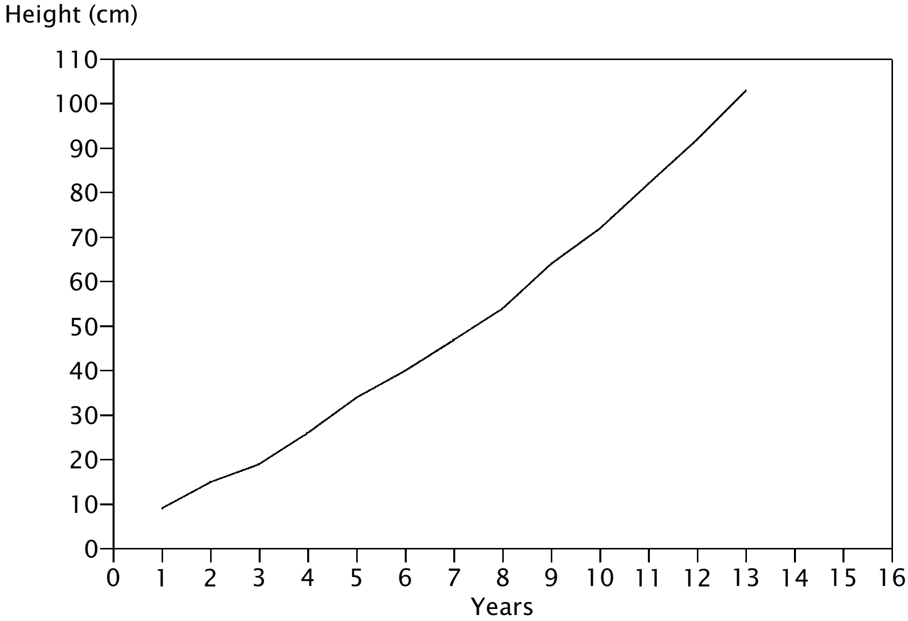

Figure 4.13: Simulated height growth

Considerations of the length growth of the simulated pine are not easy because the experimental implementation of the pine model did not turn out to be very effective in the computational sense. A single reasonably modest simulation (of only one pine) of thirteen years time (with a one month time-step) took about four days.[13] In order to collect enough data about the accuracy of the simulations, ten simulations were done[14] (note that the model includes several probabilistic features which make individual simulations differ), and computational inefficiency restricted the simulations to fairly short ones. Such short (thirteen years) simulations do not allow us to consider the long-term accuracy of the model, at least not using the current simulation environment. In Figure 4.13 the length is presented as a function of time. The curve follows the shape of experimental data presented in [34]. Unfortunately our current simulation does not reveal if longer simulations would also follow empirical data.

[13] The implementation machine was an Intel 486/33MHz based machine with 16 MB RAM. Not a very powerful machine in the computational sense. A much more faster implementation of a partly different version of the model was experimentally written in FORTRAN on Convex C3420, but those results are not analyzed here. Although the direct FORTRAN implementation shows it to be possible to implement causal process models by using conventional programming languages, the implementation proved that "by-hand" implementation of causal process models is very complicated and difficult. In order to enable faster creation of causal process models a complete implementation language should be defined and methods of compiling models written in that language into e.g. vectorizing and parallelizing FORTRAN code.

[14] All numbers used in the analysis of the simulation models are mean values of those simulations.

Generally the exponential curve representing the number of generated shoots does imply that the causal process modeling may easily produce models which expand very fast. That type of expansion is unfortunate in the computational sense, but seems to be a fundamental feature of causal process models of growth.

Although this simple analysis does not actually prove anything, it seems possible that with some modifications and additions the example model might be able to provide reliable simulation of the growth of Scotch pine.

4.5 Discussion

The developed approach to modeling and simulation differs from traditional approaches by its emphasis on the *causal structure* of domain knowledge in addition to the workings of the modeled object or phenomenon in qualitative models. The most remarkable feature of the developed approach is that it combines the usage of the structure, which is normally encountered in qualitative modeling, to *quantitative* results. The best differentiation between the original modeling approach and the developed approach is done in the section *Modeling with causal processes* above.

The developed approach has several features worth considering. Some of them are positive and some are negative. The positive features of the modeling approach could be summarized as:

- it provides a coherent approach to domain theories — an approach which is not restricted by the surface structure of a particular theory;
- it allows the modeler to use the same methodology throughout the modeling process;
- implementation is fairly easy; and
- it may provide a natural connection between models and scientific explanation, i.e. ease the task of explanation generation.

The negative features include the following ones:

- the usage of the approach may require serious restructuring of existing domain theories;
- successful use may require too much self-control from the modeler; and last but not least is the fact that
- models generated according to the methodology are easily very demanding in the computational sense.

Both negative and positive features are discussed in the following paragraphs.

Probably the most beneficial feature of the causal process modeling approach is its universal approach to domain theories. Assuming that the Salmonian theory of causality and the structure of the world is useful, it is possible to approach different domain theories from a meta-theory point of view. Such a meta-approach helps modelers uncover the underlying structure of superficially very different domain theory structures and individual domain theories. This type of uncovering should enable modelers to *combine* diverse domain theories into one domain theory whose use is planned for the actual model implementation.

Another very beneficial feature is the possibility of using the same domain theory modeling methodology throughout the whole model development process. The modeling project steps presented by Zeigler [48] do not pay any attention to methodology, they only form technical phases of the project. In order to combine all the steps under one methodology a general theory of the structure of domain theories has to be developed, and we believe that the Salmonian theory provides an acceptable general theory.[15] By means of the general theory we have been able to develop a modeling methodology which stays the same through the whole modeling process.

The ease of the implementation is relative, although it is reasonable to assume that the coherent approach to the domain modeling will ease the step from theory to implementation. Especially important in this is the possibility of using implementation techniques which are in accordance with the methodological and theoretical issues of the developed approach. We think that the causal process programming environment provides, even in its current crude form, an easy way to implement domain models created according to Salmonian concepts. This remains, however, only an assumption as long as no research on the cognitive effects of the usage of our approach has been done, and such research is outside the scope of this paper.

The connection between explanation generation in expert systems and the developed modeling approach is very promising. Because the modeling approach has itself been developed from the point of view of *scientific explanation* it is reasonable to think that it should be fairly easy to construct explanations from the implementations of causal process models. There are some considerations of the explanation generation in [3], where a logic based theory of explanation (due to [14]) is integrated to the Salmonian theories. Note that currently there is no explanation generation technique implemented in the modeling environment, and we feel that in order to enable unproblematic generation of explanations the modeling language should be extended to cover both explanation definitions and actual modeling. This is left to further research because the current system is developed only to show the usability of the developed modeling approach.

The first, and probably the worst, problem with the approach is that it may require severe restructuring of existing domain theories before they can be turned into implementable causal process models. This is a quite unfortunate effect which may complicate the actual modeling processes. This can, however, be reliably estimated after several practical modeling projects, and because the modeling technology has not been tested on various domains, this question is left open.

[15] This is, however, open to philosophical discussion. From our philosophical point of view the Salmonian theory is right and we will not consider its possible philosophical problems. Such discussion is left to those who are better aware of such matters.

Because the modeling approach is not, at least yet, supported by a strictly guiding development system it requires quite a degree of self-control from the modeler. It is modelers' responsibility to utilize the approach and concepts defined by it, and that may cause problems because all the concepts may not be as clear as they should. That is, however, unavoidable because the modeler is always responsible in the same way as a programmer is responsible for the readability and correctness of a program. With proper tools the modeler's task might be made easier, though.

The computational problem can be avoided by hiding the deepest structure of the model from normal computations. Such hiding could be done by using the method described in [2]. If computations are needed, the causal processes and interactions needed could be created when required and disposed when the need no longer exists. The hiding of the more precise structure of a model in order to avoid computational problems seems to promise a solution to the question of the computational effectiveness of causal models, see e.g. [10].

Although the causal process modeling approach has several disadvantages, we feel that the advantages outweigh the disadvantages. We believe the most valuable advantage is the possibility of using the *same* point of view throughout the whole modeling process, and that this is an element which has been left out of most discussions about modeling or knowledge representation.

References

[1] Adams, T. L. (1986). Model-Based Reasoning for Automated Fault Diagnosis and Recovery Planning in Space Power Systems. In *Proc. 21st Intersociety Energy Conversion Engineering Conference.* San Diego.

[2] Ahonen, J. J. and Saarenmaa, H. (1991). Model-based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents. In [20]. 193-200.

[3] Ahonen, J. J. (1991). *Domain Knowledge and Explanation.* Presented in *BISFAI-91* symposium.

[4] Ahonen, J. J. (1993). Deep Knowledge and Domain Models. Report series A, Computer Science, University of Joensuu, 3. Submitted to *Informatica.*

[5] Ahonen, J. J. (1994). On Qualitative Modeling. *AI & Society*, 8. 17-28.

[6] Chandrasekaran, B. and Mittal, S. (1983). Deep Versus Compiled Knowledge Approaches to Diagnostic Problem Solving, *International Journal of Man-Machine Studies*, 22. 425-436.

[7] Clancey, W. J. (1992). Model Construction Operators. *Artificial Intelligence*, 53. 1-115.

[8] Cota, B. A. and Sargent, R. G. (1992). A Modification of the Process World View. *ACM Transactions on Modeling and Computer Simulation*, 2. 109-129.

[9] Cross, S. E. (1984). Model-based Reasoning in Expert Systems: An application to enroute air traffic control. In *Proc. AIAA/IEEE 6th Digital Avionics Systems Conference*. Baltimore.

[10] D'Ambrosio, B., Fehling, M. and Adams, T. (1985). Levels of Abstraction in Model Based Reasoning. In *Proc. ROBEXS '85 (Robotics and Expert Systems)*. NASA. Johnson Space Center.

[11] de Kleer, J. and Williams, B. C. (1987) Diagnosing Multiple Faults, *Artificial Intelligence*, 32. 97-130.

[12] Dowe, P. (1992). Wesley Salmon's Process Theory of Causality and the Conserved Quantity Theory. *Philosophy of Science*, 59. 195-216.

[13] Fetzer, J. H. (1987). Critical Notice: Wesley Salmon's Scientific Explanation and the Causal Structure of the World, *Philosophy of Science*, 54. 597-610.

[14] Fetzer, J. H. (1981). *Scientific Knowledge*. D Reidel Publishing Company. Boston.

[15] Fishwick, P. A. (1992). An Integrated Approach to System Modeling Using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. *ACM Transactions on Modeling and Computer Simulation*, 2. 307-330.

[16] Futo, I. and Gergely, T. (1990). *Artificial Intelligence in Simulation*. Ellis Horwood. Chishester.

[17] Hacking, I. (1967). Slightly More Realistic Personal Probability, *Philosophy of Science*, 34. 311-325.

[18] Hayes, P. J. (1979). The Second Naive Physics Manifesto. In Michie, D. (ed) *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press. Edinburgh.

[19] Husserl, E. (1970). *Crisis of European Sciences and Transcendal Phenomenology*. Northwestern University Press. Evanston.

[20] Hälker, M. and Jaeschke, A. (eds) (1991). *Computer Science for Environmental Protection — Proceedings of the 6th Symposium*. Springer-Verlag. Berlin.

[21] Irtzik, G. and Meyer, E. (1987). Causal modeling: New Directions for Statistical Explanation, *Philosophy of Science*, 54. 495-514.

[22] Iwasaki, Y. and Simon, H. A. (1993). Retrospective on "Causality in device Behavior". *Artificial Intelligence*, 59. 141-146.

[23] Kiviat, P. J. (1991). Simulation, Technology and the Decision Process. *ACM Transactions on Modeling and Simulation*, 1. 89-98.

[24] Koton, P. A. (1985). Empirical and Model-Based Reasoning in Expert Systems. In *IJCAI-85 Proceedings*. Los Angeles.

[25] Kourtz, P. (1987). Expert system dispatch of forest fire control resources. *AI Applications in Natural Resource Management*, 1. 1-8.

[26] Kuipers, B. (1993). Qualitative simulation then and now. *Artificial Intelligence*, 59. 133-140.

[27] Kuipers, B. (1985). The Limits of Qualitative Simulation. in: *Proceedings of International Joint Conference on Artificial Intelligence*. 128-136.

[28] Lakatos, I. (1986). *The Methodology of Scientific Research Programs*. Campbridge University Press. Cambridge.

[29] Lewis, T. G. and Smith, B. J. (1979). *Computer Principles of modeling and Simulation*. Houghton Mifflin. Boston.

[30] Miller, D. P., Firby, R. J., Fishwick, P. A., Franke, D. W. and Rothenberg, J. (1992). AI: What Simulationists Really Need To Know. *ACM Transactions on Modeling and Computer Simulation*, 2. 269-284.

[31] Minsky, M. A Framework for Representing Knowledge. In: Haugeland, J. (ed.) (1981). *Mind Design*. MIT Press. Cambridge.

[32] Nardi, B. A. and Simons, R. K. (1986). *Model-Based Reasoning and AI Problem Solving*. IntelliCorp Technical Article Reprint. IntelliCorp.

[33] Niiniluoto, I. (1983). *Scientific Reasoning and Explanation* (in Finnish). Otava. Keuruu.

[34] Oker-Blom, P., Kellomäki, S., Valtonen, E. and Väisänen, H. (1988). Structural development of Pinus sylvestris Stands with Varying Initial Density: a Simulation Model, *Scandinavian Journal of Forest Research*, 3. 185-200.

[35] Rich, S. H. and Venkatasubramanian, V. (1987). Model-Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants, *Computers and Chemical Engineering*, 11. 111-122.

[36] Saarenmaa, H. (1988). Model-Based Reasoning in Ecology and Natural Resource Managemant. In *Proc. Resource Technology 88: International Symposium on Advanced Technology in Natural Resource Management*. Fort Collins (Colorado).

[37] Saarenmaa, H., Kaila, E., Nuutinen, T. and Kolström, T. (1991). Operational forest management planning with logic programming. In: *Current Advances in the Use of Computers in Forest Research — Proc. of the IUFRO Workshop February 14, 1991*. Bulletins of the FFRI Vol. 395. 61-68.

[38] Salmon, W. C. (ed) (1971). *Statistical Explanation and Statistical Relevance*. University of Pittsburg Press. Pittsburg.

[39] Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton University Press. Princeton.

[40] Salmon, W. C. (1994). Causality Without Counterfactuals. *Philosophy of Science*, 61. 297-312.

[41] Sober, E. (1984). Common Cause Explanation, *Philosophy of Science*, 51. 212-241.

[42] Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning, *Cognitive Science* , 12. 257-285.

[43] Thompson, J. R. (1989). *Empirical Model Building*. John Wiley & Sons, Inc. New York.

[44] Weld, D. S. and de Kleer, J. (eds) (1990). *Qualitative Reasoning about Physical Systems*. Morgan Kaufmann Publishers, San Mateo, California.

[45] Widman, L. E. and Loparo, K. A. (1989). Artificial Intelligence, Simulation and Modeling: A Critical survey. In Widman, Loparo and Nielsen [46]. 1-44.

[46] Widman, L. E., Loparo, K. A. and Nielsen, N. R. (1989). *Artificial Intelligence, Simulation & Modeling*. John Wiley & Sons. New York.

[47] Xiang, Z. and Srihari, S. N. (1986). Diagnosis Using Multi-level Reasoning. In *Proc. Expert Systems in Government Symposium*, McLean (VA USA).

[48] Zeigler, B. P. (1976). *Theory of Modeling and Simulation*. Wiley. New York.

Chapter 5

Domain Knowledge and Explanation

Jarmo J. Ahonen
Department of Computer Science, University of Joensuu, P.O.Box 111, 80101 Joensuu, Finland

June 18, 1991

Abstract

Explanation has been considered to be one of the most valuable contributions of AI systems. Because the difference between domain knowledge and reasoning knowledge seems to be an acknowledged fact, it is reasonable to develop different explanation strategies for each of them. In this paper an approach to the explanation of domain knowledge is developed. The approach is based on a promising approach to the structure of knowledge and formalized using a calculus specially developed for causal explanations. The formalization is achieved by modifying an event oriented approach and the calculus according to the process oriented approach.

5.1 Introduction

Many fields of potential AI usage are concerned with physical domains. AI systems concerned with physical domains are often called expert systems. At first such systems were developed according to the production system paradigm, but recently there has been development that seems to turn expert systems into a different form. That new form has been named model based reasoning. The basic idea of model based reasoning is to keep the descriptions of the domain and the problem solving separate. Model based reasoning can be called a methodology rather than a technique, because it defines only the approach to problems, not the implementation of the system [16].

The idea of model based reasoning has received quite a large amount of interest. Several works have been related to different aspects or practical uses of model based reasoning, e.g. de Kleer and Williams [3], Koton [11], Nardi and Simons [12], Xiang and Srihari [20], Rich and Venkatasubramanian [15], Cross [2], and Adams [1]. The importance of domain models to intelligent systems seems now to be a well established opinion.

Note, however, that the remarkable importance of the models to the development of intelligent systems has earlier been advocated by Hayes, who argued, in a very convincing way, that an approach which is concerned with just reasoning is not enough [9]. He proposed the use of models of the physical reality as a method to avoid the problem of too simple and simplified problem domains. Despite the amount of work concerned with the ideas proposed by Hayes, the concept of model based reasoning did not seem to acquire very broad acceptance. The emergence of the model based reasoning theme, although implicitly proposed by Hayes, acquired its current importance due to the idea of diagnostic programs designed to find and analyze faults in physical systems.

Diagnostic systems, or any other type of intelligent systems have to have a method to reveal the internal workings of the system to possible users. Such a feature, explanation, has been considered to be one of AI's most valuable contributions [19]. Although the importance of explanation has been agreed, the theoretical structure of explanation of domains has not often been explicitly studied. In this work the structure of explanations of a problem domain is considered.

The theory of explanation according to which explanation is done has to respect the structure of domain knowledge and still provide a strict enough formalism. The formalism is required in order to enable a system to provide explanations, and the structure of domain knowledge has to be defined in a way which allows the use of best scientific knowledge of the domain and provides a connection to the formalism of adequate explanations. In order to achieve both of these goals, the probably most successful attempt to capture the structure of our scientific knowledge is used as a basis to which a promising formalism for explanations is connected.

This paper consists of two parts, the first part presents both the approach to scientific knowledge and a language oriented formalism for explanation. The second part of this paper is a proposal of a connection between these theories.

5.2 Knowledge of physical domains

Often there have not been very strict restrictions on the domains modelled in an AI system. Models have been thought to be descriptions of either physical or ideal entities [8]. It may, however, be that the structure of knowledge of physical domains is different when compared to the structure of knowledge of ideal domains. The theory of the structure of knowledge used in this work had originally been intended to outline scientific knowledge. Because most of the scientific domains are physical domains, it is reasonable to assume that a theory which is usable with scientific knowledge is usable with physical domains.

Probably the most promising approach to scientific knowledge is proposed by Salmon [17]. In order to avoid the Humean problem of causality he proposed the concept of causal processes instead of causal events (the philosophical basis of processes will not be discussed here. Although such considerations are, of course, important and worth the effort, they are not useful in order to achieve our aim). Of causal processes he says that

> 'Causal processes propagate the structure of physical world and provide the connections among the happenings in the various parts of space time. Causal interactions produce the structure and modifications of structure that we find in the patterns exhibited by the physical world. Causal laws govern the causal processes and causal interactions, providing the regularities that characterize the evolution of causal processes and the modifications that result from causal interactions.' [17, p. 132; emphasis deleted].

The concept of causal processes is not, however, very clear; many of their features have to be defined with more precision. One reason to define the features is, of course, the question of the identification of causal processes. The features of causal processes include mark transmission, structure transmission, principle of causal influence, and causal interaction.

The most important criterion of a causal process is its ability to transmit a mark. By mark transmission we mean that a causal process can transmit a mark from point *A* to point *B* (and every point between them) without further interactions. The mark transmission (MT) is defined, in a more explicit way [17, p. 148], as:

> Let *P* be a process that, in the absence of interactions with other processes, would remain uniform with respect to a characteristic *Q*, which it would manifest consistently over an interval that includes both the space-time points *A* and *B* $(A \neq B)$. Then, a mark (consisting of a modification of *Q* into Q'), which has been introduced into process *P* by means of a single interaction at point *A*, is transmitted to point *B* if *P* manifests the modification Q' at *B* and all stages of the process between *A* and *B* without additional interventions.

Note that marks in a process are, actually, changes in the process itself. Therefore we can say that a transmission of a mark is a transmission of the changed structure of the process transmitting the mark, and a process can always be said to transmit its own structure, changed by a mark or not. The *principle of structure transmission* (ST) can be formulated as follows [17, p. 154]:

> If a process is capable of transmitting changes in structure due to marking interactions, then that process can be said to transmit its own structure.

The fact that a process does not transmit a particular type of mark does not mean that it is not a causal process. Consider, as an example, processes of a hard rubber ball and a particular beam of light (caused by a lamp and colored white). It is possible to paint a green mark on the surface of the ball but it is not possible to do the same to the beam of light, although it is possible to change the color of the beam to green by using a green filter. Therefore marks must be consistent with the structure and properties of causal processes — a causal process cannot be marked by every method.

In accordance with the principle of structure transmission there must be a way to define how a causal process propagates causality from one space-time locale to another. The *principle of causal influence* (PCI) can be defined as [17, p. 155]:

> A process that transmits its own structure is capable of propagating a causal influence from one space-time locale to another.

Combined together the concepts MT, ST and PCI define what a causal process is. Although a causal process can be very effectively defined by them, no interactions between processes have been defined. Obviously processes interact — consider a billiard game with balls colliding with each other — and interactions constitutes the actual structure of causal relations.

As can be deduced from the definitions of MT, ST and PCI, there exist, in addition to causal processes, a great deal of non-causal processes (or pseudo processes). The existence of non-causal processes makes the definition of causal interaction (CI) between processes quite difficult a task, and one proposition is made by Salmon [17, p. 171] and is as follows:

> Let P_1 and P_2 be two processes that intersect with one another at the space-time point S, which belongs to the histories of both. Let Q be a characteristic that process P_1 would exhibit throughout an interval (which includes subintervals on both sides of S in the history of P_1) if the intersection with P_2 did not occur; let R be a characteristic that process P_2 would exhibit throughout an interval (which includes subintervals on both sides of S in the history of P_2) if the intersection with P_1 did not occur. Then the intersection of P_1 and P_2 at S constitutes a causal interaction if:
>
> 1. P_1 exhibits the characteristic Q before S, but it exhibits a modified characteristic Q' throughout an interval immediately following S; and
> 2. P_2 exhibits the characteristic R before S, but it exhibits a modified characteristic R' throughout an interval immediately following S.

The features of causal processes and causal interactions are not easy to formalize [4]. Although a physical domain may be modelled according to the concept of causal processes, a strict formalism is difficult to develop due to the counterfactuals and the non-logical modalities used in the definition of causal processes. Such a formalism is not presented, and the problem of explanation is approached using a formal calculus and an event oriented approach to explanation. The calculus and the event oriented approach are modified and used later on to express explanations according to the concept of causal processes. First the calculus and the event oriented approach are introduced.

5.3 Formal explanations

A formalism for explanations is required in order to enable expert systems to provide them. In order to avoid the problem of counterfactuals and non-logical modalities a language oriented calculus is used, and an event oriented approach to explanation is presented. The philosophical foundations of the calculus and the event oriented approach to explanation are not discussed in this paper, for them we refer to Fetzer and Nute [6] [7] and Fetzer [5].

5.3.1 A framework for explanations

Considering explanations from a language oriented point of view, it is possible to assume that individual language frameworks exist. Such a *language framework*, $\mathcal{L}zt$, consists of all sentences used or potentially used by a person z at time t. The language framework $\mathcal{L}zt$ includes semantical and syntactical features of the language used by z.

Similarly it is possible to assume a knowledge context, which may be defined as follows:

> Let $\mathcal{K}zt$ denote a collection of the sentences representing the knowledge possessed by a person z at time t. The set $\mathcal{K}zt$ is called the *knowledge context* of the person z at time t. The sentences of $\mathcal{K}zt$ are sentences of $\mathcal{L}zt$, i.e. $\mathcal{K}zt \subseteq \mathcal{L}zt$.

For the sentences constituting $\mathcal{K}zt$ we may require different conditions to be satisfied before accepting a set of sentences stating that a particular predicate is the case to $\mathcal{K}zt$. Such conditions may include consistency (at least in the context of the domain), the requirement of evidence for the proposition, the method of acceptance, etc. It seems to be the case that the reasons for accepting sentences into $\mathcal{K}zt$ are always relative to the individual and the time of the acceptance.

For later purposes it is useful to assume that there exists a kind of common language framework, $\mathcal{L}*$, which allows people to understand each other (and information provided by computer programs). The *common language framework* may be defined as follows:

> Let $z_1, \ldots, z_n$ be any people, and $\mathcal{L}*$ a set of sentences for which it is true that:
>
> 1. the language framework $\mathcal{L}z_it$ of the person z_i includes $\mathcal{L}*$, $1 \leq i \leq n$, and
> 2. the interpretation of the sentences of $\mathcal{L}*$ is sufficiently identical in all $\mathcal{L}z_it$.

The definitions of language frameworks[1] and the knowledge context are not very accurate and cannot be defended by explicit proofs. They are, however, useful, as will be seen later on.

5.3.2 A calculus for causal explanations

The distinction between the physical causality and the logical implication suggests that the physical causality requires a new operator. Such a new operator should be able to capture every possible type of causality and be able to coexist with normal logic. The coexistence is required because some uses of such calculus may require both the logical implication and the physical causality. To achieve this purpose the causal calculus **C** developed by Fetzer and Nute [6][7] has been chosen. In this section some features of the calculus are briefly introduced and the intended interpretations of its new operators examined. A brief introduction to the calculus **C** is presented in the appendix for the convenience of readers.

The calculus **C** includes the "fork" operator '$\ni\!\!-$' and several axioms for it. Although some axioms (A2 and A3 in the appendix) seem to suggest that '$\ni\!\!-$' is the same as the logical implication '$\supset$', it is not the suggested interpretation. Actually the operator '$\ni\!\!-$' denotes property projection or property transfer. The sentence '$p \ni\!\!- q$' is generally interpreted as 'if p were the case, then q would be the case', which could be interpreted in a practical situation as a less formal sentence 'if this item is made of wood, then it can burn, i.e. it has the burning property of wood'. The practical interpretation of the '$\ni\!\!-$' operator will be discussed in the next section.

Another interesting new operator is the universal conditional '$\ni\!\!-_u$'. The sentence '$p \ni\!\!-_u q$' is interpreted as 'if p were the case, it would always bring, i.e. cause, q to be the case'. The last one of the operators is '$\ni\!\!-_n$' which is interpreted as 'if p were the case, it would always bring, i.e. cause, q to be the case with the probability of n'. Formally that probabilistic conditional can be defined as follows:

Let S be a non-decreasing sequence of finite sets of logically possible worlds relative to the actual world **W**, i.e. if $i \le j$ then $S_i \subseteq S_j$. Then the sentence '$p \ni\!\!-_n q$' is true of **W** if and only if 'almost every sequence' S of possible sets of possible worlds satisfies the following condition (I):

$$\lim_{k\to\infty} \frac{\text{the number of worlds in } S_k \text{ in which } q \text{ is true}}{\text{the number of worlds in } S_k} = n.$$

Likewise the notion of 'almost every sequence' can be made more precise by: the sentence '$p \ni\!\!-_n q$' is true of **W** only in the case when the following condition (I*) is fulfilled:

$$\lim_{k\to\infty} \frac{\text{the number of members of } (S^m | m \le k \text{ and } S^m \text{ satisfies (I)})}{\text{the number of members of } k} = n$$

[1] The common language framework is essential in order to allow a system to produce explanations understandable by many people. Although the definition of $\mathcal{L}*$ is philosophically questionable, we will proceed according to the assumption that it is reasonable.

Note that according to the definition of **C**, it is obvious that '$p \mathrel{\ni\!\!\!-}_u q$' and '$p \mathrel{\ni\!\!\!-}_1 q$' are not the same thing. Also it is true that '$p \mathrel{\ni\!\!\!-}_0 q$' and '$p \mathrel{\ni\!\!\!-}_u \neg q$' are not the same. Although the frequency of an event suggests probability 1, it is not certain that the event holds universally[10, p. 5]. Therefore '$p \mathrel{\ni\!\!\!-}_1 q$' does not guarantee that q comes after p.

5.3.3 Explanative inferences

Explanations may be considered to be inferences. For such inferences there have to be constraints and other definitions by which the adequacy of explanations may be evaluated. In this section an event oriented approach to the definition of adequate explanation is presented.

A principal feature of explanations is the concept of lawlike sentences. The following definition of lawlike sentences is adapted from Fetzer [5, p. 44]. A sentence S is *lawlike* within $\mathcal{L}*$ if and only if

(i) S is *completely general*, i.e. S is not limited to any finite number of instances on the basis of syntactical or semantical features of $\mathcal{L}*$; and

(ii) S is *essentially dispositional*, i.e. S attributes a permanent dispositional property $\mathcal{X}$ to every member of a reference class K (under an appropriate description 'K') in $\mathcal{L}*$.

Note that accidental considerations are not lawlike. If we say that 'All cars here are blue', we do not say a lawlike sentence because a red car can arrive anytime. That type of non-lawlike sentences is discussed by many writers (e.g. Popper [14, p. 427-428]). Because all sentences are not lawlike the concept of lawlike sentences requires further consideration.

We can comfortably assert that the *basic form of lawlike sentences* is the one reading ⌈For all x and all t, if x were K at time t, then x would be $\mathcal{X}$ at t⌉. Hereafter we will use the symbols '⌈' and '⌉' to show intended interpretations. That type of sentence can be expressed formally by using a subjunctive conditional "fork" operator as follows:

$$(\forall x)(\forall t)(Kxt \mathrel{\ni\!\!\!-} \mathcal{X}xt) \tag{5.1}$$

where the fork makes the sentence mean the same as ⌈For all x and all t, if x were K at time t, then x would be $\mathcal{X}$ at t⌉.

Considering lawlike sentences within $\mathcal{L}*$, we have to note that most of the sentences claim that the occurrence of an event of kind T^i causes the occurrence of an event of kind O^i. Some of the dispositions are of statistical strength and some are of universal strength, which causes the causal conditional to be probabilistic. A probabilistic causal conditional is applicable with degrees of strength n whose values may range through varying statistical strengths n from zero to one, i.e. '$\mathrel{\ni\!\!\!-}_n$' to universal strength, i.e. '$\mathrel{\ni\!\!\!-}_u$' (not to be confused with probabilities of one, for reasons to be considered later on), where the appropriate numerical value is determined by the strength of the disposition that is involved [5, p. 48].

The *causal form* of lawlike sentences is, therefore, of the form

$$(\forall x)(\forall t)(Kxt \mathrel{\ni\!\!\!-} (T^i xt \mathrel{\ni\!\!\!-}_n O^i xt^*)) \tag{5.2}$$

which asserts that ⌈For all x and all t, if x were K at time t, then the strength of the dispositional tendency for T^i-ing x at t to bring about O^i-ing x at t^* is n⌉, or, less formally ⌈For all x and all t, if x were K at time t, then T^i-ing x at t would either invariably or probably bring about x's O^i-ing at t^*⌉ [5, p. 48].

Note that the explanatory force of subjunctive conditions and causal conditionals depends on the precise character of the phenomenon or the object to be explained. Fetzer differentiates lawlike sentences and their instantiations calling them lawlike sentences and nomological conditionals, respectively [5, p. 49], and defines scientific conditionals as follows:

> Scientific conditionals can be specified as
>
> (A) lawlike sentences
>
> (a) simple forms
> $(\forall x)(\forall t)(Kxt \ni\!\!- Xxt)$;
>
> (b) causal forms
> $(\forall x)(\forall t)(Kxt \ni\!\!- (T^ixt \ni\!\!-_n O^ixt^*))$;
> $(\forall x)(\forall t)((Kxt \wedge T^ixt) \ni\!\!-_n O^ixt^*)$;
> $(\forall x)(\forall t)(T^ixt \ni\!\!- (Kxt \ni\!\!-_n O^ixt^*))$;
>
> (B) nomological conditionals:
>
> (a) simple forms
> $Kbt \ni\!\!- Xbt$,
>
> (b) causal forms
> $(Kbt \ni\!\!- (T^ibt \ni\!\!-_n O^ibt^*))$;
> $((Kbt \wedge T^ibt) \ni\!\!-_n O^ibt^*)$;
> $(T^ibt \ni\!\!- (Kbt \ni\!\!-_n O^ibt^*))$.

It appears to be that a sentence of kind (B) logically entails a corresponding sentence of kind (A). In order to achieve a kind of consistency amongst our sentences, restrictions have to be developed. One possibility to achieve the required consistency is to use the requirement of maximal specificity, which is proposed by Fetzer [5, p. 50], and define it as follows:

> The *requirement of maximal specificity*: If a nomically relevant predicate is added to the reference class description of a scientific conditional S which is true in $\mathcal{L}*$, then the resulting set of sentences S^* is such that either
>
> (i) S^* is no longer true in $\mathcal{L}*$ (because its antecedent is now self-contradictory), or
>
> (ii) S^* is logically equivalent to S in $\mathcal{L}*$ (because the predicate was already entailed by the antecedent of S).

Preferring causal explanations we have to adjust the requirement of maximal specificity to causal explanations. Fetzer [5, p. 125-126] defines the requirement of strict maximal specificity in order to be able to define an adequate concept of explanation incorporating a rule for selecting the appropriate lawlike sentence to include in the explanans for the explanation of all such explananda. The language framework in the following definitions is the language framework of the one (man or machine) who is doing the explaining, and the definition of the requirement of strict maximal specificity is as follows:

> The *Requirement of Strict Maximal Specificity* (RSMS): An explanation of why an explanandum event — the possession of a property $\mathcal{X}$ by an individual c or an outcome response O^i - occurs is adequate *only if* every property described by the antecedent condition(s) of any lawlike sentence S in the explanans of that explanation is nomically relevant to the occurrence of its attribute property $\mathcal{X}$ or outcome response O^i, within $\mathcal{L}zt$.

On the basis of the considerations above a criterion for single case explanations can be advanced. That kind of criterion is for singular events, and does not require any considerations of the law behind the explanation — the law can be a universal or a statistical nomological generalization. This criterion is proposed by Fetzer [5, p. 126-127]:

> A set of sentences S, known as the "explanans" provides an *adequate nomically significant causal explanation* of the occurrence of a singular event described by another sentence E, known as its "explanandum", relative to the language framework $\mathcal{L}zt$, if and only if:
>
> (a) the explanandum is either a *deductive* or a *probabilistic consequence* of its explanans;
>
> (b) the explanans contains at least one lawlike sentence of (universal or statistical) 'causal' form that is actually required for the deductive or probabilistic derivation of the explanandum from its explanans;
>
> (c) the explanans satisfies the requirement of strict maximal specificity with respect to its lawlike premise(s); and,
>
> (d) the sentences constituting the explanation — both the explanans and its explanandum — are true, relative to the language framework $\mathcal{L}zt$.

Obviously an explanation must include all and only causal (or nomical) predicates that are relevant to the occurrence of the explanandum phenomenon. In an AI system this would mean that all explanations are minimized.

Universal lawlike sentences used in explanation make the explanation a creation of deductive powers. As an example we can consider the deductive universal explanation given by Fetzer [5, p. 127-128]. The phenomenon explained is obvious on the basis of the following formalisms:

(CL) For all x and for all t, if x were gold at t, then heating x to 1063°C at t would invariably bring about its melting at t^*;

(C1) Jan's bracelet b is made of gold at t;

(C2) Jan’s bracelet *b* is heated to 1063°Cat *t*;

(ES) Jan’s bracelet *b* melts at *t**.

The flow of deduction can be more formally presented as:

$$\begin{array}{c}(\forall x)(\forall t)(Gxt \mathrel{\ni\!\!-} (Hxt = 1063^{\circ}\mathrm{C} \mathrel{\ni\!\!-_{u}} Mxt^{*}))\\ Gbt \wedge Hbt = 1063^{\circ}\mathrm{C}\\ \hline Mbt^{*}\end{array}[u] \qquad (5.3)$$

In the schematization (5.3) the bracketed *u* means that the deduced consequence is of universal strength, or, in the other words, that the consequence Mbt^* is the only possible. The explanation above is a deductive one, there are no probabilistic or statistical considerations in it.

Being interested in explanations in AI systems we must note that deductive explanations are not problematic — their structure is so near to the structure of logical deduction that their use and creation is obvious. Probabilistic explanations are not so obvious.

As an example of a probabilistic explanation for another explanandum event we will use the example given by Fetzer [5, p. 128-129].

(CL) For all *x* and all *t*, if *x* were polonium^{218} at *t*, then a time trial of three minutes duration at *t* would bring about the loss of nearly half the mass of *x* by $t + 3$ minutes with strength 0.9.

(C1) Smith brought a sample *b* of eight grams of polonium^{218} into the lab for analysis at *t*;

(C2) Smith weighted the sample *b* three minutes after bringing it in;

(ES) The mass of *b* at $t + 3$ minutes was approximately four grams total.

The flow of the reasoning can be schematized as follows:

$$\begin{array}{c}(\forall x)(\forall t)(Mxt \mathrel{\ni\!\!-} ((Txt = 3\mathrm{min}) \mathrel{\ni\!\!-_{0.9}} (Mxt^{*} = \tfrac{1}{2}Mxt)))\\ (Mbt = 8\mathrm{g}) \wedge (Tbt = 3\mathrm{min})\\ \hline\hline Mbt^{*} = 4\mathrm{g}\end{array}[0.9] \qquad (5.4)$$

The double line in the schematization separates the explanans from the explanandum and indicates that the logical property between them is partial. The number in brackets indicates the degree of nomic expectability.

The use of probabilities in explanation can cause some problems. It is not, however, reasonable to consider those problems and paradoxes here, we just refer to [5, p. 128-136].

Since the explanation facility of any expert system needs to consider and be able to explain situations which are not actually causal, it is reasonable to consider non-causal explanations. It is a universal fact that there is a temperature where paper burns. Now we can know that a book is made of paper. From the universal fact and from the material used in the book we can deduce that the book has the property that it burns at the temperature where paper burns. This can be written as:

(CL) For all x and all t, if x were paper at t, then x would burn in the temperature *temp*.

(C1) This book is made of paper

(ES) This book burns in the temperature *temp*;

which can be schematized as follows:

$$\begin{array}{c} (\forall x)(\forall t)(Pxt \mathrel{\ni\!\!-} (Bxt = \mathit{temp})) \\ Pbt \\ \hline Bbt = \mathit{temp} \end{array}. \tag{5.5}$$

The reasoning in (5.5) is obviously an explanation for the fact that this book burns. Note that the explanation is not a causal one. The simple explanation (5.5) is an example of non-causal explanations which explain why particular things have specific properties at certain times by invoking simple lawlike sentences.

The concept of non-causal explanation requires a more accurate definition. Since the concept of adequate causal explanation is specified as Fetzer does, another specification proposed by Fetzer [5, p. 141-142] will do, namely: A set of sentences S, known as the “explanans” provides an *adequate nomically significant non-causal explanation* for the occurrence of a singular event described by another sentence E, known as its “explanandum”, relative to the language framework $\mathcal{L}zt$, if and only if:

(a) the explanandum is either a deductive consequence of its explanans;

(b) the explanans contains at least one lawlike sentence of the simple form that is actually required for the derivation of the explanandum from its explanans;

(c) the explanans satisfies the requirement of strict maximal specificity with respect to its lawlike premise(s); and

(d) the sentences constituting the explanans are true, relative to the language framework $\mathcal{L}zt$.

The example (5.5) and the definition of adequate nomically significant non-causal explanations tells what kind of sentences can be used in non-causal explanations. Fetzer [5, p. 140-161] considers the adequacy of non-causal explanations and achieves a conclusion according to which non-causal explanations are adequate. As a general definition of adequate explanations he gives one which connects the concept of non-causal and causal explanations.

A set of sentences S, known as the “explanans” provides an *adequate nomically significant explanation* for the phenomenon — whether singular or general — described by another sentence E, known as its “explanandum”, relative to the language framework $\mathcal{L}zt$, if and only if:

(a) the explanandum is either a deductive or a probabilistic consequence of its explanans;

(b) the explanans contains at least one lawlike sentence (of either causal or simple form) that is actually required for the deductive or probabilistic derivation of the explanandum from its explanans;

(c) the explanans satisfies the requirement of strict maximal specificity with respect to its lawlike premise(s); and

(d) the sentences constituting the explanation — both the explanans and its explanandum — are true, relative to the language framework $\mathcal{L}zt$.

The definition of an adequate nomically significant explanation does not satisfy the requirement of respect to the structure of domain knowledge. The seemingly versatile nature of the calculus **C** does, however, suggest that by proper modifications the calculus can be used in explanations which do respect the structure of domain knowledge. In the following section such modifications are presented.

5.4 Formal explanation and the concept of causal processes

In the above sections adequate explanations and the form of causal explanations have been considered. The formalism is explicit and usable for explanative inferences, but it does not respect the structure of domain knowledge. Although the approach to domain knowledge seems to include a large amount of counterfactuals, extensional conditionals, and non-logical modalities, it is possible to use a formal mechanism of generating adequate explanations which respect earlier considerations of domain knowledge.

5.4.1 Formal explanations of causal processes

Fortunately we do not have to reformulate the concept of causal processes in order to be able to provide explanations which respect that concept. It seems to be sufficient to create a formally usable method to generate explanations of causal processes, and leave the processes themselves to their current state. As the method to deal with causal processes, the promisingly versatile calculus C seems to require only some modifications to be used for such a purpose. The modifications mostly seem to concentrate on the interpretation of the calculus, and not so much on the calculus itself, as will be seen later on.

At first it is useful to reconsider the previous example of heating a piece of gold to the temperature of 1063°C. We propose that the schematization (3) should be written in a form which could be interpreted according to the concept of causal processes. One possibility would be:

$$\frac{\begin{array}{c}(\forall x)(\forall t)(Gxt\text{-} \ni\!\!- (Hxt = 1063^{\circ}\mathrm{C} \ni\!\!-_{u} Mxt+))\\ Gbt\text{-} \wedge Hbt = 1063^{\circ}\mathrm{C}\end{array}}{Mbt+}[u]. \tag{5.6}$$

When reading the schematization (5.6) we should say that ⌈For all causal processes x and all t, if x were characterized by being gold at time interval t- before a causal interaction at t, then the strength of the dispositional tendency for a causal interaction, which causes x to have a mark of being heated to the temperature 1063°C to be imposed on it, at t to bring about the changed characteristic, melted form, for the causal process x at time interval t+ after the causal interaction at t is universal⌉. The example shows that we can interpret explanations expressed in the form of causal calculus as explanations using the concept of causal processes as the basis.

According to the spirit of the example above, the basic form of lawlike sentences is the one reading ⌈For all causal processes x and all t, if x were characterized by K at time t, then x would be characterized by X at t⌉. That type of sentence can be expressed formally as follows:

$$(\forall x)(\forall t)(Kxt \mathrel{\ni\!\!-} Xxt). \qquad (5.7)$$

Note that although the sentence (5.7) looks exactly like (5.1), it reads as ⌈For all causal processes x and all t, if x were characterized by K at time t, then x would be characterized by X at t⌉.

Although (5.7) looks like (5.1), its new interpretation is different, as will be seen later on. It is possible that the basic form of lawlike sentences would be cognitively more tempting if it were not read according to the concept of causal processes — but it must, anyway, be noted that the causal process interpretation of the basic form of lawlike sentences respects the structure of domain knowledge better. The causal process interpretation outlaws all uses of pseudo-processes and defines what kind of things the sentence is about.

Similarly the causal form of the lawlike sentences can be formulated according to the example above, as:

$$(\forall x)(\forall t)(Kxt\text{-} \mathrel{\ni\!\!-} (T^i xt \mathrel{\ni\!\!-}_n O^i xt+)). \qquad (5.8)$$

The sentence (5.8) asserts that ⌈For all causal processes x and all t, if x were characterized by K at time interval t- before a causal interaction at t, then the strength of the dispositional tendency for a causal interaction, which causes x to have a mark T^i to be imposed on it, at t to bring about the changed characteristic O^i for the causal process x at time interval t+ is n⌉.

Interpretations of (5.7) and (5.8) imply that restrictions and definitions created for event oriented explanative sentences can be used in a modified form when speaking of causal processes and their interactions. Although sentences (5.7) and (5.8) are useful for explanations which respect the concept of causal processes, some other features of explanation do not hold any longer.

The probabilistic explanation provides some problems. Considering the example of polonium^{218} (and its schematization (5.4)) it must be noted that the provided explanation is senseless when considered according to the causal process approach. Obviously the tendency to lose mass is characteristic to the causal process of polonium^{218}, in which case there has not been a causal interaction. The original lawlike sentence

> For all x and all t, if x were polonium^{218} at t, then a time trial of three minutes duration at t would bring about the loss of nearly half the mass of x by $t + 3$ minutes with strength 0.9.

has to be written in another form. Because the tendency to lose mass is characteristic to polonium^{218}, the new form is the one saying that:

> For all causal processes x and all t, if x were characterized by being polonium^{218} at time t, then x would be characterized by having the tendency to lose nearly half the mass of x by $t + 3$ minutes with strength 0.9.

The new form is essentially the same as (5.7). The schematization of an explanation of loss of mass of a sample of polonium^{218} would use the characteristics of polonium^{218} and look like (5.5) with notations changed to proper forms.

The fact that characteristic features like the one above are not expressed as causal sentences any longer does not mean that probabilistic features could be thrown away. Consider, as an example, a situation in which a coin is standing on its edge, not laying on its side. If something hits the coin at the top, there is a probability of 0.5 to have heads on the upper side of the coin after the hit. That may be written as:

(CL) For all causal processes x and all t, if x were characterized by being a coin and standing on its edge at time interval t- before a causal interaction at t, then the strength of the dispositional tendency for a causal interaction, which causes x to have a mark, in this case of being subjected to kinetic force making it change its stability, to be imposed on it, at t to bring about the changed characteristic, in this case to end up laying on its either side, for the causal process x at time interval t+ is 0.5;

(C1) The coin c has been standing on its edge during a time interval t-;

(C2) The coin is hit on its top, i.e. subjected to kinetic energy, at time t;

(ES) The coin that has been standing on its side is lying with its head side up during the time interval t+ with the probability of 0.5.

The flow of reasoning in the example has essentially the schematization (5.4) with notations changed to proper forms.

The reasoning above is not very accurate in the physical sense, but the intended interpretation should be clear. Considering the reasoning it has to be noted that it can be made more and more accurate, finally using very precise reasoning with elementary particles. Such a possibility suggests that explanations and causal processes form hierarchies. That possibility is not, however, in the scope of this work.

Our previous scientific conditionals may now be written into forms which take causal processes into account. Scientific conditional can now be specified as

(A) lawlike sentences

- (a) simple forms
  $(\forall x)(\forall t)(Kxt \ni\!\!- \mathcal{X}xt)$;
- (b) causal forms
  $(\forall x)(\forall t)(Kxt\text{-} \ni\!\!- (T^ixt \ni\!\!-_n O^ixt+))$;
  $(\forall x)(\forall t)((Kxt\text{-} \wedge T^ixt) \ni\!\!-_n O^ixt+)$;
  $(\forall x)(\forall t)(T^ixt \ni\!\!- (Kxt\text{-} \ni\!\!-_n O^ixt+))$;

(B) nomological conditionals:

- (a) simple forms
  $Kbt \ni\!\!- \mathcal{X}bt$,
- (b) causal forms
  $(Kbt\text{-} \ni\!\!- (T^ibt \ni\!\!-_n O^ibt+))$;
  $((Kbt\text{-} \wedge T^ibt) \ni\!\!-_n O^ibt+)$;
  $(T^ibt \ni\!\!- (Kbt\text{-} \ni\!\!-_n O^ibt+))$.

It appears to be that a sentence of kind (B) logically entails a corresponding sentence of kind (A), as in the original definition of scientific conditionals.

The original requirement of strict maximal specificity is not very useful under new interpretations. Therefore it could be written in the form of *modified requirement of strict maximal specificity* (MRSMS), as follows:

> An explanation of why an explanandum characteristic — the possession of a characteristic $\mathcal{X}$ by an individual causal process c or an outcome characteristic O^i of a causal process after a causal interaction — occurs is adequate only if every characteristic and causal interaction described by the antecedent condition(s) of any lawlike sentence S in the explanans of that explanation is causally or characteristically relevant to the occurrence of its attribute characteristic $\mathcal{X}$ or outcome characteristic O^i, within $\mathcal{L}zt$.

Due to the changes, the definition of an adequate nomically significant explanation is no longer more useful than the original requirement of strict maximal specificity. According to the new interpretations the definition may be written as:

> A set of sentences S, known as the "explanans" provides an *adequate characteristically or causally significant explanation* (ACCSE) for the existence of a characteristic of a causal process (a characteristic describing one process or common to many processes, i.e. singular or general) described by another sentence E, known as its "explanandum", relative to the language framework $\mathcal{L}zt$, if and only if:
>
> (a) the explanandum is either a deductive or a probabilistic consequence of its explanans;
>
> (b) the explanans contains at least one lawlike sentence (of either causal or simple form) that is actually required for the deductive or probabilistic derivation of the explanandum from its explanans;

(c) the explanans satisfies the modified requirement of strict maximal specificity (MRSMS) with respect to its lawlike premise(s); and,

(d) the sentences constituting the explanation - both the explanans and its explanandum - are true, relative to the language framework $\mathcal{L}zt$.

5.4.2 The role of modifications in C

Considering the original causal calculus and its parts, we have to note that the calculus **S** does not require any modifications. That is easy to note from the fact that the formulations (5.1) and (5.7) are essentially the same. Hence the calculus **S** can be used in its original form, only its interpretation changes into the form of (5.7).

The cases of **U** and **C** require closer examination. The sentences of causal processes are more or less the same as (5.8). The problem with that form is, essentially, that Kxt-, T^ixt, and O^ixt+ include a variety of references to time. The distinction between the use of references to time in **S** versus sentences of the type (5.8) is that simple forms of lawlike sentences refer to one time-point each, and sentences of the type (5.8) refer to two different time intervals and a time-point between them. If the use of sentences of the type (5.8) does not obligate us to make syntactic modifications to the causal calculus **C**, we should be able to use the calculus **C** in its current form.

Fortunately the problem is not a severe one. The original definition of causal calculus **C** did not define the types of predicates used in sentences. Hence it is the case that the original causal calculus can be safely used for sentences of causal processes, only if the interpretation of the calculus is changed.

5.5 Discussion

Although ACCSE provides a theoretical foundation for explanations of domain models, it leaves several interesting questions open. The questions include probabilistic explanations and possible paradoxes of probability, and the relativization to $\mathcal{L}*$, and the relation of formal basis to explanations given by actual systems, and the implementation of domain models that respect the concept of causal processes.

Problems with probabilities originate from the interpretation of probability. The frequency interpretation of probability leaves open a possibility that ‘q_b’ happens to be the case and the available explanations use either ‘$p \ni\!\!-_1 q_a$’ or ‘$p \ni\!\!-_0 q_b$’. Although we can refer to the fact asserted by Kolmogorov [10, p. 5], an explanation (of the case ‘q_b’) which uses ‘$p \ni\!\!-_0 q_b$’ is not tempting. The actual use of probabilities in explanations requires further study.

Knowledge has to be considered with some respect to the relativity of language. In that respect our considerations of adequate explanations are much more closely connected with the works of Tuomela [18] and Fetzer [5] than Salmon [17]. But it may not be reasonable to try to get rid of the relativity. Given the linguistic character of arguments and explanations there seems to be no way to avoid the relativization to a language. Unfortunately we still lack explicit proofs and definitions of $\mathcal{L}*$ and other required linguistic features, and we have to depend on more or less defended assumptions, not on proven facts.

The use of user models may be necessary in order to enable efficient use of ACCSE. Although ACCSE provides a formalism according to which explanations should be generated, it does not provide solutions to problems that concern the interface between users and intelligent systems. It may be the case that the definition of ACCSE would be best used in the internal activities of the system, and that explanations provided to a user could vary greatly on the basis of available user models.

The definition of adequate explanations that respect the structure of domain knowledge may suffer from incomplete formalization, which follows from the fact that the concept of causal processes is not formalized. Fortunately causal processes seem to be able to be modelled. The modelling of causal processes does not require the use of logic formalisms in order to be possible, and the structure and implementation of models of causal processes may vary depending on the case. The use of the concept of causal processes is left to the knowledge engineer(s) and the domain expert(s) who develop the domain model.

Despite the outlined weaknesses of our definition of adequate explanations it seems to be reasonable to say that the definition does provide a mechanizable theoretical basis on which the explanation abilities of intelligent systems may be based. Note, however, that the basis is for explanations of the physical domain only. Explanations of reasoning may require a different formalism.

References

[1] Adams, T. L.: Model-Based Reasoning for Automated Fault Diagnosis and Recovery Planning in Space Power Systems, in: *Proceedings of the 21st Intersociety Energy Conversion Engineering Conference*, San Diego, 1986, pp. 1763-1767.

[2] Cross, S. E.: Model-based Reasoning in Expert Systems: An application to enroute air traffic control, in: *Proceedings of the AIAA/IEEE 6th Digital Avionics Systems Conference*, Baltimore, 1984, pp. 95-101.

[3] de Kleer, J. & Williams, B. C.: Diagnosing Multiple Faults, *Artificial Intelligence* 32 (1) (1987), pp. 97-130.

[4] Fetzer, J. H.: Critical Notice: Wesley Salmon's Scientific Explanation and the Causal Structure of the World, *Philosophy of Science* 54 (1987), pp. 597-610.

[5] Fetzer, J. H.: *Scientific Knowledge*, D Reidel Publishing Company, Boston, 1981.

[6] Fetzer, J. H. & Nute, D. E.: Syntax, Semantics, and Ontology: A Probabilistic Causal Calculus, *Synthese* 40 (1979), pp. 453-495.

[7] Fetzer, J. H. & Nute, D. E.: A Probabilistic Causal Calculus: Conflicting Conceptions, *Synthese* 44 (1980), pp. 241-246.

[8] Futo, I. & Gergely, T.: *Artificial Intelligence in Simulation*, Ellis Horwood, Chishester, 1990.

[9] Hayes, P. J.: The Naive Physics Manifesto, in Michie, D. (Ed.): *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press, Edinburgh, 1979, pp. 242-270.

[10] Kolmogorov, A. N.: *Foundations of the Theory of Probability*, Chelsea, New York, 1956.

[11] Koton, P. A.: Empirical and Model-Based Reasoning in Expert Systems, in: *IJCAI-85 Proceedings* (1985), Los Angeles, pp. 297-299.

[12] Nardi, B. A. & Simons, R. K.: Model-Based Reasoning and AI Problem Solving, *IntelliCorp Technical Article Reprint*, IntelliCorp, 1986.

[13] Nute, D. E.: Counterfactuals, *Notre Dame Journal of Formal Logic* 16 (1975), pp. 475-482.

[14] Popper, K.: *The Logic of Scientific Discovery*, Harper & Row, New York, 1965.

[15] Rich, S. H. & Venkatasubramanian, V.: Model-Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants, *Computers and Chemical Engineering* 11 (2) (1987), pp. 111-122.

[16] Saarenmaa, H.: Model-Based Reasoning in Ecology and Natural Resource Managemant, in: *Proceedings Resource Technology 88: International Symposium on Advanced Technology in Natural Resource Management*, Fort Collins (Colorado), June 1988.

[17] Salmon, W. C.: *Scientific Explanation and the Causal Structure of the World*, Princeton Uni- versity Press, Princeton, 1984.

[18] Tuomela, R.: Inductive Explanation, *Synthese* 48 (1981), pp. 257-294.

[19] Wick, M. R. & Slagle, J. R.: An Explanation Facility for Today's Expert Systems, *IEEE Expert* 4 (1989), pp. 26-36.

[20] Xiang, Z. & Srihari, S. N.: Diagnosis Using Multi-level Reasoning, in: *The Proceedings of Expert Systems in Government Symposium*, McLean (VA USA), 1986, pp. 151-158.

APPENDIX: A brief overview of the causal calculus C

Let us assume the familiar set of primitive symbols and formation rules for constructing the calculus, namely:

(i) countably many sentence letters A, A', $\ldots$, B, B', $\ldots$;

(ii) the logical signs $\neg$ and $\supset$, which will be interpreted as usual; and

(iii) parenthesis, brackets and braces for punctuation,

i.e. the primitive signs of a classical sentential calculus. To the calculus we add a new operator '$\ni\!\!-$' to those above and define the well formed formulae x as: an object x is a formula of calculus **S** iff x belongs to every set which contains every sentence letter and also contains '$\neg p$', '$p \supset q$' and '$p \ni\!\!- q$' whenever it contains 'p' and 'q'. Let '$\wedge$', '$\vee$' and '$\equiv$' have their standard contextual definitions, and let us define the modal operators '$\Box p$' and '$\Diamond p$' contextually as '$\neg p \ni\!\!- q$' and '$\neg(p \ni\!\!- \neg q)$', respectively. The rules of inference for calculus **S** are modus ponens (MP) and the Gödel rule of necessitation (NEC), i.e. NEC: if 'p' is a theorem, then '$\Box p$' is a theorem.

A formula of **S** is an axiom of **S** iff it is tautologous or has one of the following forms:

A1. $\Box(p \supset q) \supset (\Box p \supset \Box q)$;

A2. $\Box(p \supset q) \supset (p \ni\!\!- q)$;

A3. $(p \ni\!\!- q) \supset (p \supset q)$;

A4. $(p \ni\!\!- (q \supset r)) \supset ((p \ni\!\!- q) \supset (p \ni\!\!- r))$;

A5. $(\Box(p \supset q) \wedge (q \ni\!\!- r)) \supset (\Diamond(p \wedge q) \supset (p \ni\!\!- q))$;

A6. $((p \ni\!\!- r) \wedge (q \ni\!\!- r)) \supset ((p \wedge q) \ni\!\!- r)$.

The calculus **S** is actually an extension of a calculus developed by Nute [13]. For any well formed formula 'p' and the set Γ of wffs, '$\vdash_S p$', '$\Gamma \vdash_S p$', 'Γ is **S**-consistent', and 'Γ is maximally **S**-consistent' are defined in the usual way. For our reasons it is sufficient to present the following theorems, for proofs we refer to [6]. We will not, actually, present the theorems required for the exact definition of **S** or the following **U** and **C**, we will only mention the results achieved by those theorems and refer to the original sources.

According to [6], it is possible to show that the following are theorem schemata for S:

a. $\vdash_S p \ni\!\!- q$;

b. $\vdash_S \Box p \supset (q \ni\!\!- p)$;

c. $\vdash_S \Box\neg p \supset (p \ni\!\!- q)$;

d. $\vdash_S \Diamond p \supset ((p \ni\!\!- q) \supset \neg(p \ni\!\!- \neg q))$;

e. $\vdash_S ((p \ni\!\!- q) \wedge (q \ni\!\!- p)) \supset ((p \ni\!\!- r \equiv (q \ni\!\!- r))$;

f. $\vdash_{\mathbf{S}} ((p \wedge q) \ni\!- r) \supset (p \ni\!- (q \supset r))$;

g. $\vdash_{\mathbf{S}} (p \ni\!- (q \wedge r)) \supset ((p \wedge q) \ni\!- r)$;

h. $\vdash_{\mathbf{S}} (p \ni\!- \neg q) \vee ((p \ni\!- (q \supset r)) \supset ((p \wedge q) \supset r))$.

In the same paper they have shown that the calculus **S** is consistent. The consistency result is, of course, very important when considering the usability of the calculus.

It is also possible to develop a model theory for the calculus **S**, according to which it is possible to define the truthness of a wff 'p', i.e. to say when 'p' is *true* in an **S**-model. We will say that a wff 'p' is **S**-valid just in case 'p' is true in every **S**-model. The model theory for **S** is related to **S** in the usual ways. In particular, it is possible to obtain soundness and completeness results for **S**. For those results we refer to [6].

For reasons to be considered later, we will not propose an interpretation for the calculus **S** at this point. Before giving any interpretation we will present two extensions to **S**. The first one has an additional operator '$\ni\!-_u$' and the same inference rules as **S**. In addition to the axioms A1-A6, the following wffs are axioms of the calculus **U**:

A7. $(p \ni\!-_u q) \supset \neg\Box(p \supset q)$;

A8. $(p \ni\!-_u q) \supset \neg(\neg q \ni\!-_u \neg p)$;

A9. $(p \ni\!-_u q) \supset (p \ni\!- q)$;

A10. $(p \ni\!-_u (q \supset r)) \supset ((p \ni\!-_u q) \supset (p \ni\!-_u r))$;

A11. $(\Box(p \supset q) \wedge (p \ni\!-_u q)) \supset (\Diamond(p \wedge r) \supset ((p \ni\!-_u r) \vee \Box(p \supset r)))$;

A12. $((p \wedge q) \ni\!-_u r) \supset ((p \ni\!-_u r) \wedge (q \ni\!-_u r))$;

A13. $(p \ni\!- (q \ni\!-_u r)) \wedge \Diamond(p \wedge r) \supset (((p \wedge q) \ni\!-_u r) \vee \Box((p \wedge \supset r)))$;

A14. $((p \ni\!-_u q) \wedge \Box(q \supset r)) \supset ((p \ni\!-_u r) \vee \Box(p \supset r))$.

For any well formed formula 'p' and the set Γ of wffs, '$\vdash_{\mathbf{U}} p$', '$\Gamma \vdash_{\mathbf{U}} p$', '$\Gamma$ is **U**-consistent', and 'Γ is maximally **U**-consistent' are defined in the usual way.

It is possible to show that the following are theorem schemata for **U**:

a. $\vdash_{\mathbf{U}} \neg\Box(p \ni\!-_u q)$;

b. $\vdash_{\mathbf{U}} (p \ni\!-_u q) \supset (\Diamond p \wedge \Diamond\neg q)$;

c. $\vdash_{\mathbf{U}} (p \ni\!-_u q) \supset \neg(p \ni\!-_u \neg q)$;

d. $\vdash_{\mathbf{U}} (p \ni\!-_u q)\neg\Box(p \supset \neg q)$;

e. $\vdash_{\mathbf{U}} (p \ni\!-_u (q \supset r)) \supset (\Box q \supset (p \ni\!-_u r))$;

f. $\vdash_{\mathbf{U}} (p \ni\!-_u (q \supset r)) \supset (\Box(p \supset q) \supset (p \ni\!-_u r))$;

g. $\vdash_{\mathbf{U}} (p \ni\!-_u q) \supset (p \ni\!-_u (p \wedge q))$;

h. $\vdash_{\mathbf{U}} ((p \mathrel{\ni\!-}_u q) \vee (p \mathrel{\ni\!-}_u r)) \supset ((p \mathrel{\ni\!-}_u (q \vee r)) \vee \Box(p \supset (q \vee r)))$;

i. $\vdash_{\mathbf{U}} ((p \mathrel{\ni\!-}_u q) \wedge (p \mathrel{\ni\!-}_u r)) \supset (p \mathrel{\ni\!-}_u (q \wedge r))$.

In addition to the proof of the theorem schemata, Fetzer and Nute [6] have shown that the calculus **U** is, in fact, consistent.

As in the case of the calculus **S**, it is possible to develop a model theory for the calculus **U**, according to which it is possible to define the truthness of a wff 'p', i.e. to say when 'p' is *true* in a **U**-model. We will say, as in the case of **S**, that a wff 'p' is **U**-valid just in case 'p' is true in every **U**-model. The model theory for **U** is related to **U** in the usual ways. In particular, it is possible to obtain soundness and completeness results for **U**. For those results we refer to [6].

The other extension to **S** is actually an extension to **U**, too. The new calculus **C** is obtained by introducing a new operator '$\mathrel{\ni\!-}_n$', where $n \in [0,1]$. The inference rules are the same as for S and U. In addition to the axioms A1-A14, the following wffs are axioms of the calculus **C**:

A15. $(p \mathrel{\ni\!-}_n q) \supset (\neg(p \mathrel{\ni\!-} q) \wedge \neg(p \mathrel{\ni\!-} \neg q))$;

A16. $(p \mathrel{\ni\!-}_n q) \supset \neg(\neg p \mathrel{\ni\!-}_m \neq q)$;

A17. $((p \mathrel{\ni\!-}_u q) \wedge \Box(p \supset q)) \supset ((p \mathrel{\ni\!-}_n (q \supset r)) \supset (p \mathrel{\ni\!-}_n r))$;

A18. $((p \mathrel{\ni\!-}_1 q) \supset ((p \mathrel{\ni\!-}_n (q \supset r)) \supset (p \mathrel{\ni\!-}_n r)))$ for $n \neq 0$;

A19. $((\Box(p \supset q) \wedge (p \mathrel{\ni\!-}_n r)) \wedge \neg(p \mathrel{\ni\!-} \neg r)) \supset ((p \mathrel{\ni\!-}_n r) \vee (p \mathrel{\ni\!-} r))$;

A20. $((p \wedge q) \mathrel{\ni\!-}_n r) \supset (((p \mathrel{\ni\!-} q) \vee (p \mathrel{\ni\!-}_1 q)) \supset (p \mathrel{\ni\!-}_n r))$;

A21. $((p \mathrel{\ni\!-} (q \mathrel{\ni\!-}_n r)) \wedge \neg(p \mathrel{\ni\!-} \neg r)) \supset (((p \wedge q) \mathrel{\ni\!-}_n r) \vee ((p \wedge q) \mathrel{\ni\!-} r))$;

A22. $(p \mathrel{\ni\!-}_n q) \equiv (p \mathrel{\ni\!-}_{1-n} \neg q)$;

A23. $(p \mathrel{\ni\!-}_n q) \supset \neg(p \mathrel{\ni\!-}_m q)$ for $m \neq n$;

A24. $((p \mathrel{\ni\!-}_u q) \vee (p \mathrel{\ni\!-}_1 q)) \supset ((p \mathrel{\ni\!-}_n r) \supset (p \mathrel{\ni\!-}_n (q \wedge r)))$;

A25. $(((p \mathrel{\ni\!-}_m q) \wedge (p \mathrel{\ni\!-}_n r)) \wedge (p \mathrel{\ni\!-}_k (q \wedge r))) \supset$
$((p \mathrel{\ni\!-}_{(m+n)-k} (q \vee r)) \vee (p \mathrel{\ni\!-}_m (q \vee r)))$;

A26. $\Box\neg(p \wedge q) \supset (((r \mathrel{\ni\!-}_m p) \wedge (r \mathrel{\ni\!-}_n q)) \supset$
$((r \mathrel{\ni\!-}_{m+n} (p \vee q)) \vee ((r \mathrel{\ni\!-} (p \vee q)))))$.

For any well formed formula 'p' and the set Γ of wffs, '$\vdash_{\mathbf{C}} p$', '$\Gamma \vdash_{\mathbf{C}} p$', '$\Gamma$ is **C**-consistent', and 'Γ is maximally **C**-consistent' are defined in the usual way.

Note that according to the definition of **C**, it is obvious that '$p \mathrel{\ni\!-}_1 q$' and '$p \mathrel{\ni\!-}_u q$' are not the same thing. Also it is true that '$p \mathrel{\ni\!-}_1 q$' and '$p \mathrel{\ni\!-}_u \neq q$' are not the same. Although the frequency of an event suggests probability 1, it is not certain that the event holds universally[10, p. 5]. Therefore '$p \mathrel{\ni\!-}_1 q$' does not guarantee that q comes after p.

The following are theorem schemata for **C**:

a. $\vdash_{\mathbf{C}} (p \mathrel{\ni\!-}_n q) \supset (\neg\Box(p \supset q) \wedge \neg\Box(p \supset \neg q))$;

b. $\vdash_{\mathbf{C}} (p \ni\!\!-_n q) \supset (\neg(p \ni\!\!-_u q) \wedge \neg(p \ni\!\!-_u \neg q))$;

c. $\vdash_{\mathbf{C}} (p \ni\!\!-_n q) \supset (\diamond p \wedge \diamond\neg q)$;

d. $\vdash_{\mathbf{C}} (p \ni\!\!-_n q) \supset \diamond(p \wedge q)$.

The required proof for the theorem schemata can be found from [6], there is also a proof for the consistency of the calculus **C**.

The matter of a model theory for **C** differs from the developmet of model theories for **S** and **U**. Fetzer and Nute have themselves developed more than one model theory for the calculus **C**, but, as the authors say, the difference between the theories is much in the intended interpretation, and the choice of the interpretation is not clear. It is, anyway, possible to develop a model theory for **C**, in which case it is more or less easy to say that a wff '*p*' is true in a **C**-model just in the case that an appropriate condition is fulfilled. It is possible to say, in the same way, that a wff '*p*' is **C**-valid just in case '*p*' is true in every **C**-model.

Although it is possible to show that every theorem of the calculus **C** is **C**-valid, it may be impossible to obtain a completeness result for the calculus **C**, at least with current methods used in such proofs [6]. It seems, however, that the impossibility is more due to the nature of the procedures by which such results are achieved. It seems to be the case that new techniques will be called for before a completeness proof may be possible.

Chapter 6

Model based reasoning about natural ecosystems: An algorithm to reduce the computational burden associated with simulating multiple biological agents

Jarmo J. Ahonen
Department of Computer Science, University of Joensuu, P.O.Box. 111, 80101 Joensuu, Finland
Hannu Saarenmaa
Finnish Forest Research Institute, Unioninkatu 40 A, 00170 Helsinki, Finland

December 4, 1991

Abstract

Model-based reasoning and industrial fault diagnosis offer intriguing possibilities for solving forest health problems that have proven very complicated. The first requirement for the use of MBR is to build object-oriented models of trees, plants, and various harmful agents. Unfortunately, an object oriented model of a live tree is very heavy in the computational sense because such a model must span over several hierarchical levels. The problem of computational complexity may, however, be relieved by using simulation in cases that have not been encountered earlier. This can be achieved by using a simple method which allows the system to "learn" from its earlier behavior. In this paper such a method is presented. The use of MBR for diagnostic tasks in ecology is also discussed.

6.1 Introduction

Biological systems are inherently very complex and their behavior is often difficult to predict. The very name of the chaos theory is the best manifest of this. Yet the workings of the individual basic units, such as leaves of a tree, are well understood. The difficult issue is the composition of the whole from the parts. Already in the fifties, a theory of hierarchy was created to deal with these issues. Later it has been shown how simple rules give rise to complex structure and behavior (Lindenmayer 1968, 1987, Mandelbrot 1982, Prusinkiewicz and Hanan 1989).

Scientists that work with computer aided decision making in the fields of agriculture, forestry, and other natural resources management are increasingly turning to knowledge-based systems to be able to do better planning among all the agents that interact in the natural environment (Stone and Engel 1990). Individual problems such as animal/habitat interaction (Saarenmaa et al. 1988, Saarenmaa and Nikula 1989, Stone 1990), tree growth simulation (Lorenz et al. 1989), and forest fire management (Cohen et al. 1989) have been solved with object-oriented programming and multiple agents working on a geographical platform. But making true model-based multi-agent multi-model planning work in natural resources management requires taking control over the complexity of the whole natural environment. This requires models of natural objects that span over several hierarchical levels and are instantiable to any of these.

The requirements that are set for these natural objects are transparency and scalability, and a true correspondence of the model representation with the natural objects. Transparency of these biotic agents is necessary because they must be accessible to the external planners for manipulation and explanation. Scalability means that the models must be zoomable from a large to a miniature level of abstraction. For instance, a silviculturist planner must be able to look at the forest as a whole, as a tree, as parts of tree such as trunk attributes, and finally as the foliage elements that may show symptoms of damage. The third requirement, the one-to-one representation rises from these two first ones: scalability has meaning only if the planner can assume to find a meaningful set of objects at a more detailed level should it be necessary to dive therein.

The logical consequence of this modeling approach is that the models embed a complex topology, and that the computational burden becomes overwhelming. Yet the whole computation is rarely needed, because the likelihood that a planning agent would need the data from a particular object is small. Some solutions to this are massive data-parallelism and concurrent object-oriented languages (Agha 1990). Other solutions for general purpose hardware and software are also needed. In this paper we present a simple method how a complex hierarchical model that describes the topological structure of a tree can be made to learn from its previous behavior. This way, the computation burden can be reduced without compromising the true topological representation of the tree and its functioning.

6.2 Model structure

DEEP-TREE is a model-based diagnostic system of forest damage (Saarenmaa 1988). The goal of the DEEP-TREE project is to have a single representation of a tree's structural topology and functioning that can be used by different consultation and planning systems. The architecture of the DEEP-TREE model derives from the model-based diagnosis of industrial devices (DeKleer and Williams 1987, Herbest and Williams 1987). Some of the goals are causal explanation of the mechanisms of forest damage and predictions for growth and development, but also general issues of forest management.

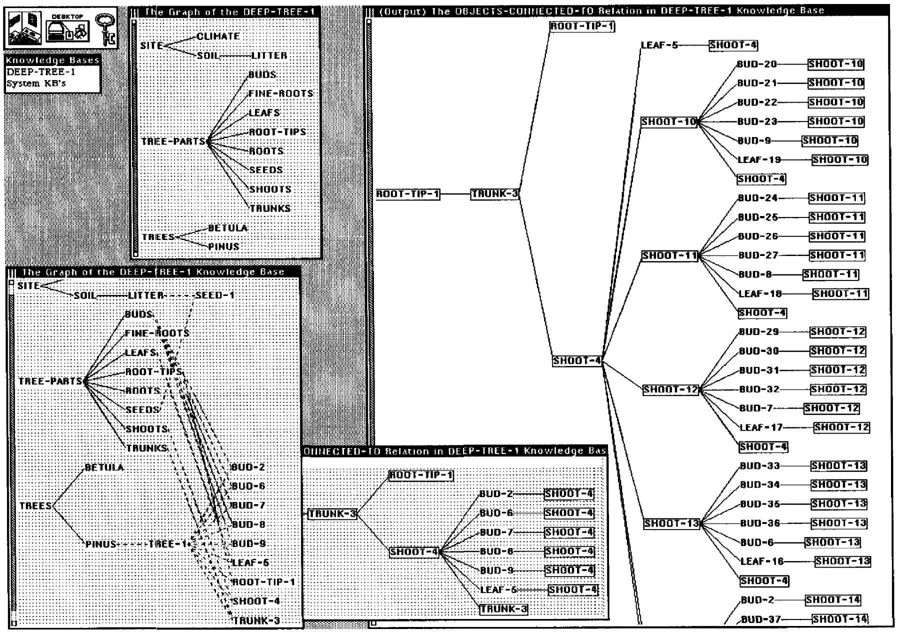

Figure 6.1: The class and object hierarchy of the DEEP-TREE model and its topological structure in a slot graph. The initial classes and the growth of the tree during the two first years is shown.

DEEP-TREE is a prototype that has been written in KEE. Its class hierarchy is apparent from the smaller two windows in Figure 6.1. There are three hierarchy levels which are 1) the whole tree, 2) tree part compound objects (trunk, foliage, etc.), and 3) individual organs (buds, shoots, leaves, etc.). While a single tree is one object, and all the tree parts total about ten objects, individual organs cumulate up to tens of thousands in a mature tree. The tree is not built ready as full grown. Instead, it is always generated from a seed, a single object. The topology of the tree is formed by methods of

Class	Generates	Connected to
BUDS	SHOOTS, BUDS	SHOOTS
LEAVES	-	SHOOTS
SHOOTS	LEAVES	BUDS, LEAVES, TRUNK, BARK
TRUNK	a sink	SHOOTS, ROOTS, BARK
BARK[a)]	TRUNK	TRUNK, SHOOTS, ROOTS
ROOTS	FINE-ROOTS	TRUNK, BARK, FINE-ROOTS, ROOT-TIPS
FINE-ROOTS	-	ROOTS
ROOT-TIPS	FINE-ROOTS, ROOT-TIPS	ROOTS
SEED	ROOT-TIPS, BUDS	detaches after germination
LITTER	a sink	-

[a)] New, not in Figure 6.1.

Table 6.1: Tree organs and the topology they create with their methods.

organs (Table 6.1). The development of the structural topology of the organs of a two year old tree is shown in figure 6.1.

In a live tree, each of the organs also functions. Currently, DEEP-TREE is made to simulate the effects of carbon/nutrient (mainly nitrogen) dynamics. C/N balance is the most important factor that determines how palatable the tree is to herbivores such as moose, voles, and harmful insects (Bryant et al. 1983, Tuomi et al. 1984). A detailed description of the mechanisms and effects of the C/N balance on pest susceptibilty is beyond the scope of this paper, but it is necessary to know that it varies by season, age of organ, soil type, damage in organ, and damage elsewhere in the tree. To be able to perform diagnostic reasoning of the tree model, the C and N levels of the target objects must be known at all times as well as the damage status of the organs.

From this design raises the problem that we need to compute the functioning and status of all organs. For instance, a herbivore attack to leaves on the upper crown leaves causes N levels to rise at the lower crown because now there is more N available. This rises the susceptibility of the lower crown leaves to further defoliation. On the other hand, the tree defends itself with C-based compounds, and induces their synthesis after defoliation, causing further N-excess, but also a higher level of defence in the whole tree. This example shows that on a sudden disturbance the needs for computation become very large . Even by combining all the leaves connected to a shoot into a single object, there are tens of thousands of leaves in a mature tree. They all need to be recomputed in this case.

So far, DEEP-TREE has been designed for quantitative models of tree physiology, mainly outlined by Landsberg (1986). Qualitative modeling and reasoning will be studied in future, because of the explanations and robustness they provide.

6.3 The Method

From the brief description of the tree model it is easy to see that the computational effort used to compute required values may be enormous. One tempting escape from the computational complexity is to use previously computed values to approximate the function of the objects of the model. Since the use of the described type of models may be defended by their ability to behave correctly in very different situations, it is obvious that such approximations are useful only in cases which have a very close resemblance to previously encountered cases. One possible method of performing such an approximation without losing the positive effects of the use of hierarchical object-oriented models may be to use a simple method by which the system can "learn" from its earlier function. In the following text the notation $\overline{X}$ denotes an n-ary vector.

```
Algorithm FIND-APPROX:
1) IF there are previously computed values for parameters (hereafter
      denoted by $\overline{V}$), according to which an object functions, within
      a chosen range THEN
2)    RETURN(compute-approx($\overline{V}$))
3) ELSE
4)    the state of the specific level, denoted by $\overline{value}$ = compute-value($\overline{V}$);
      "compute values by using the objects themselves"
5)    save-approx($\overline{value}$, $\overline{V}$); "save sufficient information for later use"
6)    RETURN($\overline{value}$) "return the value computed using the actual objects"
7) ENDIF
8) END.
```

The algorithm FIND-APPROX is called every time an object of any object-class present in the model is intended to be used. The aim is to use the actual object only in cases when previously computed values do not provide a possibility of calculating an accurate enough approximation of the behavior of the actual object. In practice this means that if we already know how a shoot behaves in certain conditions, we can approximate its behavior without computing the exact behavior of the shoot and its leaves etc. If the function of an object is explicitly computed, then information of the parameters and the computed values are saved into a balanced binary tree for further use. Before the actual saving of the information required for later use is done, the algorithm SAVE-APPROX computes values which are later used to compute approximations of almost similar cases.

```
Algorithm SAVE-APPROX($\overline{v}$,$\overline{V}$):
1) $\overline{d}_l$ = change($\overline{v}, \overline{V}, -$);
   $\overline{d}_u$ = change($\overline{v}, \overline{V}, +$);
   "$\overline{d}_l$ and $\overline{d}_u$ are vectors which will be used when computing approximations"
2) save_in_balanced_btree($\overline{v}$,$\overline{V}$,$\overline{d}_l$,$\overline{d}_u$);
3) END.
```

The function CHANGE returns vector which has as many dimensions as $\overline{v}$, and each of the dimensions of the returned vector $\overline{d}_u$ is computed by

$$\overline{d}_{u_i} = -\frac{\overline{v}_i - \text{nearest\_value}(\overline{V}, \overline{v}_i, +)}{\overline{V}_i - \text{nearest\_point}(\overline{V}, +)}$$

in which the function NEAREST_VALUE returns $\overline{v}_i$ in case that there is no previously computed point $\overline{p}$ for which $\overline{V}_i \leq \overline{p}_i \leq \overline{V}_i + \overline{\epsilon}_i$ holds when i ranges over the dimensions of $\overline{p}$, and where the symbol $\overline{\epsilon}$ denotes the chosen approximation shift (i.e. a half of the range for which the current approximation may be used). The function NEAREST_POINT returns $\overline{p}_i$ in the case that such a previously computed point exists in the direction dictated by '-' or '+' (note that the direction is same for NEAREST_VALUE and NEAREST_POINT).[1] In a similar way, each of the dimensions of $\overline{d}_l$ is computed by

$$\overline{d}_{l_i} = \frac{\overline{v}_i - \text{nearest\_value}(\overline{V}, \overline{v}_i, -)}{\overline{V}_i - \text{nearest\_point}(\overline{V}, -)}$$

in which the function NEAREST_VALUE returns $\overline{v}_i$ in the case that there is no previously computed point $\overline{p}$ for which $\overline{V}_i - \overline{\epsilon}_i \leq \overline{p}_i \leq \overline{V}_i$ holds.

The algorithm SAVE_IN_BALANCED_BTREE saves its parameters to a balanced binary tree. The balancing and other operations performed to the tree are managed by commonly known binary-tree algorithms.

```
Algorithm COMPUTE-APPROX()
$\overline{out} = \{\}$;
$(\overline{v}, \overline{p}, \overline{d}_l, \overline{d}_u, direction)$ = return_from_btree(nearest_point($\overline{V}$));
   "At this point the earlier values, the parameters according to
   which the values were computed, the vectors used to compute the
   approximation, and the direction in which the new parameters are when
   compared to the earlier parameters, are returned from the tree"
IF $direction$ = '+' THEN
      $\overline{d} = \overline{d}_u$
   ELSE
      $\overline{d} = \overline{d}_l$
FOR $i$ ranges over the dimensions of $\overline{v}$ DO
      $\overline{out}_i = \overline{v}_i + \overline{d}_i|\overline{V}_i - \overline{p}_i|$
      "compute the approximated value for every dimension of the value vector"
ENDDO
RETURN($\overline{out}$)
END.
```

The actual approximation algorithm COMPUTE-APPROX searches the nearest previously computed point and saves the values found from the tree to local variables. In addition to the saved values, the function RETURN_FROM_B-TREE returns the direction in which the new point is compared to the nearest

[1] If the dividend turns out to be zero, then zero is returned automatically and the divisor will not even be computed.

previously computed value. From the direction in which the new parameters are the vector $\overline{d}_l$ or the vector $\overline{d}_u$ is chosen. The values of the dimensions of the chosen vector are then used to compute the values of the returned approximation. The chosen vector provides values from which it is possible to guess the difference between the value computed at $\overline{p}$ and the value at $\overline{V}$, i.e. the value at $\overline{V}$ is approximated by using the previously computed value at $\overline{p}$, and an estimate of the amount of the change which depends on the difference between $\overline{V}$ and $\overline{p}$.

One obvious feature of the method is that if $\overline{\epsilon}$ is large, we lose accuracy but obtain more improvement in the computational sense. In this respect, the choice of $\overline{\epsilon}$ depends on the nature of the modelled phenomenon or objects. If the behavior of the modelled phenomenon or object has very rapid changes, the approximation shift $\overline{\epsilon}$ has to be kept small.

An unfortunate feature of the method is that it may require a large amount of space for the binary tree in which the previously computed values and other information are saved. The requirement of space for saved information may cause one to be tempted to use a larger $\overline{\epsilon}$, in which case some accuracy is lost, but space saved. Obviously it is important to find a balance between the requirement of accuracy versus the requirement of small space for generated information and computational savings. Further improvements include improving the method used to compute the approximation or creating a mechanism by which different approximation shifts and spaces defined by such shifts may be combined, the usability of the proposed method may obviously be improved from its current state.

6.4 Discussion: the tasks in ecological fault diagnosis

The forest damage that the model is intended to diagnose can be caused by biological agents such as insects, or it may be the effect of environmental change. The knowledge that describes these types of “faults” in the forest ecosystem is stored in two class hierarchies that can be loaded next to the DEEP-TREE model. The actual problem solving knowledge is also stored in its own class hierarchy. Detailed description of these is provided elsewhere (Saarenmaa 1988; Saarenmaa et al 1991), but their use with the present model is briefly discussed here.

The biological agents that may attack trees form one taxonomy that is arranged according to the Linnean classification. For each taxa, such as the family of bark beetles or a disease such as root-rot, the classes of those tree parts that the agents attack are specified. This knowledge, as well as other pertinent information of the agent’s distribution, life cycles, appearance, symptoms, etc. is stored in the slots of the classes built with KEE (figure 6.2).

The conditions and symptons of abiotic disturbances, such as nutrient deficiences, air pollution, etc. are also stored in a similar class hierarchy. All these are prototypical knowledge of various faults.

The problem-solving knowledge is stored in yet another class hierar-

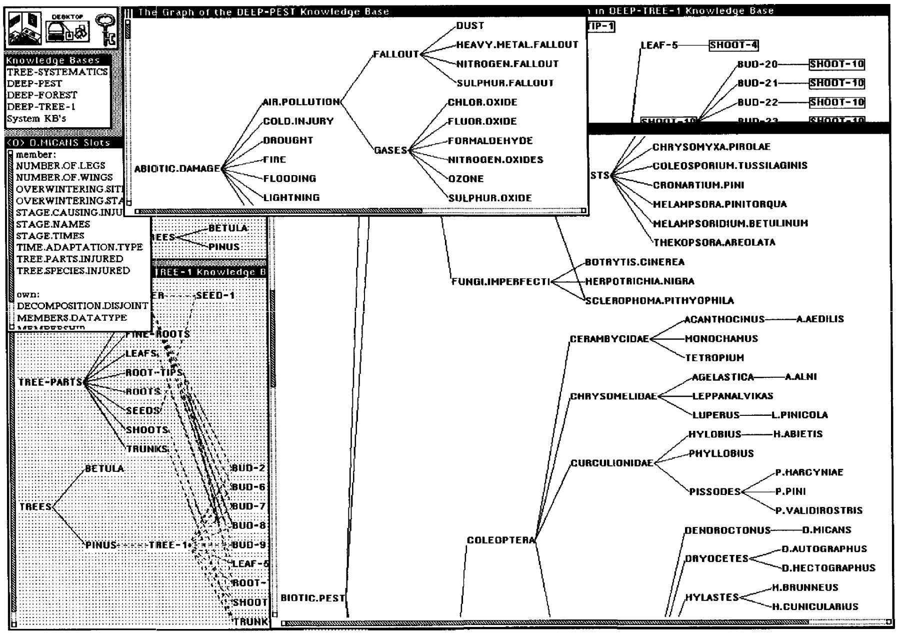

Figure 6.2: The class hierarchy of the DEEP-PEST knowledge base.

chy, called "expert agents". These perform the reasoning that match the pests and disturbances with the observed symptons. Each expert agent has a problem solving agenda. The agenda of a diagnosis expert has the following steps:

1. gather initial data;
2. reason possible causes from among the protypical faults and pests with forward chaining;
3. compare the symptons in the tree with those of the prototypical faults and pests with backward chaining; and
4. reason the causality and create the explanations.

The fourth step is particularly complex, since it requires generate-and-test experimenting with the present DEEP-TREE model. This is where the present algorithm has a central role.

Let us consider a case in which the system performs a generate-and-test diagnosis for a large number of trees. If every feature of every object presented in the system has to be explicitly computed, the overall usability of the system might be questionable due to computational reasons. But in an actual simulation several similar objects function in almost similar conditions, and if the function of one object in such conditions has been computed, then the function of such objects in similar conditions may be approximated by using the previously computed values. At first such approximations can be done for some of the leaves of one tree, and after that for some of its shoots, etc. Finally the functioning of trees may be approximated by using previously computed knowledge of the functioning of trees in similar conditions. Currently, we are working on steps 3 and 4 mentioned above.

After the diagnosis expert agent has done its job, any prescriptions of silvicultural measures will be made by the silviculturist expert agent. This agent is supposed to deal with the real causes of the problem as determined by the diagnosis expert. In essence, it is planned to solve and remedy the forest health problem by the cooperation of multiple experts reasoning about the multiple simulations and fault models. This kind of MBR systems have been proposed and designed for a variety of industrial problems (see, e.g., DeKleer and Williams 1987), but they have not been tried in natural resource management (Stone and Engel 1990). Since natural ecosystems are open, chaotic, and poorly known in many cases, applying industrial fault diagnosis methodology in this field represents a formidable challenge.

References

Agha, G. 1990. Concurrent object-oriented programming. *Communications of the ACM* 33(9): 125-141.

Bratko, I., Mozetic, I. & Lavrac, N. 1989. *KARDIO. A study in deep and qualitative knowledge for expert systems.* 260 pp. The MIT Press, Cambridge, Massachusetts.

Bryant, J.P., Chapin, F.S.III & Klein, D.R. 1983. Carbon/nutrient balance of boreal plants in relation to vertebrate herbivory. *Oikos* 40: 357-368.

Cohen, P.R. , Greenberg, M.L., Hart, D.M. & Howe, A.E. 1989. Trial by fire: understanding the design requirements for agents in complex envionments. *AI Magazine* 10(3): 34-48.

DeKleer, J. & Williams, B.C. 1987. Diagnosing multiple faults. *Artificial Intelligence* 32: 97-130.

Herbert, M.R. & Williams, G.H. 1987. An initial evaluation of the detection and diagnosis of power plant faults using a deep knowledge representation of physical behavior. *Expert Systems* 4(2): 90-95.

Landsberg, J.J. 1986. *Physiological Ecology of Forest Production.* Academic Press, London. 198 pp.

Lindenmayer, A. 1968. Mathematical models of cellular interaction in development. I & II. *Journal of Theoretical Biology* 18: 280-315.

Lindenmayer, A. 1987. Models for multicellular development: characterization, inference, and complexity of L-systems. In: Kelmenova and Kelmen (eds.) *Trends, techniques and problems in theoretical computer science.* Lecture Notes in Computer Science 281: 138-168.

Lorenz, G., Uhrmacher, A., Simon, K.-H. & Bossel, H. 1989. Application of artificial intelligence methods to the representation and modeling of tree growth. 234-243. In: Burkhart, H.E., Rauscher, H.M. & Johann, K. (eds.) 1989. *Artificial Intelligence and Growth Models for Forest Management Decisions.* Proceedings of IUFRO Meeting, Vienna, Austria, September 18-22, 1989. FWS-1-89. 419 pp. School of Forestry and Wildlife Resources, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Mandelbrot, B. 1982. *The Fractal Geometry of Nature.* W.H. Freeman, New York. 468 pp.

Prusinkiewicz, P. & Hanan, J. 1989. *Lindenmayer systems, fractals, and plants.* Lecture Notes in Biomathematics 79. 120 pp.

Saarenmaa, H. 1988. Model-based reasoning in ecology and natural resource management. 141-157. In: Buhyoff, G. (editor) *Resource Technology 88.* Proceedings of International Symposium on Advanced Technology in Natural Resource Management. Fort Collins, Colorado, June 19-23, 1988.

Saarenmaa, H. , Stone, N.D., Folse, L.J., Packard, J.M., Grant, W.E., Mäkelä, M.E., & Coulson, R.N. 1988. An artificial intelligence approach to simulating animal / habitat interactions. *Ecological Modelling* 44: 125-141.

Saarenmaa, H. & Nikula, A. 1989. Managing moose damage in forest plantations: a deep model of animal behavior on a geographic information system platform. 825-831. Conference Proceedings. *9th International Workshop on Expert Systems and Their Applications*, Avignon, France, May 29 - June 2, 1989.

Saarenmaa, H. & all. 1991. Object-oriented design of topological tree and forest models. *First European Symposium on Terrestrial Scosystems: Forest and Woodlands.* Florence, Italy. May 20-24, 1991.

Sticklen, J. & Chandrasekaran, B. 1985. Use of deep level reasoning in medical diagnosis. In: Karna, K.N. (ed.) Expert Systems in Government Symposium. 496-501. Publ. IEEE.

Stone, N.D., Engel, B.A. 1990. Knowledge-based systems in agriculture and natural resource management. *AI Magazine* Fall/90: 20-22.

Tuomi, J., Niemelä, P., Haukioja, E., Siren, S. & Neuvonen, S. 1984. Nutrient stress: an explanation for plant anti-herbivore responses to defoliation. *Oecologia* 61: 208-211.