Modelling domain knowledge for intelligent simulation learning environments

Citation for published version (APA):

Joolingen, van, W. R., & Jong, de, A. J. M. (1992). Modelling domain knowledge for intelligent simulation learning environments. *Computers and Education*, 18, 29-37. [https://doi.org/10.1016/0360-1315\(92\)90033-2](https://doi.org/10.1016/0360-1315(92)90033-2)

DOI:

[10.1016/0360-1315\(92\)90033-2](https://doi.org/10.1016/0360-1315(92)90033-2)

Document status and date:

Published: 01/01/1992

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

MODELLING DOMAIN KNOWLEDGE FOR INTELLIGENT SIMULATION LEARNING ENVIRONMENTS*

WOUTER R. VAN JOOLINGEN and TON DE JONG

Eindhoven University of Technology, Department of Philosophy and Social Sciences, P.O. Box 513,
NL-5600 MB Eindhoven, The Netherlands

Abstract—Computer simulations are an often applied and promising form of CAL. A main characteristic of computer simulations is that the domain knowledge is represented in a *model*. This model contains all necessary information to calculate the behaviour of the simulation in terms of variables and parameters and a set of *rules* or *constraints* which determine the changes to the values of the variables. In order to increase the learning effects of computer simulations additional support and guidance should be offered to the learner. This means that simulations should be embedded into a supportive environment, which we will call an Intelligent Simulation Learning Environment (ISLE). One of the basic components of the ISLE should be a formalised representation of the domain. In this paper the structure of this domain representation and its authoring will be discussed. It is argued that the simulation model is a necessary but certainly not sufficient source of information for building a domain representation for an ISLE. Besides a behavioural description as given by the simulation model (the term *runnable model* will be used) also a cognitive description of the domain is needed. This *cognitive model* forms the basis for a number of functions to be performed in an ISLE, like diagnosis, instruction and support. The current paper presents a framework which can be used to formalise the cognitive model. In particular the component of the cognitive model which contains a conceptual representation of the domain, the *conceptual model* will be discussed. An important element of the framework presented is a *relation typology* which describes the interrelationships between relations that are used for the construction of a cognitive model. This typology will be an important knowledge source for an ISLE and can support the author with constructing the conceptual model.

1. INTRODUCTION

Computer simulations are a promising application of Computer Assisted Learning because they enable *exploratory learning*. However, research has shown that this type of learning needs additional support for the learner to control and/or support the specific exploratory learning processes that are needed for successful learning. Therefore a supportive and adaptive environment should be built around a computer simulation [1]. A simulation embedded in such an instructional environment will be termed an Intelligent Simulation Learning Environment (ISLE).

The supportive environment can provide the student with additional instruction and learner tools for supporting simulation specific learning processes. To establish these facilities in a flexible and adaptive way the environment will maintain a learner model, containing information about certain attributes of the learner and about the current learner knowledge state. To enable all these functions the ISLE will need a formalised representation of the domain. This leads to a natural separation of the ISLE into four different modules: the domain representation, a learner model, an instructional strategy module and a learner interface module [2,3].

The aim of the current paper is the construction of a framework for domain representation for ISLEs as a basis for all domain related functionality of the ISLE. It will appear that the presence of a simulation model (we will use the term *runnable model*) has implications for the structure of the complete representation of the domain [4]. Therefore, the framework should take into account the simulation-specific characteristics of ISLEs (it should not be a generic framework for use for all possible types of intelligent tutoring systems).

The design of this framework is quite an ambitious task, therefore we do not claim to present an ultimate framework which is capable of describing all simulation related domains, our work

*The research reported was conducted in the project SIMULATE. SIMULATE is part of SAFE, a R&D project partially funded by the CEC under contract D1014 within the Exploratory Action of the DELTA programme. In total 17 partners are involved in the SAFE consortium of which Philips TDS is the prime contractor. The partners directly involved in SIMULATE are: Philips TDS (Fed. Rep. Germany), University of Leeds, University of Lancaster (U.K.), TIFSA (Spain), University of Amsterdam, Eindhoven University of Technology, Courseware Europe (The Netherlands).

is still under progress. The present paper is to be considered as a first step in the direction of a complete domain representation framework. We do believe that the proposed basic structures allow for a powerful extension towards such a complete domain description framework.

We will present our framework in a semi-formal form. Of course a truly formalised domain representation will be needed when our framework is to be included in an actual ISLE. However, the current status of research has not reached the state of actual implementation. Moreover the semi-formal approach used in the current paper allows for offering a conceptual view on the domain representation framework.

2. KNOWLEDGE RELATED TO COMPUTER SIMULATIONS

The first function of a domain representation for a simulation learning environment is to provide information for enabling the simulation of the real system which is the subject of instruction. The model that drives this simulation is called the *runnable model*, being the complete and runnable description of the domain as it is represented in the ISLE. This means that all important aspects of the domain are modelled in the runnable model, explicitly or implicitly. The runnable model is defined in terms of a *state* subject to change, due to a set of *state transition rules* or *constraints* [4,5]. The term “runnable” means that this description is unambiguous, in the sense that for each state, described by a set of variables, the transition to another state can be calculated in a unique way.

The completeness of the runnable model does not imply completeness in the sense that all knowledge required to allow teaching of the domain is available in the runnable model. There are two main reasons why this is the case:

- Much knowledge, contained in the runnable model, is only present implicitly, therefore unsuitable for direct use in teaching. To enable teaching about the domain this knowledge should be made explicit. Also the variables and parameters in the runnable model are meaningless entities, usually just numbers. This is illustrated by the fact that very often one runnable model can be used to represent a number of different domains. For teaching purposes a meaning needs to be assigned to these entities.
- Reasoning about a domain often takes place at a higher level than that of the runnable model by introducing higher order concepts, which are not represented in the runnable model.

These two reasons form the rationale for the introduction of a *conceptual reasoning layer* on top of the runnable model. This layer will have to enable reasoning at a higher, often qualitative, level than the runnable model.

Hartog[6] even argues that, considering the fact that reasoning about a model often takes place at a qualitative level, also the simulation itself should be qualitative, in other words that the simulation should be driven by the same model that the learner is to acquire as a mental model of the system, which should imply that the simulation should be performed qualitatively. We would argue that this is neither necessary nor desirable. The simulation itself should be performed efficiently and fast, something that is not always possible using qualitative simulation. Also, the results of qualitative simulation are not always accurate enough to allow for a faithful representation of the simulated system[7]. We do acknowledge a conceptual, often qualitative, representation of the simulation model should be present in an additional knowledge base. This statement does not mean that we want to exclude qualitative simulation for the runnable model in all cases.

Apparently there is a need for a supplemental domain representation besides the runnable model in an ISLE. Moreover, we argue that this domain representation, and not the runnable model, will be the *main* source of domain related information within the learning environment. This is true because the interaction of learners with the ISLE will be of a complex nature and will require monitoring and reasoning at a high conceptual level, a level which will not be supported by the runnable model. The different ISLE tasks that the domain representation should enable are the following:

Reasoning about the domain. Any intelligent learning environment should be able to model a domain expert. This functionality is needed in order to allow the environment to derive new

relations about the domain when needed and, specifically for simulations, to draw conclusions from events in the runnable model.

Providing alternative conceptualisations of the domain. Often, in simulation-based learning, there can be several *views* on the same domain. Looking at a domain from different viewpoints can be useful in obtaining a complete conceptual representation of a domain. Also different views may be used to introduce different levels of complexity. One may start with a view which is in fact a simplification of the domain and move via one or more steps to a view which contains all important concepts of a domain. This would be a form of progressive implementation, which is a certain instructional strategy. White and Frederiksen[8,9] have implemented a learning environment in which this strategy is used.

Giving instruction about the domain. In order to enable instruction about the domain the domain knowledge base should contain instruction specific information. This would include information on how to manage different views and information on which concepts are central for learning and how concepts are dependent of each other. Also means for providing *examples* and information on *critical situations*, should be included.

Providing diagnosis. To enable intelligent instruction about the domain the learning environment should monitor the achievements of the learner and diagnose possible misconceptions. Therefore, a model of the learner's mental model about the simulation should be expressed within the same framework as the domain representation itself. Moreover, the domain representation should contain a description of possible and/or common misconceptions in order to allow for appropriate action if such a misconception should be diagnosed.

Guiding the interaction with the model. The interaction with the model takes place by varying variables and parameters. In general it will not be possible to vary all parameters and variables at the same time. Which variables can be varied depends on the *role* the learner plays at a certain moment. In van Joolingen and de Jong[4] the concept of *scenario* has been introduced as being the description of the possible interaction that can take place at certain moments. This scenario should be part of the domain representation.

The conceptual layer enabling all these functions will be called the *cognitive model*. It will include possibly overlapping submodels, each dedicated to one of the domain related functions mentioned above. The cognitive model as a whole will provide an expert teacher's view on the domain. The remainder of this paper will investigate the properties of these submodels, and of one of these in particular.

3. FORMALISATION OF DOMAIN KNOWLEDGE

3.1. Why formalisation?

Formalisation of knowledge is to describe that knowledge in a way leaving no room for interpretation. A formal description of some piece of knowledge should be done in a language with well-defined syntax and semantics. An example of such a formal language is predicate logic. Also a programming language is a formal language in this sense.

It is clear that eventually, when an ISLE is actually implemented, its functionality must be described in such a formal language, together with the information that should be contained in the knowledge base which serves the ISLE, but also earlier in the design stage a formal description of ISLE related information is necessary to enable the final implementation in a consistent way. Devising such a formal language is a major effort, which has not been completed at the current state of development. In this paper we will take the view on formalisation adopted by de Jong *et al.*[10], who strive at a semi-formal structural language for describing ISLE related information.

3.2. A model based approach

For the organisation of the knowledge base serving the ISLE, containing all relevant knowledge the ISLE needs, one may search for suitable decompositions of this knowledge base. For our purpose the knowledge base may be viewed as consisting of a set of, possibly overlapping *models*, for the domain, learner, instruction and learner interface, partly reflecting the traditional design of intelligent tutoring systems, using domain, learner, instruction and interface *modules* [2]. We add an extra flavour to this traditional design by showing the interdependencies between models and

Fig. 1. Structuring of an ISLE knowledge base. The domain knowledge in the conceptual and operational models serves as a basis for the description of ISLE-related knowledge.

allowing overlap. This is illustrated in Fig. 1, there neglecting the overlap for reasons of clarity. In de Jong *et al.*[10] this model based approach, based on the KADS knowledge base development methodology[11], is elaborated further.

The basis of the ISLE knowledge structure models is formed by the *conceptual* model and the *operational* model. The operational model describes *tasks* that are associated with the domain, whereas the conceptual model is the knowledge base reflecting non-operational knowledge of the domain. Both static and dynamic properties of the domain should be included in the conceptual model.

The operational and conceptual models are *objects of instruction* for the ISLE[10]. We need additional models which enable us to use and teach this knowledge: the *instructional* model, the *learner* model and the *interface* model. These three dedicated models draw upon knowledge stored in the conceptual and operational models.

Every model will be constructed using *elements*, *relations* and *structures* as primitives. Knowledge is stored as relations between elements, organised into structures, defining logical chunks of knowledge. Cross model communication is furnished by allowing the various models to use structures of other models as elements. Especially, structures of the conceptual and operational model will serve as elements in the learner, instructional and interface models.

The model based approach described above will be reflected in the cognitive model. We can now define the cognitive model as consisting of those parts of the knowledge base models that contain domain-related elements, where notice should be taken that “elements” of the instructional, learner or interface models may be elements, relations or structures in the conceptual and operational models.

It will be clear that the conceptual and operational models will belong completely to the cognitive model, whereas the learner, instructional and interface model will partly belong to the cognitive model. For the part of the latter three models that are inside the cognitive model we will use new terms: the *diagnosis* model, the *instruction* model and the *interaction* model (see Fig. 2). These names

Fig. 2. The three domain-related (sub-)models in the ISLE knowledge base. This figure can be regarded to be a top-view of Fig. 1.

reflect the functions the three domain related (sub-)models should enable: diagnosing the learner's domain knowledge, giving instruction about the domain and guiding the interaction with the model.

The remainder of this paper will concentrate on the structural description of the *conceptual* model as an example of the structures used in our framework. The basic elements, relations and structures of this model will be defined and examples of ISLE functionality that is related to the conceptual model will be presented. In the current paper the instruction, diagnosis and interaction models will not be treated systematically. A more elaborate discussion of the complete cognitive model is provided in van Joolingen and de Jong[12].

4. THE CONCEPTUAL DOMAIN MODEL

4.1. Basic elements, relations and structures

In constructing a framework for a domain knowledge base for intelligent simulation learning environments, one should emphasise the nature of the simulation, reflected in the presence of a runnable model. This emphasis is necessary because of two reasons:

- The runnable model is a central source of information for both the student and the reasoning module in the ISLE. Therefore the structure of the domain representation should be such that fluent incorporation of the results of the simulation into the knowledge base is furnished, meaning that important aspects of the runnable model should be reflected in the conceptual model.
- For the author, developing a conceptual model, the structure of the runnable model itself is a source of knowledge. Consequently, the authoring process can be simplified by offering primitives that are similar to those that are used to define the runnable model.

On the basis of these two reasons a choice has been made for the primitives to use for the conceptual model description. Thereby, the runnable model is considered to be a collection or database of states, described by variables and parameters. State transition rules are the generators of this database, and need no further elaboration at the conceptual level.

The conceptual model (cf. Norman[13]) should give a description of the simulation model that emphasises the important features of the model, using only relevant information, while neglecting relatively insignificant model properties. The conceptual model will contain only the information to enable this kind of high level reasoning. This means that not all information contained in the runnable model will be represented at the conceptual level, which may restrict the scope of the conceptual level reasoning functions.

On the other hand the conceptual model may introduce new concepts, not represented at the runnable level. These new concepts may be used to state new relations which are, in principle, derivable from the runnable model. Therefore these relations will be called *implicit relations*. In our conceptual modelling framework many of these implicit relations will be stated explicitly in the conceptual model, because deriving them may be very hard or impossible for the ISLE. However, many implicit relations may also be derived at run time, using a built-in reasoning mechanism.

In the conceptual model the domain characteristics will be expressed in a more natural way than in the runnable model. As a consequence a description at the conceptual level will also (together with a theory on learning with simulations) give information on possible difficulties the student may have while learning the model.

4.1.1. Elements. The basic elements that will be used in the conceptual model framework are extensions of the basic elements of the runnable model: variables and parameters. In the conceptual model we will use the same names for the corresponding elements. When there is danger of confusion the term "conceptual variable" will be used, in contrast to "runnable variable". In the runnable model these elements just represent a value of some system characteristic, quantitatively or qualitatively. The conceptual counterparts of the runnable variables and parameters extend this meaning, the conceptual variables add an interpretation to the runnable model, often in qualitative terms.

The characteristics of a variable or parameter are largely determined by the relations it has with other variables. Not only the value of a variable is of importance but especially the position it has in the relational network of variables.

A variable or parameter is characterised by a range of possible values, by the fact that it may be dependent on time, which is an important characteristic if the runnable model is *dynamic*. An important characteristic of variables is that they can exist in an hierarchical structure. In the semi-formal conceptual model framework this is expressed by allowing *inheritance* between variables. A child variable does not only inherit the characteristics of its parent but also its position in the relational variable network. This enables the introduction of various levels of reasoning. Relations between variables high in the inheritance tree will be very global, but are representative for all lower variables, which in their turn enable more precise reasoning.

In a frame representation (see Brachman and Schmolze[14]), very much in the style of the data description language designed for KADS[15] a variable can be represented by:

```
variable <name> [inherit <varframe name>]
range <range>
time-dependency <y/n>
initial value <value>
```

4.1.2. Relations. The variables and parameters are still quite useless unless one can define relations between them. For our framework for conceptual modelling there was a need to extend the “traditional” relation concept (e.g. Cerri *et al.*[16]), because traditional relations are not suitable to express fuzzy knowledge like: “If A increases B also increases”. Such a statement is not a relation between A and B but merely a statement *about* the relation between A and B without further specifying that relation. To cope with this kind of statements we introduce the concept of *generic relation* which defines a subset on the set of all relations that are possible between two variables. The subset contains all relations that satisfy a certain characteristic. Stating that a generic relation “holds” between two variables is equivalent to specifying a property of the relation that holds.

Relations can be expressed using a similar frame representation as for variables.

Generic relations allow the definition of a *relation typology*, a hierarchically organised collection of relations and generic relations. The major reasons for introducing this typology are:

- An author creating a domain representation can be supported by the typology in finding the relations s/he needs for expressing domain knowledge.
- The ISLE could use a relation typology to derive new domain relations from ones that are explicitly present in the conceptual model. This can be used for run-time reasoning about the domain.

We think of the relation typology as one of the most powerful elements of the conceptual modelling framework.

4.1.3. Structures. The relations and generic relations that are introduced in the previous section are still quite general. For example they are not typed, they can describe relationships between any kind of variable as long as the range of the variables is suitable. There is a need to combine the variables and relationships into structures, *models*, as we will call them. A *simple model* is a collection of variables, parameters and definitions of (generic) relations between them.

A model is a representation of a chunk of domain knowledge. It represents the characteristic of a physical or logical subsystem in the form of (generic) relations between variables. In that sense a model is a kind of inference rule or set of inference rules: it can be used to generate knowledge about variables on the basis of knowledge about other ones.

An important characteristic of the model structures is that they can be combined into *compound models*, which have all “outside” characteristics of a simple model. This means that they can be combined into even larger knowledge structures. This implies that the complete domain representation will have a granular structure, it will consist of a collection of interconnected models, each again decomposable into smaller units until the level of simple models has been reached.

In Fig. 3 the structure of the conceptual model framework is summarised. In this figure the main elements and features are drawn: the variable hierarchy, the relation typology, the model structure

Fig. 3. Overview of the primitives used for representing the conceptual model.

(in which variables are represented both by lines and boxes) and the combination of models into compound models.

4.2. Functionality derived from the conceptual model

The conceptual model can only be useful when the ISLE is able to actually use the knowledge stored in it. This section will discuss a few examples of such related ISLE functionality.

A first function that can be directly based on the conceptual model is *generating predictions* of experimental results and, more important, motivating these predictions. This can be provided by using the relations in the model structures as inference rules. A prediction will describe all possible behaviours obtained from earlier experiments with the simulation.

Furthermore the conceptual model framework can be used to derive new relations between variables on the basis of the relation typology. The relation typology will contain relations between relations (meta-relations) to make this possible. The new relations can be used to derive extra information about variables, or to reason about them at a different level, e.g. a qualitative level.

A last example is the matching of learner *hypotheses* against the conceptual model. One can formalise a hypothesis about the model as a statement that a certain (generic) relation holds between two variables[17]. The hypothesis and the model match if the hypothesized relation can be derived from the conceptual model, the rules from the relation typology that have been used to derive the relation give extra information about the hypothesis, e.g. if it is more or less general than corresponding relations in the conceptual model. The information obtained from this matching can be incorporated in the learner model.

4.3. An example

We conclude this section by giving a small example. Suppose we want to model a physical system consisting of a number of particles. A runnable model which will be able to calculate the velocity and momentum and possibly the kinetic and potential energy of the particles can be defined for this system using the laws of physics.

On the conceptual level, new variables can be defined representing more abstract concepts used to reason about the simulation, e.g. a variable “energy”. This variable can be used to express, say, the law of conservation of energy, using relations in the relation typology, meaning: the sum of the partial energies is a constant, when the system is closed. (The fact that a system is “closed” also needs to be expressed using variables and relations.) We may also create different *types* of energy, kinetic and potential energy, which will be subtypes of the energy variable. This is depicted in Fig. 4.

The conceptual model contains rules for reasoning about the model, e.g. predicting the future energy of particles or subsystems. This reasoning should use the information generated by the runnable model. Therefore at least some of the variables need to be defined in terms of runnable model variables. Establishing the link between runnable and conceptual model is called

Fig. 4. An example model and instantiation.

instantiation. At which level this instantiation should take place is to be decided by the author of the domain representation. If the conceptual model should address properties of individual particles, these properties should be represented at the conceptual level. If the conceptual level reasoning only deals with more global concepts, there is only need for variables representing these, in one way or another defined in terms of runnable model variables. In the case of many particle systems this may even imply that the conceptual model only contains statistical relations between the newly defined variables, because not all deterministic information is used in the definition of these variables. In Fig. 4 the instantiation process is illustrated.

5. DISCUSSION

The framework for building a domain representation as described in this paper promises to be an effective way of describing knowledge related to computer simulations. Until this moment there are no actual implementations of ISLEs which use this framework, but as a theoretical object it can already show its value. For an experiment with a computer simulation about error analysis in chemistry, we used the framework for the analysis of learner hypothesis and experimental plans[17]. All inferences about the model were done by hand, but it would be little effort to formalise the assessment scheme that was used and to implement it.

Compared to traditional knowledge representation techniques it can be remarked that the framework offers some extras by introducing the relation typology and that it offers strong typing and classifications of variables. Both extras are introduced quite naturally and can support an author building a knowledge base and serve as a basis for intelligent reasoning about the domain.

An important feature of the presented framework is that it is *declarative*, meaning that the knowledge is stored independently from the context in which it is used. This implies that it can be used for several purposes, as indicated in Section 4.2.

The usefulness of the framework depends on possibilities to build interpreters for the proposed domain representation language. Since not even all aspects have been elaborated in a formal way it is too early to state that this will be possible, but the contemporary presence of qualitative simulation systems like QSIM [7,8] and the developments in the area of model based reasoning (e.g. [19]), which employ techniques that could be applied in the proposed framework, make it comprehensible that such an interpreter can be built. At the moment we are working on a prototype of such an interpreter.

A final feature of the conceptual model is its granular structure. This structure can be extended to the complete knowledge base of the ISLE, also containing domain independent knowledge. This would reduce the complexity of the authoring task, which may become a process of selecting, from dedicated libraries, tailoring and combining the structures, also called *basic building blocks*, into a knowledge base which the ISLE can use for generating support and guidance for the learner[10].

Acknowledgements—The authors would like to thank the partners in the SIMULATE project who have contributed directly or indirectly to the ideas expressed in this paper and also the participants of the session at CAL 91 for their input in the discussion.

REFERENCES

1. de Jong T. (Ed.), Computer simulations in an instructional context. *Education & Computing*, 6. Elsevier Science, Amsterdam (1991).
2. Duchastel P., Models for AI in education and training. In *Artificial Intelligence Tools in Education* (Edited by Ercoli P. and Lewis R.), pp. 17–29. Elsevier, Amsterdam (1988).
3. Wenger E., *Artificial Intelligence and Tutoring Systems*. Kaufmann, Los Altos (1987).
4. van Joolingen W. and de Jong T., Characteristics of simulations for instructional settings. *Educ. Comput.* **6**, 241–262 (1991).
5. Zeigler B. P., *Theory of Modelling and Simulation*. Wiley, New York (1976).
6. Hartog R., Qualitative simulations and knowledge representation for intelligent tutoring. *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL* (Edited by Maurer H.), pp. 193–213. Springer, Berlin (1989).
7. Kuipers B., Qualitative simulation. *Artif. Intell.* **29**, 289–338 (1986).
8. White B. Y. and Frederiksen J. R., Progressions of qualitative models as a foundation for intelligent learning environments. Technical Report 6277, Bolt Beranek and Newman Laboratories, Cambridge, Mass. (1986).
9. White B. Y. and Frederiksen J. R., Qualitative models and intelligent learning environments. In *Artificial Intelligence and Education* (Edited by Lawler W. and Yazdani M.), Vol. 1, pp. 281–305. Ablex, Norwood, N.J. (1987).
10. de Jong T., Tait K. and van Joolingen W. R., Authoring for intelligent simulation based instruction; a model based approach. Paper presented at the *DELTA and Beyond Conference*, The Hague, The Netherlands (1990).
11. Breuker J. A. and Wielinga B. J., KADS: structured knowledge acquisition for expert systems. *Proceedings of the Fifth International Workshop on Expert Systems and their Applications*, Avignon, France (1985).
12. van Joolingen W. R. and de Jong T., An instruction related domain representation for simulations. Toward formalising the components of an intelligent simulation learning environment. Report SIM/19-20. Final report on formalisation. Eindhoven University of Technology, Department of Philosophy and Social Sciences, Eindhoven (1991).
13. Norman D. A., Some observations on mental models. In *Mental Models* (Edited by Gentner D. and Stevens A.). Erlbaum, Hillsdale, N.J. (1983).
14. Brachman R. J. and Schmolze J. G., An overview of the KL-ONE knowledge representation system. *Cogn. Sci.* **7**, 171–216 (1985).
15. Schreiber G., Wielinga B., Hesketh P. and Lewis A., A KADS design description language. Deliverable B7, ESPRIT project P1098. Amsterdam UvA, Department of Social Science Informatics (1990).
16. Ceri S., Gottlob G. and Tanca L., *Logic Programming and Databases*. Springer, Berlin (1990).
17. van Joolingen W. R. and de Jong T., An hypothesis scratchpad as a supportive instrument in simulation learning environments. OCTO Report 91/103. OCTO, Eindhoven University of Technology, Eindhoven (1991).
18. Fishwick P. A., A study of terminology and issues in qualitative simulation. *Simulation* **52**, 5–9 (1989).
19. Leitch R. R., Wiegand M. E. and Queck H. C., Coping with complexity in physical system modelling. *AI Commun.* **3** (1990).