

Reusability of simulation models

Anita Pos^a, Pim Borst^a, Jan Top^b, Hans Akkermans^{a,c}

^aInformation Systems Department, University of Twente, P.O. Box 217, NL-7500 AE Enschede, Netherlands

^bAgro-Technical Research Organization ATO-DLO, P.O. Box 17, NL-7500 AA Wageningen, Netherlands

^cNetherlands Energy Research Foundation ECN, P.O. Box 1, NL-1755 ZG Petten, Netherlands

Abstract

Modelling and simulation are widely used methods in design and other engineering tasks. Providing libraries of reusable model fragments is a promising way of supporting engineering modelling. The paper discusses means of structuring such libraries in a generic and reusable way. Model content facts can be organized on the basis of the *ontologies* that are generally important in physical systems engineering. In addition, the paper suggests that such libraries need to support various kinds of *modelling assumptions* about model fragments. Modelling can be usefully approached as a form of assumption management, as a means of reflecting the metalevel knowledge that is employed by experienced engineers. A categorization of modelling assumptions is proposed that aims at further support for reusability and shareability.

Keywords: Engineering modelling; Reusability; Model fragment libraries; Assumption management

1. Introduction

The construction of models is an important aspect of problem solving in engineering disciplines. A model is used as an artificial structure for investigating the behaviour of an engineering system. For example, in design tasks, modelling and simulation are increasingly used approaches to evaluating alternative design solutions for physical systems. This approach is cheaper and faster, and it produces higher quality than an approach that is based on prototyping and experimentation.

However, engineering modelling is in itself a complex and knowledge-intensive task and hence the question arises as to how this task can be supported. One idea which is currently around in both AI and various engineering disciplines is to supply libraries of reusable models and model fragments [1–4]. Redesign of simulation models (or ‘remodelling’, for short) is based on the reuse of previously constructed simulation models. The general idea in reusing existing models is that adapting a previously constructed model will be easier and less error-prone than designing a new simulation model from scratch, thus speeding up the entire modelling and design process.

At present, there is not yet a clear consensus on the organizing principles underlying reusable model

libraries. For real-life applications such libraries tend to become very big, so that the stored information must be made available in very structured ways in order to be helpful to the modeller. In addition, the aim of reusability introduces its own requirements. It is certainly possible to build a structured library for a specific application area, but it is quite another matter to come up with organizing principles that are valid across many domains. However, this is necessary because many large-scale applications cut across more than one domain of physical systems engineering. For example, heating systems, cars and machine tools (all fields for which reusable model libraries are being developed) are composed of hydraulic, thermal, control, electrical and mechanical components.

In this paper we suggest two different ways of enhancing the reusability of model fragment libraries. First, there are generic *ontologies* with respect to physical systems that can be employed to increase the structuredness of libraries and to constrain the modelling decisions that have to be taken. Secondly, experienced engineers have a lot of metaknowledge about the models they deal with. It would be very useful for model libraries not only to provide the content of model fragments (such as equations, diagrams, and formal graphs), but also to be able to handle metaknowledge about models and modelling.

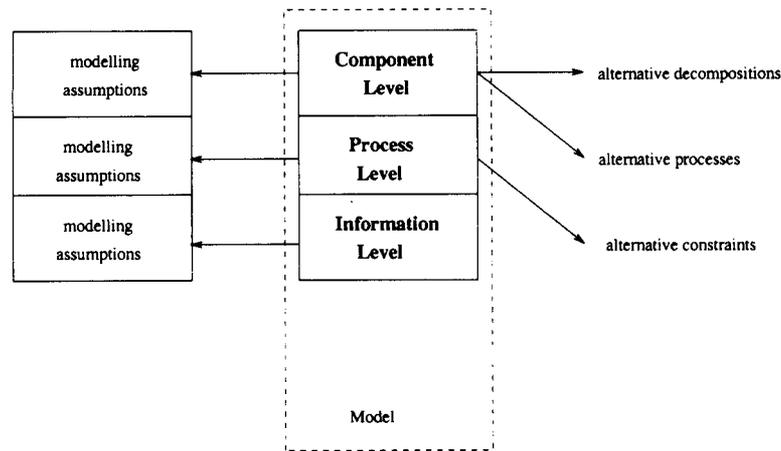


Fig. 1. Library organization.

Such metainformation is often viewed by physicists and engineers as *modelling assumptions*. Usually many different models exist for a physical device, and the validity of each model depends strongly on its context. Modelling assumptions can be useful in restricting and exploring the model space in order to find a correct model for a specific context. Furthermore, underlying modelling assumptions can provide clues for suitable modelling revisions, thereby focusing the process of redesign. In this paper we outline how a reusable categorization of modelling assumptions may be achieved.

In Section 2 we discuss our ontological organization of model libraries. This section focuses on the question of how the content of model fragments can be structured and stored in a way that is generic across a large range of different applications and tasks. In Section 3 we consider how metaknowledge about models may be laid out in terms of different, generic types of modelling assumptions, by way of a simple example. Then, in Section 4, we show that this approach can also be used to structure user requirements, providing a bridge between the informal problem and goal statement expressed by the user and the formal representation of modelling assumptions necessary in automated modelling. Section 5 discusses related work and suggests that recent advances in knowledge acquisition [5] may be relevant here. Achieving reusability of libraries can be based on a knowledge modelling approach to the expertise of the engineers who are the end users of model fragment libraries.

2. Ontological organization of library models

During the modelling process, engineering systems are considered from various different ontological points of view. The representation of the library contents of a model must reflect this. Furthermore, it is useful to *separately* store in a library the ontological views of a model for various reasons. It imposes more data and knowledge structure which is important for large

libraries. This also allows modelling decisions to be made in a more evolutionary piecemeal fashion. This reduction of the grain size of both model content and modelling decisions further improves the reusability of the information in a library.

Our library structure, depicted in Fig. 1, shows three ontological viewpoints on physical systems: system components, physical processes and mathematical relations. Each ontological level has its own representation, and a complete physical model comprises all of these three levels. The component level provides information on is-a and part-of hierarchies relating to the system. The process level defines which physical mechanisms are present in the system, in the form of a bond graph [6]. The information level provides the set of constraint equations necessary for symbolic analysis and numerical simulation. This framework is used in practice as a basis for organizing the library under development in the European Esprit-III OLMECO project, which is concerned with reusable models for mechatronic component design. The model ontology will be illustrated in some more detail in the next section, in conjunction with the modelling assumption typology presented here.

In addition to the actual model content facts, each model level contains a set of *modelling assumptions*. These assumptions state important meta-aspects of the model, like its properties and various aspects of its behaviour. Our claim is that explicitly representing these meta-aspects of simulation models in the library facilitates the reuse of these models. In the next section a categorization of these modelling assumptions will be presented. For a more detailed description of our model ontology see [3].

3. Organization of modelling assumptions

When modelling a system, modellers do not only consider the contents of the model. They are also interested in all kinds of meta-aspects of the model, like its

properties and various aspects of its behaviour. To facilitate reuse of simulation models, and guide the model revision process, these additional meta-aspects should be represented along with the actual model contents. Therefore, we need a structured and genetic framework for these additional modelling assumptions. As a first step towards this goal, we present in this section a generic classification of these additional modelling assumptions. We suggest the following classes of modelling assumptions:

- **Content assertions:** Content assertions refer to model content facts that are required, available or absent. For example, one can describe which elements are (not) represented in the model contents, e.g. this model does not contain friction.
- **Goal statements:** A goal statement describes the operational goal that is to be achieved with a specific model. Examples are causal explanation or prediction by numerical simulation.
- **Behavioural statements:** Both quantitative and qualitative behavioural aspects of a given model can be represented in terms of behavioural assumptions. Simulation results are a typical example.
- **Metalevel statements:** Metalevel knowledge about a simulation model is described by means of metalevel statements. For example, the knowledge that a model is linear is practically very useful, because it is an indicator that certain classes of problem solving methods are applicable (e.g. transfer function analysis).
- **Validity assumptions:** These assumptions express the situations and conditions in which a model is considered to be valid. For example, a linear model is only valid if the changes in the environment are relatively small.
- **Performance qualifications:** For large models it can be useful to have indicators for the performance aspects of a simulation model, like complexity and the computation time necessary to produce results. These are described by performance qualifications, for example complexity measures such as system order.

Below we will discuss the organization of library models and modelling assumptions in some more detail, illustrated by the following simple running example. The system to be modelled consists of a mass connected by a spring to a solid wall (see Fig. 2). The informal question to be answered is as follows.

Consider the system in Fig. 2; How will the block move in time when it is pulled away from the wall and then released?

3.1. System components

The component level is the initial abstraction level in our model representation. The central concept on the

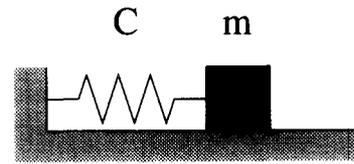


Fig. 2. Diagram of spring–mass system.

component level is the system component, representing a distinguishable object in the real world. At this level, is-a hierarchies and part-of decompositions of systems and subsystems are represented. In addition the system topology is given by means of a diagram containing the system component, their energetic ports and the connections between these ports.

Fig. 3 is a graphical representation of the contents of the component level of the model for our spring–mass example, together with some additional modelling assumptions. Is-a and part-of hierarchies are not necessary in this simple example. Some additional modelling assumptions on this level are as follows:

- **Behavioural statements:** All the behaviour of this system is of a mechanical translational type (this is useful information since it determines the dynamic quantities of interest occurring at the other levels).
- **Metalevel statements:** The model boundary is closed, i.e. there are no disconnected ports (open ports indicate incompleteness of the model).
- **Validity assumptions:** This model is valid if there are no other relevant components in the environment influencing the behaviour of the system.

3.2. Physical processes

The process level describes the physical processes that underly system behaviour. One way of expressing these processes in engineering is called the *bond graph* [6]. In this representation, nodes stand for generic physical mechanisms, such as power sources (Se, Sf), dissipation (R), energy storage mechanisms (C, I), energy conversion (TF, GY) and energy distribution mechanisms (0, 1) based on conservation principles. Edges (or bonds) stand for energy exchange paths. The bond graph as a whole represents the topology of the physical mechanisms active in the system.

Fig. 4 is a graphical representation of the contents of

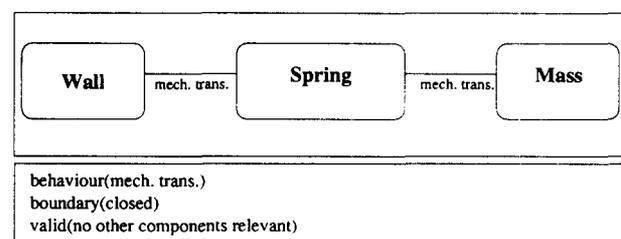


Fig. 3. Component level description of the spring–mass system.

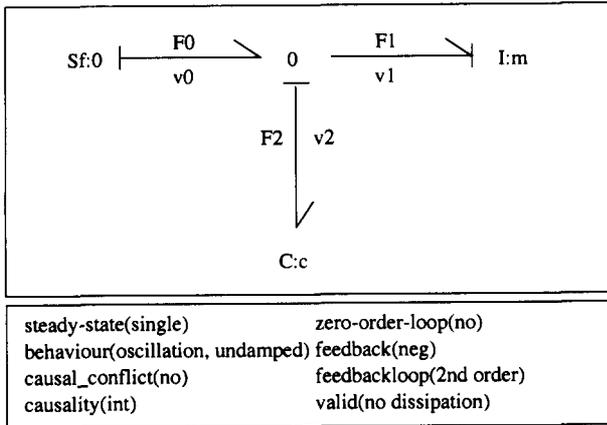


Fig. 4. Process level description of the spring–mass system.

the process level of the model for our spring–mass example, together with some of the additional modelling assumptions for this level. Some additional modelling assumptions on this level are as follows:

- **Behavioural statements:** By examination of the bond graph we can deduce that this model has a single steady state, instead of an infinite number of solutions or no solutions at all. Furthermore, we can deduce that the model will exhibit undamped oscillatory behaviour. These are important qualitative aspects of the behaviour of the system that are obtained without the performance of numerical simulation.
- **Metalevel statements:** By looking at the causality assignments in the model we can see that the model contains no causal conflicts, and that it has strict integrating causality. This is important in most simulation methods, since differentiating causality often causes numerical instability. In addition, integrating causality is necessary to generate causal explanations that fit common sense. The model also contains no zero-order loops, the occurrence of which cause a model to be numerically stiff and thus mean that special simulation methods are required. The model contains a negative feedback loop of order 2.
- **Validity assumptions:** The model is valid under the condition that no dissipation is present, i.e. the system contains no friction.

3.3. Mathematical relations

The information level represents the mathematical structure of the model. This information or signal structure is expressed in terms of (ordinary) differential and algebraic equations, and ranges of values/parameters. An alternative representation is the so-called block diagram which is a standard representation in control engineering.

Fig. 5 shows the contents of the information level of

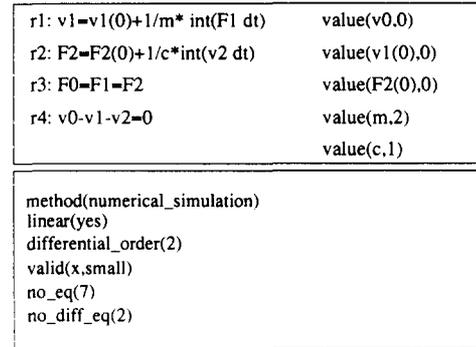


Fig. 5. Information level description of the spring–mass system.

the model for our spring–mass example, as well as some of the additional modelling assumptions, described in terms of parameters, variables and relations between them. Some of the additional modelling assumptions on this level are as follows:

- **Goal statement:** This model is constructed for the purpose of prediction (task) by means of numerical simulation (method).
- **Behavioural assumptions:** The simulation results act as behavioural statements. Because of lack of space they are not included in the picture.
- **Metalevel statements:** The model is linear and of differential order 2. The higher the differential order of a model is, the more difficult it is to analyze the model by symbolic mathematical methods. If the model is nonlinear, there are essentially no symbolic techniques, and only numerical and graphical methods are applicable.
- **Validity assumptions:** The use of a linear spring–mass model is valid under the condition that the displacement of the spring is small. If the displacement becomes larger, the spring will behave in a nonlinear manner.
- **Performance qualifications:** One aspect of performance may be captured by giving complexity measures for the model. There are several possibilities. Since both the number of differential equations and the total number of equations influence the computation time needed to simulate the model, these numbers may be employed as complexity measures. The example model consists of seven equations, two of them differential ones.

4. User requirements

A human modeller usually starts the (re)modelling process with an informal problem statement. During the (re)modelling process this informal problem statement becomes formalized in various steps that go along with the actual construction of the model itself. However, in automated modelling, formal requirements are

a necessity. These formal requirements should on the one hand be close to the concepts used by human modellers, and on the other hand they must be connected with the computational concepts used in automated modelling. Formal requirements thus form a bridge between the human modeller and the automated (re)-modelling system. A generic classification of modelling requirements can help the user in the process of transforming an informal problem statement into a set of formal requirements.

An informal problem statement for redesigning a simulation model usually consists of one or more of the following elements: a specific question to be answered (that is, the modelling goal), observations or experimental data about the physical system that is to be modelled, scenario information about the real-world situation in which the system behaviour occurs, and possibly additional requirements concerning the performance of the simulation model, like complexity or turn-around time. Collectively, these elements constitute the user requirements for the modelling task. The point is that of how to formalize this in a way that is still natural to the engineer. Here, we propose that the classification of modelling assumptions into content, goal, metalevel, behavioural and performance statements, as discussed in the previous section, yields a useful framework for formalizing this kind of informal problem statement. Elements of the informal problem statement can hereby be translated relatively easily into formal modelling assumptions.

Let us consider our spring–mass example again. A possible reason for revising our current model could be the following extension to the original problem statement as given in the preceding section (Fig. 2).

I have in addition some observational data, say in the form of a table of the position of the block as a function of time (see Table 1). The prediction results from the model must comply with these observations.

This extension of the original problem statement can be translated in terms of our typology of modelling assumptions/requirements:

- *Content assertions:* The position of the block will be added to the list of quantities of interest, in addition to velocity and force. This is a content assertion on the information level.
- *Query, validity, metalevel and performance assumptions:* These assumptions for the model under revision remain unchanged (for the time being).
- *Behavioural statements:* The observation data given in Table 1 are added as a behavioural statement about the model.

The newly stored modelling assumptions about the desired aspects of the model act as additional requirements for the model revision process.

Table 1
Observation data for the spring–mass system

t , s	x , cm
0	0.00
1	6.10
2	0.00
3	–2.20
4	0.00
5	0.80
6	0.00
7	–0.30
8	0.00
9	0.10

In the context of reusing engineering models, an important reason for including not only the structural representation of the model but also its underlying modelling assumptions is to focus the process of (re)-design. Violated underlying assumptions provide clues for suitable revisions to the model and thus provide a focus for the remodelling process. From a knowledge acquisition point of view, this is probably a general modelling practice among human modellers. First the modeller considers which assumptions are violated, and only then is the impact of these assumptions on the content facts of the model considered. Experienced physicists and engineers employ a lot of ‘metatalk’ in the modelling process, especially in the earlier stages.

A small illustration of this idea can be provided by considering our previous example. The new observational requirements for the mass–spring system (see Table 1) suggest a damped vibration, i.e. the system loses energy and will come to rest. However, the simulation results from the first model version as given in Section 3 show an undamped oscillation. This constitutes a violation of the behavioural assumptions, calling for a revision of the model. In this example it is immediately clear to any human modeller that some form of dissipation has been neglected. This means that the validity assumption *valid(M, no dissipation)* on the process level (see Fig. 4) is no longer valid, and will have to be altered. This can be done in different ways: by assuming a resistive effect in the spring, or by including friction between the system and the outside world. In the first case, the process level model has to be revised by adding a dissipative mechanism in the bond graph, while the component model and its modelling assumptions remain valid. In the case of including friction between the system and the outside world, the validity assumption *valid(M, no-other-components-relevant)* on the component level (see Fig. 3) is also violated. This can be fixed by adding the floor as a new component to this part of the model, necessitating also a process-level addition for this component. In both cases the performance qualifications also have to be revised since the model complexity measures

in terms of numbers of equations change (e.g. *no-eq(7)* on the information level (see Fig. 5), although in this rather trivial example the impact of this will be practically nil).

We note that model revision can well be approached, in a terminology that is quite natural to the modelling engineer as the end user, from the viewpoint that *modelling assumptions* are the real focus of the modelling process. The previous example shows that modelling assumptions play an extensive role in the (re)modelling process. Model assumptions become violated, and selected for modification. This triggers a new chain of assumption revisions, and only at the end are the model content facts updated. Although our work is still tentative, viewing modelling as *assumption management* seems to be a promising approach.

5. Discussion

Advancing automated modelling of physical systems is important to reduce the engineering bottleneck in the design of technical systems and related tasks. Aiming for reusability of model fragments and of modelling knowledge (two different things, by the way) is a good strategy here, since it helps to reduce the model construction task from a kind of novel design task into a simpler configuration task.

Then, the structuring principles underlying reusable model information are an important research topic. With regard to the contents of stored library models we propose to distinguish and separate the different generic ontologies involved. Engineering modelling evolves around three engineering ontologies: system components, physical processes and mathematical relations. Library models need to reflect these ontologies. As they have their own representations (e.g. system schemas, bond graphs, and constraint equations) as well as their own problem solving methods (e.g. context modelling, causal analysis, and numerical simulation), it is possible to store these ontological views in an engineering model as separate entities. This not only helps to structure the library itself, but also enables the modeller, or the automated modelling system, to limit the number of decisions that have to be taken at a time. In addition, this separation according to ontology increases the reusability of model fragments. We are currently involved, with other companies, in building various libraries along these lines.

As a further step we have put some emphasis on the need to store *assumption knowledge about* the stored library models. The point is that of how to do this in a generic, reusable way, since much of this metatype knowledge at first sight seems to be application-dependent. However, it is possible to single out, within each of the mentioned ontologies, some general classes of modelling assumptions:

- content assertions,
- goal statements,

- behavioural statements,
- metalevel statements,
- validity assumptions,
- performance qualifications.

We considered how this organization of assumptions can be used to describe metaknowledge about engineering models, thereby making them more reusable, and can also be used to structure user requirements for engineering models.

Other work in the literature relates to similar issues. Compositional modelling [2] is an approach focusing on the composition of a model by selecting relevant model fragments from a library of reusable model fragments, and combining them to form a complete model. Their specification of model fragments typically combines the mentioned ontologies which in our approach are separated in order to enhance structure and reusability. In the compositional model fragments there are special slots that allow assumption knowledge to be specified, for example applicability conditions [7] which are essentially our validity assumptions. Our approach focuses on (adapting) complete models, instead of on selecting and assembling model fragments. Therefore, in our approach also global statements about models, like behavioural assumptions and goal statements, can be expressed explicitly in the model library. The Graphs of Models (GoM) approach [8] could be viewed as a method of modelling a certain type of assumption about models, i.e. the idealization or approximation relations between a family of model fragments. The nodes of the graph are models and the edges are the assumptions that have to be changed in going from one model to another.

Probably closest to our suggestions is the work on relevance reasoning [9]. In order to activate a model fragment, different classes of conditions must be satisfied. These classes of assumptions are represented explicitly. They comprise instantiation conditions, activation conditions and relevance conditions. The activation conditions are similar to our validity assumptions. The instantiation conditions resemble not so much one of our assumption classes but rather our component model fragment. This is due not only to the different ontological organization of our libraries, but also to the intended scope of the modelling task: in relevance reasoning and compositional modelling the task is model selection given a set of system objects or components as input. Our approach attempts to extend this scope by including construction of instantiation conditions as part of the modelling task. The relevance conditions are a group of metalevel assumptions that relate, in a heuristic way, the model fragment to the problem solving goal, which we call 'goal statements'. It is noted however that experienced modellers have a lot of metaknowledge about modelling that is reusable but relatively independent of the task or goal. For example, notions about the usefulness of linearity or knowledge about classes

of useful approximations to the current model (as in GoM) can be specified largely independently of whether the task is design, prediction or explanation. Covering this type of task-independent metalevel assumptions would therefore be a useful enhancement of current modelling support.

The categorization of assumptions that we suggest is a very tentative one. However, based on our own experience as well as modelling experiments carried out, we think that modelling by experienced modellers is to a considerable extent a form of *assumption management*. They usually do not directly deal with the content facts of a model, but rather with meta-assumptions about classes of content facts. Assumptions rather than model facts are the handles for modelling and model revision. The importance of modelling assumptions in the (re)modelling process is already shown by the very simple example in this paper. However, the specific technique used in this example (the recursive revision of modelling assumptions leading to violations leading to assumption revisions etc.) may not be appropriate in all situations because of the possibility of infinite looping. Another possible technique, adopted for example in the approach of Yost [10] to elevator design, is to apply revisions only if this does not result in extra violations. This means that recursive revisions are not possible, and infinite looping cannot occur. The relative benefits and drawbacks of each of these strategies in the context of remodelling is a topic for further research.

The problem remains of how to represent and support modelling assumptions in ways that are natural to the model engineer. Basically, this is a matter of modelling expert knowledge, a topic of central concern in the knowledge-based systems area. In recent years, knowledge acquisition has extensively addressed the question of how expert knowledge can be modelled in a structured and reusable way [5]. For example, quite a lot of thought has been given to how task/goal and domain knowledge interact and can still be specified relatively independently (see, for example, [11]). Integrating and applying these recent advances to automated engineering modelling is a step that has not yet been fully made.

Acknowledgements

The work described in this paper was partially funded by NWO/SION within project 612-322-316 (Evolutionary

Design in Knowledge-Based Systems), and it was partially supported by CEC Esprit-III project P6521 (OLMECO). The partners in the OLMECO project are PSA Peugeot Citroen (France), BIM (Belgium), Fagor (Spain), Ikerlan (Spain), Imagine (France), University of Twente (Netherlands) and the Netherlands Energy Research Foundation ECN (Netherlands).

References

- [1] A den Ouden, J.L. Top and J.M. Akkermans, EBIB — dynamic model components for a heating systems library, *Report ECN-C-94-027*, Netherlands Energy Research Foundation ECN and University of Twente, Netherlands, 1994.
- [2] B. Falkenhainer and K.D. Forbus, Compositional modeling: finding the right model for the job, *Artificial Intelligence*, 51 (1991) 95–143.
- [3] J.L. Top and J.M. Akkermans, Tasks and ontologies in engineering modelling, in B.R. Gaines and M. Musen (eds.), *Proc. 8th Int. Knowledge Acquisition Workshop KAW '94*, Vol. 2, SRDG Publications, Canada, 1994, pp. 21.1–21.24.
- [4] A. Yazman, P. Seigle, J. van Dijk, S. Stramigioli, P. Weustink, J. Top, C. Masson, FAGOR and IKERLAN, Application elementary models: development of automotive application component models: Power steering systems: development of machine tool component models, *Esprit P6521 OLMECO Deliverables 2/1 and 2/2*, OLMECO Consortium, 1994.
- [5] B.J. Wielinga, W. Van de Velde, A.T. Schreiber and J.M. Akkermans, Towards a unification of knowledge modelling approaches, in J.-M. David, J.-P. Krivine and R. Simmons (eds.), *Second-Generation Expert Systems*, Springer-Verlag, Germany, 1993, pp. 299–335.
- [6] D.C. Karnopp, D.L. Margolis and R.C. Rosenberg, *System Dynamics: A Unified Approach*, John Wiley, USA, 1990, 2nd rev. edn.
- [7] T.R. Gruber, Model formulation as a problem-solving task: computer-assisted engineering modeling, in K.M. Ford and J.M. Bradshaw (eds.), *Knowledge Acquisition as Modeling*, John Wiley, 1993, pp. 105–127.
- [8] S. Addanki, R. Cremonini and J.S. Penberthy, Graphs of models, *Artificial Intelligence*, 51 (1991) 145–177.
- [9] A.Y. Levy, Y. Iwasaki and H. Motoda, Relevance reasoning to guide compositional modelling, in R. Leitch (ed.), *Working Notes QR '92 Workshop*, Heriot-Watt University, UK, 1992.
- [10] G. Yost, Configuring elevator systems, *Technical Report*, 1992 (Digital Equipment Corporation, 111 Locke Drive (LMO2/K11), Marlboro, MA 02172, USA).
- [11] H. Akkermans, B. Wielinga and G. Schreiber, Steps in constructing problem solving methods, in N. Aussenac et al. (ed.), *Proc. European Knowledge Acquisition Workshop EKAW '93*, Springer-Verlag, Germany, 1993, pp. 45–65.