

KMsim: A Meta-Modelling Approach and Environment for Creating Process-Oriented Knowledge Management Simulations

Anjo Anjewierden, Irina Shostak, Robert De Hoog

► **To cite this version:**

Anjo Anjewierden, Irina Shostak, Robert De Hoog. KMsim: A Meta-Modelling Approach and Environment for Creating Process-Oriented Knowledge Management Simulations. 13th European Conference on Knowledge Acquisition, Management and Modelling (EKAW), 2002, Sigüenza, Spain. Springer, 15 p., 2002. <hal-00190694>

HAL Id: hal-00190694

<https://telearn.archives-ouvertes.fr/hal-00190694>

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

KMsim: A Meta-Modelling Approach and Environment for Creating Process-Oriented Knowledge Management Simulations

Anjo Anjewierden¹, Irina Shostak², and Robert de Hoog^{1,2}

¹ Social Science Informatics, University of Amsterdam,

Roetersstraat 15, 1018 WB Amsterdam, The Netherlands, anjo@swi.psy.uva.nl

² Faculty of Educational Science and Technology, University of Twente,
PO Box 217, 7500 AE Enschede, The Netherlands, {shostak,hoog}@edte.utwente.nl

Abstract. This paper presents a new approach to modelling process-oriented knowledge management (KM) and describes a simulation environment (called KMSIM) that embodies the approach. Since the beginning of modelling researchers have been looking for better and novel ways to model systems and to use appropriate software to create simulations. The application of the approach and KMSIM make it possible to create realistic business models (BMs) and simulate the consequences of KM interventions and events. The validity of the approach and tools is being evaluated in the game KM Quest.

1 Introduction

With the ever growing interest for knowledge management, it is unavoidable that the demand for a more formal approach increases in parallel. After the first flush of ideas, whose main function it was to create awareness, more precise and hands-on methods are called for (see for example [9]). This holds in particular for models that show how knowledge and knowledge processes can influence organisational effectiveness (see [5], [4] and [8]). This “show how” becomes even more valuable when these influences can be simulated in a business model (BM), as this is the only way one can capture and understand the dynamics of knowledge. The need for modelling and simulating knowledge management relevant business models raises the question whether additional tools are required beyond the standard simulation environments already available.

This paper describes KMSIM, a set of tools which have been specifically designed to support creating and simulating knowledge management relevant business models. It is argued that the need for these tools can be derived from the nature of knowledge management as a discipline, the peculiar properties of knowledge relevant business models and the intended users of the tools. The tools were developed in the context of the KITS project. The goal of this project is to develop a comprehensive game-based collaborative learning environment for knowledge management called KM Quest [1]. An essential part of this environment is a knowledge management relevant business model that simulates the behaviour of a (fictitious) company. This model has been developed and partly validated with the tools described in this paper.

The paper consists of four sections. In Sect. 2 the factors driving the need for a specific and new set of tools are discussed. Sect. 3 describes the architecture of the simulation environment based on the requirements. The last two sections describe the functionality provided by the tools from the point of view of creating business models (Sect. 4) and simulating and validating these models (Sect. 5).

Acknowledgements We would like to thank the three anonymous reviewers whose comments we have tried to take into account. Work partially supported by the European Community under the Information Society Technology (IST) RTD programme, contract IST-1999-13078 (KITS). The authors are solely responsible for the content of this article. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing herein. Partners in the KITS project are University of Twente (NL), University of Amsterdam (NL), CIBIT (NL), ECLO (UK), Tecnopolis (I) and EADS (F).

2 Factors Driving the Design of the Tools

2.1 Knowledge Management has an “Object”

Knowledge management, as a branch of general management disciplines, has an “object” that is different from the “objects” that are the focus of other sub-disciplines. There are many simulation environments that allow one to model various kinds of business processes, including manufacturing, public systems, and service systems simulations (for example Powersim®). Most environments, however, do not provide for treating knowledge as a *simulated entity*. Rather, in these environments the simulated entities are the *implicit result* of applying knowledge in a specific domain. Knowledge is not considered an object on which different actions can be applied, for example to model “stocks” and “flows” of knowledge.

To illustrate this idea, consider a manufacturing simulation which allows one to get answers to questions like “How can work-in-process inventory and cycle time be reduced while increasing throughput?” or “When should the next piece of equipment be purchased and how many people are needed to work with this equipment?” In this simulation knowledge about manufacturing processes is *applied* while simulating inventory, amount of labour, time are *not taken as a simulated entity* in contrast to knowledge management simulations. “Stocks” and “flows” of domain-specific knowledge compose the area of interest of KM simulations. For KM simulations it is important to quantify, measure and model “manufacturing knowledge” as a simulated entity, which can be done by introducing variables such as level of competence in manufacturing and speed of knowledge gain in manufacturing. As an example of one of the first knowledge management simulations we can mention *Tango!* [2]. Apparently the business model underlying this simulation does not handle knowledge as a separate entity, but operates directly through employees on key performance indicators of a company. So, the nature of the object of knowledge management in terms of stocks and flows requires a set of tools that allows the modeller of the knowledge management relevant business to map the “paper” representation of this model with a minimum of effort on a simulation engine.

2.2 Nature of the Knowledge Management Relevant Business Model

The model that is introduced in this paper was developed for KM training and is applied in the game KM Quest. To support this function the business model should satisfy the following basic principles:

- A business units' output depends on the *level of knowledge* and the *level of knowledge usage* (or utilisation). The output of work would be more valuable if people in the company possess better and use/apply (more recent, novel, advanced) knowledge. However this result could be counteracted by non-effective organisation of work processes, and vice versa. The ideal situation consists of effective organisation of work processes and highly skilled, highly knowledgeable employees.
- Certain changes outside or inside a company should influence its knowledge household – individual and organisational knowledge.
- An important assumption is that knowledge can be seen as a quantifiable object and can be measured in relative terms.
- There is a natural depreciation of knowledge due to volatility, instability, and ageing of knowledge. If in a company nobody takes care of renewal, gaining, and retention of knowledge, the company will in the long run not be able to compete with market conditions.

These considerations play a fundamental role in our modelling approach. Thus, in the model knowledge “stocks” are introduced as the level of competence(s) and knowledge “flows” are introduced as the efficiency of processes involving knowledge, such as knowledge gaining, development, utilisation, transfer, and retention.

Simply stated, any event that happens outside or inside the company and any interventions taken inside the company can have an influence on the knowledge processes – knowledge flows. These also influence the “state” of knowledge in the organisation – knowledge stocks, which influence business processes and determine their quality. Finally, the business processes contribute and generate the values of the key organisational effectiveness variables like profit and market share.

Based on what has been said above, our modelling approach resulted in a four-level model consisting of:

Organisational effectiveness variables These variables reflect the competitive characteristics of the company and are represented by variables like market share, profit, level of sales and so on.

Business processes related variables These reflect the quality of internal processes and “how well” work is done within the company. Examples are: average time of bringing a new product to the market and production level.

Knowledge related variables These variables reflect the relevant knowledge domains (e.g. marketing, research, production) and represent the level of competence in each domain.

Knowledge processes related variables Reflect the properties of processes involving knowledge in the organisation (e.g., speed of knowledge gaining, effectiveness of knowledge transfer).

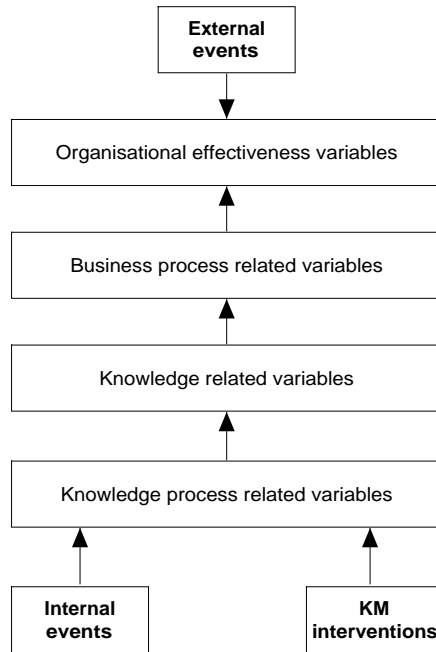


Fig. 1. Conceptual structure of the business model

The general structure of this knowledge management relevant business model is shown in Fig. 1.

Events and interventions are important components of the model. In our view, events are any changes outside or inside a company that happen independently from management of a company. Interventions are actions taken by management in order to prevent or to react to events and are aimed at improving the *knowledge household* of a company. These interventions become *knowledge management* interventions and differ from managerial interventions by its operational object(s).

However, knowledge is something that is difficult to measure in absolute terms. So, it is impossible to perform actions that (immediately) increase the “amount” of knowledge, i.e. quantitative characteristics of knowledge. On the other hand, it is relevant and possible to change the quantity of other objects such as raw materials, time, amount of labour, investments, which are the subjects of other managerial interventions. One can argue that managerial decisions concern not only these tangible, physical objects, but also include decisions about strategic development, market policy and strategies, partnership policy, and so on. Those decisions are qualitative and based on and impacting knowledge that is needed for a company to improve its value. Simpler examples include the decisions to conduct training programmes or ICT implementation. Those decisions lead to qualitative changes in the organisation and, in many cases, cannot be measured directly and more importantly they affect the *knowledge household*.

As a consequence we should treat interventions not as a quantitative, but as qualitative entities and which require a very specific implementation in the simulation.

Summarizing what was discussed above and referring to the classification of models [6], and the types of interactions that can occur between discretely changing and continuously changing state variables [7], both discrete and continuous components must be present in our simulation model, in particular the ones listed below:

- State variables change continuously with respect to time. Knowledge related variables exhibit decay behaviour and consequently influence the state of other types of variables;
- Discrete events (in our terminology - events and KM interventions) cause discrete changes in the value of continuous state variables;
- Continuous state variables achieving threshold values may cause a discrete event to occur. Threshold values of knowledge related variables could be conditioned to enable occurrence of several events. This feature of the model is relevant for the game and probably not applicable in reality, since events are unpredictable in many cases. Despite this fact, events still can be generated to consider several scenarios.

In addition we assume that the state of the business is never monitored on a permanent basis as is done both on aircraft and in many industrial processes. Usually some kind of reporting takes place at fixed points in time (monthly, quarterly, yearly). This should also be reflected in the model: it should be able to provide reports about relevant variables at pre-determined time intervals.

2.3 Practical Requirements for Tool Support

Apart from factors derived from the topic (knowledge management) and the business model, also factors reflecting the intended users of a tool are important. In general, the quality of a tool depends to a large extent on how the vocabulary of the user is made available in terms of tool functionality. The operationalisation of the functionality should be hidden from the user as much as possible.

In the KM Quest context the main concern is the need for modifying business models. As only rarely a single model can serve different purposes, one expects that people want to tailor a model to their own organisational context or even build an entirely new model. People having the knowledge to modify or build those models, usually don't have the skills needed to implement it in a simulation engine. Thus what is needed is a fairly simple and easy to learn way of creating running simulations. The technical skills we expect from users are more or less similar to the skills needed to use spreadsheets. Satisfying this requirement makes it possible that a business model can be created or modified, and simulated interactively without any technical training. Some further, more detailed, requirements are:

- Dedicated support for creating, modifying and maintaining BMs, interventions and events
- Vocabulary of the BM modeller and the tools is identical
- No limits in terms of complexity of the model and all common mathematical modelling constructs

- Automatic error detection where possible
- Extensive support for simulation, visualisation and validation

We are not aware of an existing simulation environment that is sufficiently close to what our model requires, in particular the notion of interventions and events acting on the BM variables. Using a general purpose simulation or programming environment is not an option, given the intended users.

From a historical point of view, we would like to note that in the KITS project initially extensive tool support was not deemed necessary. It soon became apparent that the nature of BMs in general and the additional complexities of connecting KM interventions to such BMs made dedicated tool support a necessity.

3 Architecture

In this section we provide an overview of the architecture of the tools. The main architectural decision is to maintain two representations of the BM. The first representation is a *specification* in terms of the meta-model. This specification can be edited and browsed by a set of model entry tools. The second representation is *operational*, and is a translation of the specification into storage, computational statements and control structures. The operational representation, called the BM engine, computes successive states of the BM and is used by the simulation and validation tools. Obviously, the translation from specification to engine is completely transparent to the user (Fig. 2).

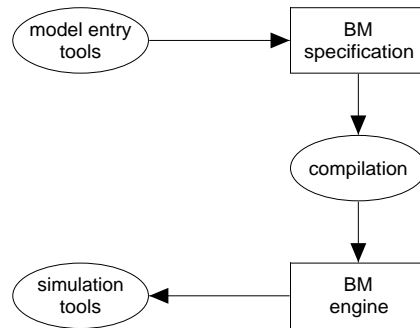


Fig. 2. Representations and tools involved in the architecture of KMSIM

The requirements are realised in detail through seven tools which are briefly described below. There is an additional tool to make the BM engine available as a server over the internet, following from a requirement of KM Quest. All tools are implemented in XPCE-Prolog [10].

Model entry (three tools) A BM can be created, modified and viewed using three model entry tools for the variables, interventions and events respectively (Sect. 4).

These tools allow the specification of the BM in terms the modeller will be familiar with: status of variables, ranges, constraints on variable values, notions of decay and depreciation, influence over time, delay and effects of interventions and events. In addition, some administrative aspects can be entered (domain within the company, precision for visualisation, description).

Charts design Charts are an important way to convey values of BM variables to the user. A high-level chart design tool supports the definition of visually attractive charts. The simulation tool automatically links values to the charts. The charts design tool is not further discussed in this paper.

Simulation Interaction between a user (model developer, validator, game player) and the BM is possible in a simulation tool (Sect. 5). The user can activate and deactivate interventions, issue events and view the effects on the BM variables as charts, numerically (HTML, XML) or as a comprehensive visualisation of all knowledge process related variables (called the *knowledge map*).

Validation and tuning support A very important aspect of a simulation environment is to provide assistance for tuning and validating the model. These tasks are supported by a tuning tool (which randomly generates events and interventions and checks whether user defined assertions are met) and a tool that traces the behaviour of the model graphically (Sect. 5.2).

Embedding A special version of the BM engine, called the BM server, can be run as a server on the internet in which it communicates using XML as input (specifying events and interventions) and outputs the BM state in XML and charts as bitmapped images. The BM server is used as part of KM Quest.

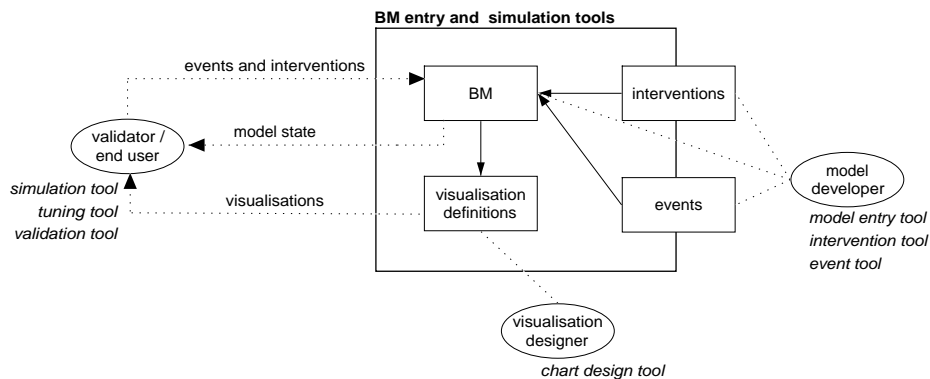


Fig. 3. Tools in KMSIM as seen from the roles of those interacting with them

Fig. 3 shows the roles of the various users involved with the tools. The *model developer* uses the model entry tools to create a BM and associate KM interventions and events with the BM. A *visualisation designer* defines how variables in the model are shown to the user. The *validator* uses the the simulation, tuning and validation tools to verify the correctness of the BM. Often the model developer and the validator will be

the same person. And finally, the *end user* interacts with the BM engine embedded in the KM Quest game.

4 Model Entry Tools

The “core” BM represents a model of a company. The variables part of the BM are related to the business process (e.g. production level, number of employees), the knowledge process (e.g. competence in marketing), and the organisational effectiveness (profit, market share). The relations between these variables are such that the organisational effectiveness of the company deteriorates (*decay*) when no attempts are made to improve the knowledge process through interventions.

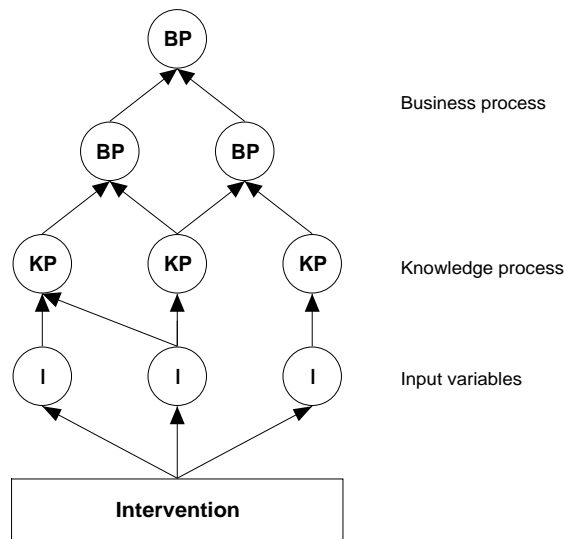


Fig. 4. Interventions (and events) influence input variables, which in turn influence the knowledge process variables and business process.

The link between the “core” BM and the knowledge management interventions and events is represented by a set of *input variables*. A “complete” BM therefore consists of the “core” BM and the input variables (see also Fig. 4). Because there are no randomised elements in the BM, it will always display the same behaviour when no interventions are implemented and no events occur.

Interventions and events are defined in terms of how they affect the input variables.³ This makes it possible to define interventions and events independently of the BM. A BM that can be simulated therefore requires: (1) a BM consisting of variables representing the business and knowledge processes, input variables that distribute the effects

³ External events can also affect the organisational effectiveness variables directly, for example when a competitor brings an innovative product to the market and thereby gains market share.

of interventions and events over knowledge process variables; (2) a set of interventions; and (3) a set of events. Different simulations can be created by replacing the interventions and/or events, without changing the BM.

4.1 Business Model Entry Tool

The BM entry tool supports the creation and modification of a business model. Each variable in the model has several attributes (see Fig. 5) defined in an ontology. Most of the concepts in this ontology are fixed, some, for example the domains in the company (marketing, research) can be changed by the user. One of the attributes is a *formula* which explicitly specifies how the value of the variable is computed and implicitly defines the influence relation between variables. An important aspect of the nature of the BM is the notion of time. This is modelled in discrete periods, which are called *cycles* in the tool.

The initial state of the model (cycle 0) is bootstrapped by computing constants. The constants represent a particular business case and can be used to “scale” the model for companies of various sizes or currencies.

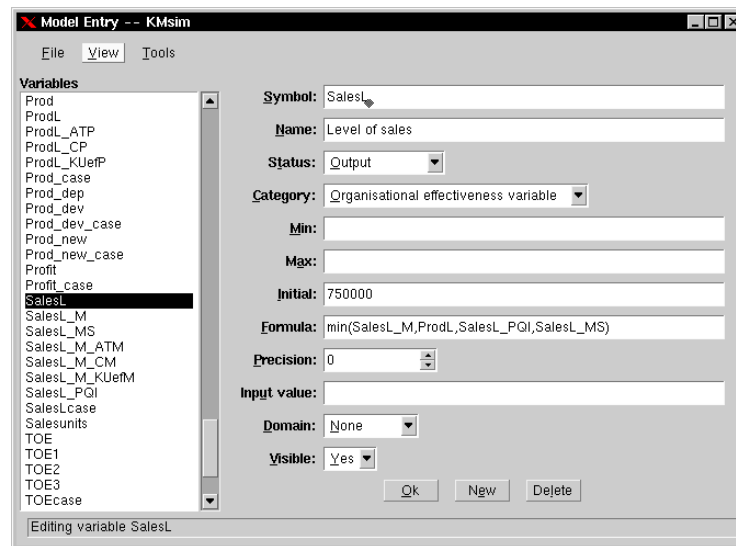


Fig. 5. Business model entry tool

A subsequent state of the model is computed by ordering the variables on their dependency on other variables. Once all dependent variables are computed, the formula associated with a variable is applied by the BM engine. Some examples of how BM notions are mapped onto formulae are given below:

Decay As explained in Sect. 2 the decay of knowledge process variables is fundamental to our meta-model. We can model the decay of knowledge utilisation with the

formula $KU = KU * C1$ (where $C1$ is a constant, e.g. $C1 = 0.94$). The BM engine translates this formula to $KU_c = KU_{c-1} * C1$, where KU_c stands for the value of KU in the current cycle and KU_{c-1} for the value in the previous cycle. Because of the propagation of values from knowledge process (via knowledge and business process variables) to organisational effectiveness, the overall performance of the company will also exhibit decay.

Propagation of influence An example of influence between variables is the formula for competence in marketing: $CM = KG + KD + KR$. The level of competence depends on knowledge gain (KG), development (KD) and retention (KR). Here, all variables are computed using the same cycle, which implies that KG, KD and KR have to be computed first and that the computation is $CM_c = KG_c + KD_c + KR_c$. A visualisation of influences and computation order can be seen in Fig. 8.

Relative change and delayed influence Relative change and delayed influence can be computed by referring to a previous cycle using the notation $V - V[-1]$ which is the difference between the value during the current and the previous cycle (computationally $V_c - V_{c-1}$).

Constrained values Values can be constrained by other values. For example, the sales level is constrained by the production level and sales level based on market share (see example in Fig. 5).

Depreciation Depreciation is an extreme case as it specifies decay in the *future*. For example, patents expire after some time. In the BM, depreciation is specified through a special function and the BM engine automatically subtracts the values for future cycles.

Scaling and natural constraints The knowledge process variables are scaled to lie between 1 and 10, this is specified in the *min* and *max* attributes of a variable. Similarly, market share has the natural constraint to lie between 0 and 100%.

Formulae may contain all common mathematical, conditional, comparison and logical operators.

The formulae are “hypotheses” about the relative dependencies that have to be tested by simulation, they are partly based on ideas from the literature. Defining the formulae is obviously part of the knowledge acquisition problem the tools support.

4.2 Intervention and Event Entry Tools

Interventions (Fig. 6) and events consist of a control and a computational part. The control part states whether the intervention or event is possible and the computational part states which input variables (see Fig. 4) are affected. Events are slightly more complicated than interventions as they may depend on the current state of the model, whereas interventions do not. For example, the event “intranet breaks down” requires that the model is in a state in which the intranet is installed (through an intervention). Events therefore may have enabling and disabling conditions. If these conditions are not specified, the event can always occur. Otherwise interventions and events are specified in precisely the same manner and we will only consider interventions in this section.

The control and computational aspects of interventions are:

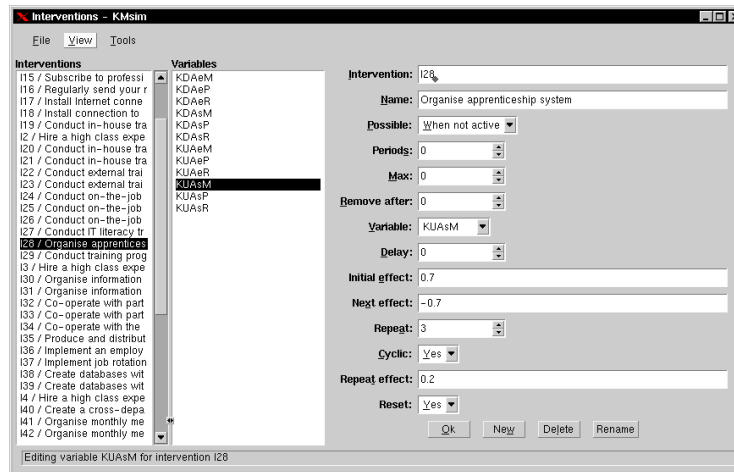


Fig. 6. Intervention entry tool

Control aspects The control aspects deal with the possibility, frequency and duration of an intervention. An intervention can be unavailable because it has already been implemented (e.g. installing an intranet). An intervention can be implemented a limited number of times (*Max*), there has to be some time between subsequent implementations (*Periods*) or the intervention is automatically removed after a certain number of cycles (*Remove after*).

Computational aspects For each input variable (middle browser in Fig. 6) affected by an intervention, the effect has to be specified. Unfortunately, a simple formula does not suffice here, as the effect may be distributed over time in complicated ways. For example, installing an intranet has immediate effect on expenses (i.e. buying equipment), but there are additional expenses periodically (i.e. hiring staff to maintain the intranet). The effect of interventions (and events) is specified using the following vocabulary:

Delay Many of the KM interventions do not take immediate effect, there is a *delay* of some cycles.

Initial effect The initial effect of an intervention is usually positive (e.g. knowledge development increases).

Next effect But, this effect disappears completely or partially.

Repeat and repeat effect Effects can repeat every so many cycles (e.g. paying for subscriptions).

The BM engine treats the effects of interventions similarly to depreciation. When an intervention is implemented the future changes to the input variables are computed and these values are used when computing subsequent states of the model. These future effects are not applied when an intervention is undone.

It may now also be apparent why existing simulation environments could not be used. The value of a variable depends on its formula, the cumulative effects of interventions and events, and depreciations.

5 Simulation and Validation

In the previous section we have described how to create a BM in KMSIM and how the BM engine computes subsequent states. Crunching out the numbers is, however, only part of the simulation.

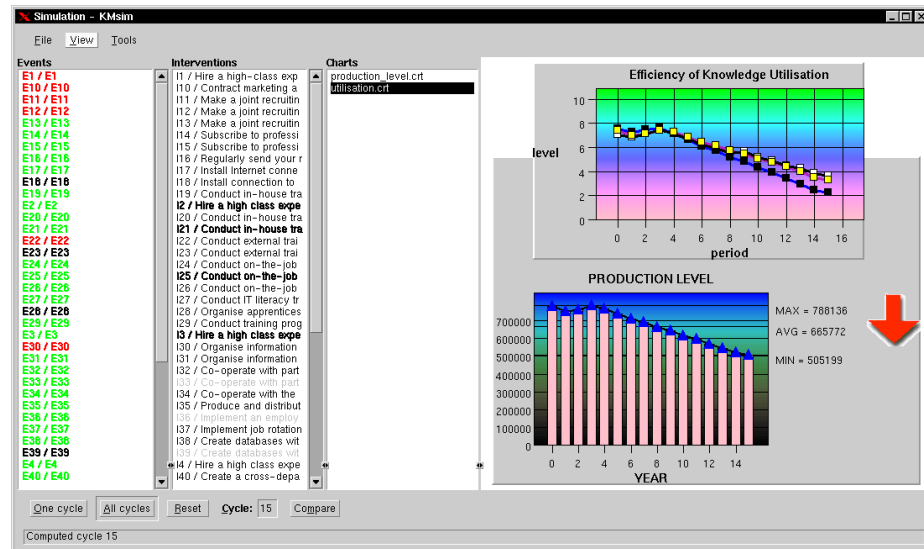


Fig. 7. Simulation tool. The two clickable browsers on the left display the events and interventions. Colour coding is used to indicate whether they are possible or active. The results are shown on the right. At the bottom are controls to run a simulation.

5.1 Simulation

There are often several reasons for simulating a BM. The BM modeller needs simulation to study the effects of interventions and events on the behaviour of the model. The main motivation for simulation for the BM modeller is to validate and tune the model.

In KM Quest the learner is only provided with a partial picture. The learner can ask for the past values of *output variables* (organisational effectiveness and business process) which are displayed using charts defined by the chart design tool. A complete simulation involves a little more. The state of the model at any point in time includes the values of the BM variables and the status of interventions and events.

Fig. 7 shows the interactive simulation tool. The status of interventions and events (active, possible) is displayed using a colour coding scheme. When validating the model developer uses the simulation tool to implement interventions and to issue events. The tool can visualise the BM variables in various ways: they can be shown in charts, as HTML tables (for later reference) or as a so called *knowledge map*.

5.2 Validation and Tuning

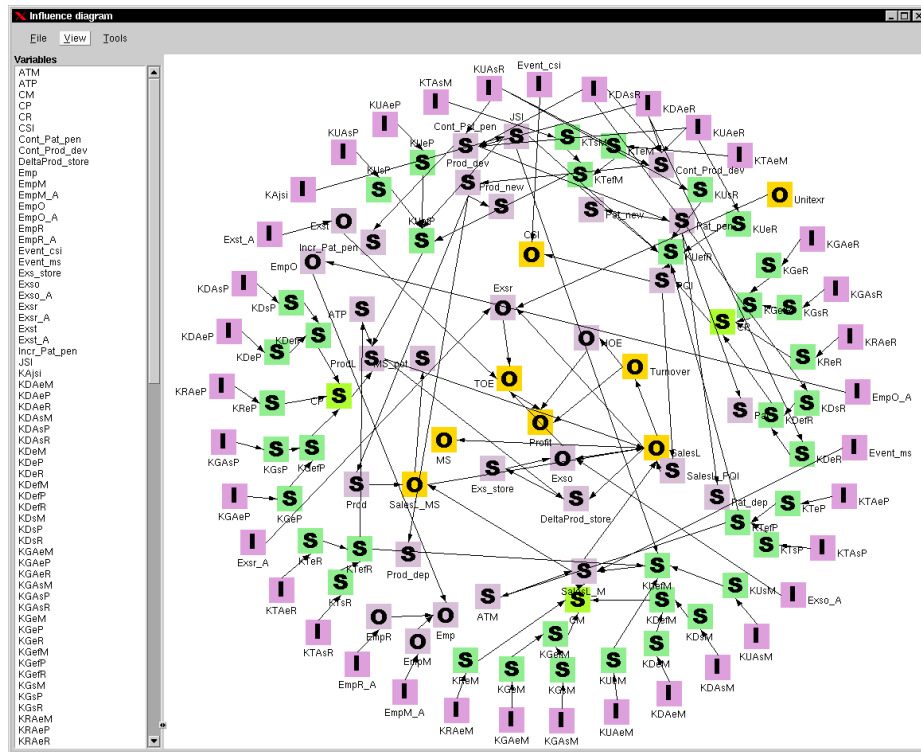


Fig. 8. Influence graph. Vertices represent variables (I=input, S=state, O=output) and edges represent influence. Colour is used to indicate whether the variable reflects organisational effectiveness, business process, knowledge process or knowledge.

Visualising the BM itself is mainly useful for the model developer. The most obvious visualisation is a graph that displays the influence relations between variables (Fig. 8). For compactness, the graph is displayed as a sphere where influence extends inwards. Algorithms to draw such graphs dynamically can be found in [3]. The vertices in the graph are colour coded and indicate the status of the variable. The visualisation makes the organisation of the meta-model clear. The outer ring contains the input variables (which are influenced by interventions and events), the second ring mainly contains knowledge variables and the inner rings contain the business process and organisational effectiveness variables (for some reason *Profit* is in the centre). The graph has turned out to be a very powerful tool for finding “obvious” errors in the model.

Fig. 9 shows a simulation of an intervention; note the use of KMSIM’s visualisation features. The vertices again represent variables in the BM. On the left are the input variables affected by the selected intervention and all vertices are decorated with a

symbol indicating the value relative to *not* implementing the intervention. For example, *Profit* (on the very right) is lower (▼) as a result of the intervention, whereas most other variables are higher (▲) or the same (●).

Although the validation tools are aimed at validating and tuning the BM, they also turned out to be of practical value for finding errors in the implementation of the BM engine.

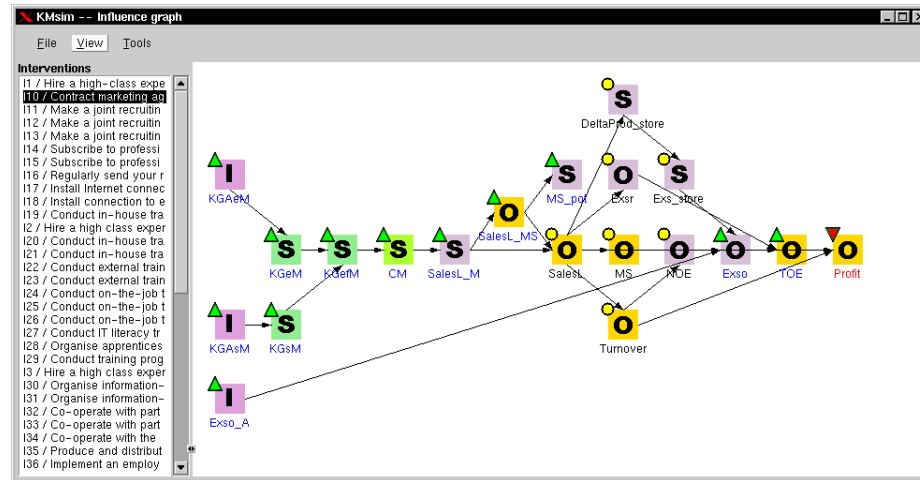


Fig. 9. Validating an intervention

6 Conclusions

The designer of tools is caught between the devil and the deep blue sea when faced with the choice between generality and specificity. Making a tool very general increases its applicability but decreases its support for the user because it will contain less “content” about the application domain. Making it very specific decreases its applicability because it can only be used in a well defined limited context but increases its support for the user because it will contain more “content” about the application domain.

The natural tendency is to go for generality: this will appeal to a larger market. As a consequence many simulation support tools are not too far removed from ordinary “visual” programming tools (e.g. Powersim®⁴), which makes them still difficult to use for domain experts without any programming experience. From a knowledge acquisition and knowledge creation perspective these domain experts are the people who really matter. In domains where acquiring and creating knowledge by means of systematic empirical investigations is either very time consuming, hard or dangerous, simulation

⁴ It should be mentioned that this environment also has a kind of “meta-model”: system dynamics.

is the preferred way to validate theoretical models on their plausibility. So, supporting domain experts in areas where these kind of limitations apply with easy-to-use tools for building, inspecting and running simulated versions of their models, can be seen as a key area for knowledge acquisition. By necessity tools that serve this purpose will be on the “specific” side of the continuum outlined above. In our domain this specificity is derived from the nature of the domain, the nature of the models that must be build and the intended users.

The domain described in this paper, knowledge management, and the tools developed are a clear demonstration of the power of this approach. Of course, more experience with the toolset is needed. For example, the range of users should be expanded, more research has to be done concerning actual ease of use, flexibility over a wide range of different business model types will be investigated. However, the application of the tool in the KITS project has significantly speeded up the creation and validation of a critical aspect of the learning environment: the business model represents in an active way the company the learning is dealing with. At the same time, the availability of the tools will make creating versions of the business model, fitting very specific requirements, much easier and this will contribute to the commercial value of the KM Quest environment.

References

1. KMQuest simulation. See for more information www.kmquest.com.
2. Tango! simulation. See for more information www.tangonow.net.
3. G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph drawing: Algorithms for the visualization of graphs*. Prentice-Hall, Upper Sadle River, 1999.
4. T.J. Beckman. The current state of knowledge management. In J. Liebowitz, editor, *The knowledge management handbook*, pages 1.1–1.22. CRC Press, Boca Raton, 1999.
5. T.H. Davenport. Knowledge management and the broader firm: strategy, advantage and performance. In J. Liebowitz, editor, *The knowledge management handbook*, pages 2.1–2.11. CRC Press, Boca Raton, 1999.
6. A. Law and W. David Kelton. *Simulation modeling and analysis*. McGraw-Hill, Boston, 2000.
7. A. Pritsker. *Introduction to Simulation and SLAM II*. John Wiley, New York, 1995.
8. R. Reinhardt. Knowledge management: linking theory with practice. In D. Morey, M. Maybury, and B. Thuraisingham, editors, *Knowledge management: Classic and contemporary work*, pages 187–221. MIT Press, Cambridge, Mass, 2000.
9. A. Tiwana. *The knowledge management toolkit: Practical techniques for building a knowledge management system*. Prentice-Hall, Upper Sadle River, 2000.
10. J. Wielemaker and A. Anjewierden. Programming in XPCE-Prolog. Available from www.swi-prolog.org, 2002.