TITLE SHEET

Dear Dr Matta,

Please find here our revised submission for inclusion at IJCAI-99 Workshop on Knowledge Management and Organizational Memories.

Title: Simulation-Based Management of Industrial Process Knowledge

Authors:

Tariq M Khan, School of Informatics and Multimedia Technology
University of North London, N7 8DB. *t.khan@unl.ac.uk; tari@cee.hw.ac.uk*
Keith Brown, Roy Leitch, Department of Computing and Electrical Engineering,
Heriot-Watt University, Edinburgh EH14 4AS.*{keb, leitch}@cee.hw.ac.uk*

Contact author is Dr Tariq M Khan.

Yours sincerely

Tariq Khan

Simulation-Based Management of Industrial Process Knowledge

Tariq M Khan, *Keith Brown, *Roy Leitch
School of Informatics and Multimedia Technology
University of North London, N7 8DB.
*Department of Computing and Electrical Engineering,
Heriot-Watt University, Edinburgh EH14 4AS.
t.khan@unl.ac.uk;{keb, leitch}@cee.hw.ac.uk

Abstract: A knowledge management toolkit is described. The essential components are knowledge-based models, numerical simulations and monitoring systems. Ideas from the field of intelligent tutoring and decision support systems are integrated here to provide a toolkit for managing a wide range of knowledge kinds including tacit and dynamic knowledge. Several basic issues in the design of simulation-based knowledge management systems are discussed in general first and then with reference to our implementations in the process industries: MOBIT and EXTRAS.

1. INTRODUCTION

Operators of an industrial process plant engage in several different kinds of activities during a normal working day. From operating controls in tightly defined sequences according to standard operating procedures to reflecting on unfamiliar situations in discussion with co-workers. All of these activities make demands on the operator for different kinds of reasoning capabilities. Social skills for people management, technical skills for machine management, and affective, meta-cognitive and political adeptness are all signal elements of an operator's realm of expertise. Through observation, many of these skills can be described as a representation of the knowledge responsible for producing the level of skill witnessed. Results achieved from executing skills can also be described in more or less formal notations. Many elements of skill, however, cannot be described formally because they are tacit and so difficult to tease from the rush of environmental factors. Our method for including these tacit elements in the organisational memory is to use simulations to provide a non-verbal representation of knowledge. The kind of knowledge represented is normally evident during the execution of operations on the controls of an industrial plant. This process involves an operator and the system interacting. Also included is a representation of the task that the user is trying to achieve. There is no attempt to try to model the interaction; rather the aim is to provide all the support media needed for reuse of the knowledge present in the interaction, which include the user, the simulation (industrial plant), and the environment. To provide later generation of the interaction, a situation must be created that brings into play all the media involved in the original interaction, or at least functionally similar media. For instance, the user may be different in the original and re-use cases, or certain environmental factors can be altered without affecting the significance of the relevant factors. Tasks and systems must remain the same functionally: for example, an alternative goal may be achieved by following a portion of an operating procedure that was originally used to achieve a different goal; or a part of the system can be worked on although the larger system may be different from the original system.

1.1 Overview of Paper

A methodology we have developed for constructing knowledge management systems that use intelligent systems (models) and simulations is described in detail in (Khan, Brown & Leitch, 1999). Procedures are described that should be followed to achieve three tasks: (a) model the elements of knowledge that can be readily represented using knowledge representation formalisms; (b) identify tasks at both the performance level and the cognitive level; and (c) to provide real time support (explanations) during interaction in the re-use case to ensure compatibility with the original case. That work is continued here as we begin by justifying the use of simulations in knowledge management and capitalisation and then discuss in detail some fundamental issues that serve as the foundation for the methodology. A high level description of the simulation-based knowledge management toolkit is described, which illustrates the different functional components comprising a knowledge management system. Following this, some of the more novel components in the architecture are described in more detail. The environment has provision for modeling of knowledge and simulation of activities, as well as for providing decision support and supervised tailored training. The EXTRAS environment, therefore, caters for many aspects of knowledge management.

2 INTERACTIVE SYSTEMS

Whether interaction is initiated by the user or by the computer, the manner of interaction must be carefully designed to facilitate natural and cognitively economic communication through the human-computer interface. Model-based interfaces are widely employed in industrial contexts to provide dynamic graphical representations of the workings of complex physical systems. These simulations can be used in at least two major roles in the capitalisation of a corporate memory: decision support and training. The emphasis in each of these roles is different in at least one important respect, namely the intention behind the provision of information. In decision support the aim is to provide information in an appropriate format and in adequate quantity to assist the user with solving real problems. In contrast, for training the aim is to provide backup information in problem situations to encourage learning by the individual and the organisation. Training, therefore, is more demanding on the knowledge management system than decision support alone. In particular, decisions must be made about the user's ability to understand and cope with the information given. How this information should be presented, and when and how much backup information is needed are both vital parts of an information provision strategy.

Knowledge Based Systems (KBS) can be useful in both decision support and training to handle these decisions. Such tools come in many forms, for example, intelligent tutoring systems (Wenger, 1987), which is a powerful technology for supporting individualized learning. All of these technologies are cognitive tools (Norman, 1993) that ease the cognitive load on users engaged in complex activities. They allow the user to *mediate* with their task related goals rather than interact with them *directly*, Figure 1.

Although the user is free to take any path to their goals, direct or mediated, cognitive tools are intended to offer the path of least resistance, that is, greatest cognitive economy. But how are these cognitive tools to be designed? Here we offer a discussion of some important issues that must be considered in the design of simulation-based knowledge management systems. This discussion will help system developers make informed decisions about several significant factors in the design and development of simulation-based knowledge management systems.

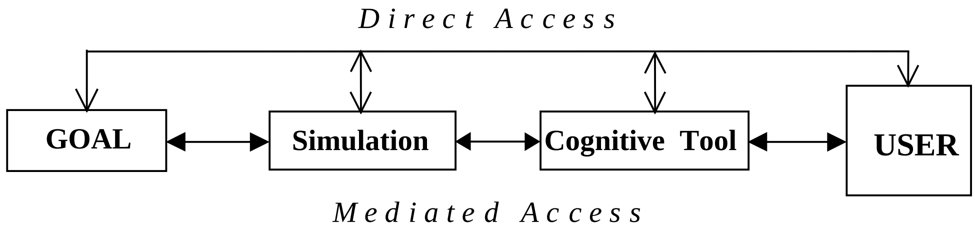

Figure 1: Two modes of access to the task goal by the user

3 SIMULATION-BASED KNOWLEDGE MANAGEMENT

Simulations and models can be used in knowledge management for several reasons: (a) provide dynamic, responsive, interactive visual information; (b) recreate simplified reality that focuses on key elements; (c) provide opportunities for exposure to hypothetical situations, cheaply, safely and quickly; (d) provide readily available expertise that can be demonstrated; and (e) allow experimentation and externalisation of mental models. An obvious benefit of simulations is their ability to represent knowledge graphically, and to demonstrate knowledge in action through animation. This ability is particularly important for showing the consequences of actions on the state of a physical system because observing a physical change is more satisfying than imagining it.

The advantage of modeling is in simplifying reality by filtering out uninteresting elements and idealizing complexity to highlight interesting aspects. Rather than try to replace reality, simulations may be best used as a complement by creating situations that would be rare or dangerous in reality. This is a well appreciated benefit of simulations, which has real worth in training skills such as emergency procedures, e.g. shut-down.

Depending on the kinds of domain model underlying the simulation, considerable expertise could be available, which can be used for solving problems, as expert systems do (Clancey, 1987), or for evaluating performance, as with intelligent tutoring (Wenger, 1987). Many forms of knowledge are evident in simulation learning environments, including explicit domain models that can be simulated (mathematical and knowledge-based), non-executable support knowledge (on-line documentation), and tacit knowledge embedded in the display (graphical and iconic). All of these knowledge forms can be included in the corporate memory, and ways to capitalize on them are needed. Picard *et al* (1999) investigate the use of simulations in knowledge management of large scientific software and training. Their *Knowledge Book* is a document that details the software from many different views. Our approach uses multiple models of different complexity to create a corporate memory that contains many kinds of knowledge (Khan *et al.*, 1998b) in executable simulations.

Organisational learning with simulations can be supported by having users externalize their mental models, for example, by setting initial conditions and predicting outcomes of the simulation. Feedback is given through the simulation by supporting experimentation, allowing observations of resultant changes in the model, and through explanations generated to explain these observations

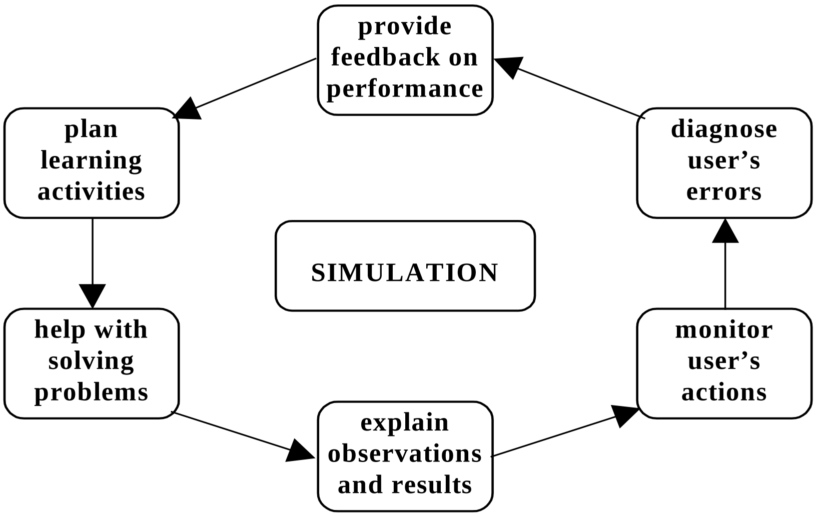

Figure 2 Functions of Simulation in Knowledge Capitalisation

Figure 2 shows a functional depiction of a simulation-based knowledge management system, in which a simulation component communicates with other components of the knowledge management environment to perform certain activities. These involve running a mathematical model to simulate changes in the state of the physical system being modeled in order to help with solving problems or for explaining results. For example, in an industrial process plant a water tank can be modeled with differential equations that describe how the rate of flow of water out of a hole affects the level of water in the tank. By setting different values for initial water level, and flow rate, the simulation can determine the new water level at a certain time, and possibly show the drop in water level graphically. However, simulation is just one component of a simulation-based knowledge management environment, and other components are responsible for planning interaction (Khan & Yip, 1996), helping with solving problems (Pěchouček, 1999), providing explanations (Forbus & Whalley, 1994), monitoring the user's actions, diagnosing problems and editing knowledge bases.

4 DESIGN ISSUES

When designing simulation-based knowledge management systems to have the functions outlined above, several issues must be considered. These are central considerations and it is important to be aware of the consequences of making choices in the design. Six important issues are examined here: fidelity of a simulation, pedagogical guidance, assumptions about expertise, assumptions about learning, role of explanations, and purpose of different activities.

4.1 Fidelity

A simulation can be used to create a similar environment to the real world by ensuring that the controls and panels on a display, for example, match the actual ones on the real system. The effects of changing these controls should be characteristic of actual effects too. This may be thought of as superficial fidelity (Reigeluth & Schwartz, 1989), since the similarities are only in what is observed. Situated learning theory argues for deeper authenticity in learning environments that attempt to replicate the ambience and conditions of a control room, and the associated psychological affects of being involved in a specific activity with specific goals in specific situations. How this level of fidelity might be achieved is not clear, however, and simulation-based environments tend to rely on what the user sees and how the simulation responds, to maintain fidelity, without including attitudes, feelings and other affective factors.

4.2 Guidance

There is considerable theoretical justification for adopting a *discovery* (Bruner, 1966) or *constructivist* (Piaget, 1926) learning strategy, and recently ideas from situated theory (Brown *et al*., 1989) suggest authenticity is a valid reason for employing veridical representations of the world. However, it is also recognized that unguided exploration fails to achieve effective learning of some aspects of a domain. Guided discovery (Elsom-Cook, 1990) is an attempt to bridge the obvious failing of purely discovery approaches, which have practically no pedagogical support. Various methods have been applied to introduce guidance to focus the user's attention and activities to make learning more meaningful in exploratory learning.

Context is the important factor here; it is what situationists have been demanding. However, authenticity appears to involve supporting indescribable elements of knowledge, such as attitudes and inclinations to believe as an expert believes. Context, in this respect, is evasive, since it is not clear how authenticity might be achieved with symbolic models. Other attempts try to contextualise learning by introducing teleology as a driving force to govern the user's explorations. This is manifested through goal-based scenarios (Bell, 1993), for example, which can be implemented with multi-media based stories (Schank, 1991) for capturing case and project knowledge. Explanations can be used to guide the user through a space of possible models during experimentation by selectively withholding certain knowledge, which should evoke the user in to seeking that knowledge. This approach is related to constructivism, and strategies that seek to implement cognitive conflict, e.g., impasse and failure driven learning (Vanlehn, 1987); the important pedagogical element is that the choice of knowledge to withhold is planned to achieve an overall instructional goal in line with organisation level objectives.

There are differences between education of students and industrial trainees which must be considered if trying to apply instructional design theories that were developed for academic purposes to industrial settings (e.g., Gagné *et al*., 1992). Students usually begin learning basic principles and conceptual knowledge and later practice applying it. Consequently, students are relatively expert in the theoretical aspects of their profession but lack the practical skills that develop only from extensive practice. There is a bias towards constructing a conceptual model containing the fundamental principles of a discipline in which theorizing comes first and practice second. In contrast, industrial trainees begin learning how to operate the plant through direct interaction in which a hands-on approach is favoured that seldom involves extensive theoretical coverage of the plant's behaviour. Clearly, there are two instructional strategies here: a theory-oriented strategy designed to develop a knowledge rich conceptual model of the plant with minimum operational skill, and a practice-oriented strategy designed to encourage highly efficient skills with limited problem-solving capabilities. The choice of setting, academic or industrial, will dictate which strategy should be incorporated into the environment.

4.3 Expertise

The constituents of expertise are often paraded as an ability to reason deeply about problems (Chi *et al.*, 1981), while still being able to apply shallower knowledge. Expertise includes several levels of knowledge, including meta-level problem solving skills. There are also affective skills (Krathwohl *et al.*, 1964), which might be re-interpreted as aspects of social knowledge (Khan *et al.*, 1998b). Other useful distinctions in knowledge for constructing corporate memories focus on *procedural* (e.g. *increase* the temperature setting, then *lower* water flow rate) versus *declarative* (e.g. temperature setting *is* 600 C) (Anderson, 1990); *compiled* (rules, e.g. IF temperature is too high THEN increase water flow rate) versus *episodic* (e.g. previously, temperature was lowered by increasing water flow rate) (Riesbeck & Schank, 1989); *strategic* (e.g. gather data on temperature before analyzing data on water flow) versus *basic* (e.g. data on temperature is retrieved from thermometers T1 and T2) (Clancey, 1987); and specific (e.g. water flow from Pump P1 flows to Tank T1) versus general (e.g. water *always* flows from pumps to tanks) (Rassmussen, 1983). Which of these distinctions is worth maintaining and which are not when modeling is not absolute; it depends much on the requirements for the corporate memory. For instance, if the goal is to develop an efficient operator of an industrial plant then an important distinction in domain knowledge is between procedural and declarative knowledge. Alternatively, if the goal is to acquire an ability to solve problems in many different, but related, functions in the organisation then distinctions between strategic/basic, and general/specific are important.

4.4 Learning

A model based approach to learning involves representing the learner's understanding of a subject as a partially complete model, which can contain many different kinds of knowledge. For instance, a mental model can be described with rules (Anderson, 1990), or causal relationships (White & Frederiksen, 1990). This view treats learning as the refinement and development of a model by adding new rules, deleting wrong rules and transforming longs chains of rules into shorter compiled rules (Newell, 1990), or progressing through a space of causal models (White & Frederiksen, 1990).

An alternative approach maintains that learning is more than evolving a single model and involves changes in different kinds of knowledge. Rules will be added to a rule-based model, causal relations will be established in a causal model, procedures will be constructed in a procedural model, concepts will be added in a conceptual model, etc. (Letich *et al.*, 1995). This is far more ambitious and represents the state-of-art in simulation-based environments. A particular benefit of this approach is that there is added flexibility in choosing objectives that specify elements of knowledge to present, and more power in the system's problem solving abilities. In contrast, environments that contain no explicit domain models of expertise allow exploration and experimentation but afford little pedagogical support; the benefit of these systems is easier development and good support for discovery learning.

4.5 Purpose

Of the many different kinds of knowledge in an organisation, a high-level distinction can be made on whether knowledge is of *how* or *what*. That is, is the emphasis of learning on accumulating declarative knowledge (principles, facts, labels, concepts) to construct a meaningful conceptual model (Ausubel, 1985), or on applying procedural knowledge (procedures, rules) to demonstrate skills? This distinction is between learning and training, and simulations can be used to support both activities (Reigeluth & Schwartz, 1989). It is necessary to understand the type of knowledge that is involved in either activity, since this will dictate *how many* domain models are needed. Also, developing skills requires a more active instructional strategy, whereas conceptual structures can be learned (partly) passively and rote, although an active approach is thought to be more effective (Jonnassen & Wang, 1993). An environment designed to support concept learning and skill development must maintain several representations of knowledge, and adopt different instructional modes.

4.6 Explanation

Closely related to instruction is explanation, which is a means of clarifying problematic situations, and for promoting reflection and conceptual change. Interaction between the user and the learning environment can take many forms, such as natural language dialogue (Cawsey, 1993). The simplest forms of explanation involve presenting ready made canned text in response to situations, or rule-tracing, which is re-presenting the rules that were used by the system in solving a problem (Clancey, 1987). More complex explanations seek to explain why the system performed some kind of reasoning, at the meta-level (Chandrasekaran *et al*., 1989). Explanations are used for simplifying and interpreting the output of simulation results too, particularly to interpret the mathematical equations in terms of causal behaviour (White & Frederiksen 1989). A flexible capitalisation strategy needs to consider how different kinds of knowledge interact during problem solving (e.g. procedures, rules, equations), and decide what kind of knowledge should be presented to the user. This is the focus of our work.

5 CAPITALISATION STRATEGY

Here we examine issue 6, Explanation, in more detail and introduce significant distinctions revealed during our work on the EXTRAS project, which aims to develop a decision support and training toolkit that uses simulations to produce dynamic explanations as part of a knowledge management system.

The *style* in which support is offered is important in enabling a problem situation to be interpreted accurately: a *reactive style* allows the user to indicate the support needed so that mediation is better focused; whereas in a *proactive style* the cognitive tool must assess the situation by identifying the mental model being employed by the user. This is possible through an analysis of the tasks used during training, although, it becomes more difficult when a novel task is being performed during decision support.

Another important factor is the *role* of the cognitive tool, which can be either *informative* or *didactic*. During an informative role, the cognitive tool is designed to place no interpretation on the information needs of the user; rather it must provide support as directly indicated by the situation, e.g. from requests of the user, or requirements of the task. In a didactic role, however, the cognitive tool must determine what support is needed beyond performance based, and include an aspect of instruction to support learning by the user. This requires more interpretation of the situation, and of the user's history.

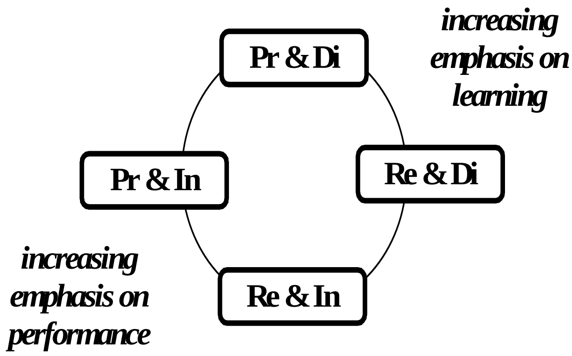

Figure 3: Operating modes for simulation systems

The role is determined by, and in turn determines, the cognitive tool's operating mode, which is a dynamic quantity that changes whenever either role or style changes. With respect to simulation-based knowledge capitalisation, and explanations in particular, the definition of operating mode of a system is tied in with the explanation strategy because the function of simulation is to facilitate communication of knowledge, and the function of explanation is to communicate that same knowledge.

The operating modes of a capitalisation system are points on a continuous spectrum that represents the learning-enabling capabilities of each operating mode. Figure 3 illustrates this. Operating modes can be one of four identifiable modes or innumerable intermediate modes. The spectrum is intended to indicate that as operating mode approaches proactive & didactic there is increasing acceptance of knowledge acquisition and refinement within the set of legitimate uses of simulation. This phenomenon is coincidentally accompanied by a decrease in the prominence of information provision as the only and main concern. Most important, though, the spectrum is a useful instrument for determining the comparative difficulty of the explanation problem in each operating mode. At one end of the spectrum there is a need only for information to be provided on request whereas the other end makes demands for intelligent decision making on the timing and content of information provision. The obvious conclusion is that providing learning places extra demands on explanation.

The range of available choices for interacting with the user provides axes for the explanation strategy to navigate. A specification of the explanation strategy involves setting values for both role and style. Certain situations do not demand complexity from explanations, however, so the explanation problem can be eased in these situations by admitting only those operating modes that make low demands on explanation (reactive and informative). We view explanation as a powerful tool for affecting the knowledge capitalisation process that must be used with awareness of its relationship with instruction. Rather than being governed by constrictive decisions implanted in an environment's design, issues prevalent in educational research, such as discovery versus expository learning, and meaningful versus contextualised authentic learning, etc., can be decided and alluded to by the insightful choice of explanations, done in real time and on-line to dynamically adjust the learning-enabling facility of an environment.

A conclusion that may be drawn from the foregoing discussion is that before we attempt to determine what to say in an explanation (local content decision) we must identify which operating mode is being used and then decide whether to remain in this operating mode or change to a more or less learning oriented mode. The change of operating modes is facilitated by a change in explanation strategy (since the two are linked). Therefore, meta-level reasoning about the choice of explanation strategy (i.e., role and style) is needed as well as reasoning about the contents of any explanation compliant with that chosen strategy. A further meta-level choice to make is with the style of interaction. Different kinds of learning can be encouraged by employing different *interaction tactics*.

6 INTERACTION TACTICS

The way in which information is presented to a user affects how they deal with it. In particular, the match between information provided and information needed will affect whether or not any further reasoning is engaged in by the user. Following results that point to there being three levels of reasoning by industrial operators (Rassmussen, 1983) three basic forms of learning can be supported in our work: deductive, inductive and rote. Each differs from the others in terms of the level of knowledge given with respect to the level needed. Levels are defined by a generality dimension (Leitch *et al.*, 1995) in which principles (constraints), associations (rules) and procedures represent three levels of knowledge that have a decreasing applicability to problems in a domain, i.e. generality. Rote learning involves presenting information that matches the level of knowledge needed, e.g. presenting associations when rule-based reasoning is needed. Deductive learning involves presenting knowledge that is at a level above that needed, e.g. presenting principles when rule-based reasoning is needed. Inductive learning involves presenting knowledge that is a level below that needed, e.g. presenting procedures when rule-based reasoning is needed.

Because industrial process operators (our chosen domain) are thought to use all these levels of reasoning in their jobs (Rassmussen, 1983) it is essential that a support system should recognize and take account of these different modes of reasoning to meet the information needs of operators in different situations.

The two sets of decisions to make in the capitalisation strategy can be summarized as: (1) decide on the role and style of an operating mode; and (2) decide on the interaction tactics.

7 THE MOBIT/EXTRAS SYSTEMS

The issues discussed above have influenced the design of two simulation-based systems developed by us: MOBIT and EXTRAS. Here we describe these systems and highlight the relationships between their implementation and the design issues.

An object oriented approach is taken with the implementation to ensure that the different components of the architecture can be developed separately (since many partners are involved), and that they can be easily integrated with new and existing simulations. The system is intended to provide explanation capabilities for simulation systems in both off-line training and on-line decision support modes by using a multiple model architecture. Table 1 lists the functions that must be supported by the EXTRAS architecture.

The central components of the MOBIT project (Mitchell *et al.*, 1996) are simulation and multiple domain models, which interact to assist the user in learning different kinds of knowledge. The EXTRAS project is now extending MOBIT with an explanation component that can generate new explanations containing different kinds of knowledge in different situations, such as training and decision support. MOBIT and EXTRAS are described here to demonstrate how the issues introduced above have practical relevance.

Table 1: Functions supported by EXTRAS architecture

Functions desired of Simulation Based Knowledge Management
interfacing to existing simulators
editing all types of domain knowledge, such as equations, associations, procedures
simulating with multiple model based simulators
defining exercises for off-line training
monitoring the plant state variables
monitoring user's actions and classifying any deviations as a particular error type
diagnosing user errors to determine their cause
logging events for debriefing at end of training session
modelling the user to guide selection of training exercises and explanations
providing on-line documentation support for the user
providing explanations to the user as advice and instruction
displaying all types of information to the user, such as explanations

The novelty of the EXTRAS architecture, Figure 4, lies in its use of simulations, modeling and knowledge-based reasoning supported by a methodology based on principles from educational and cognitive psychology as well as work practices.

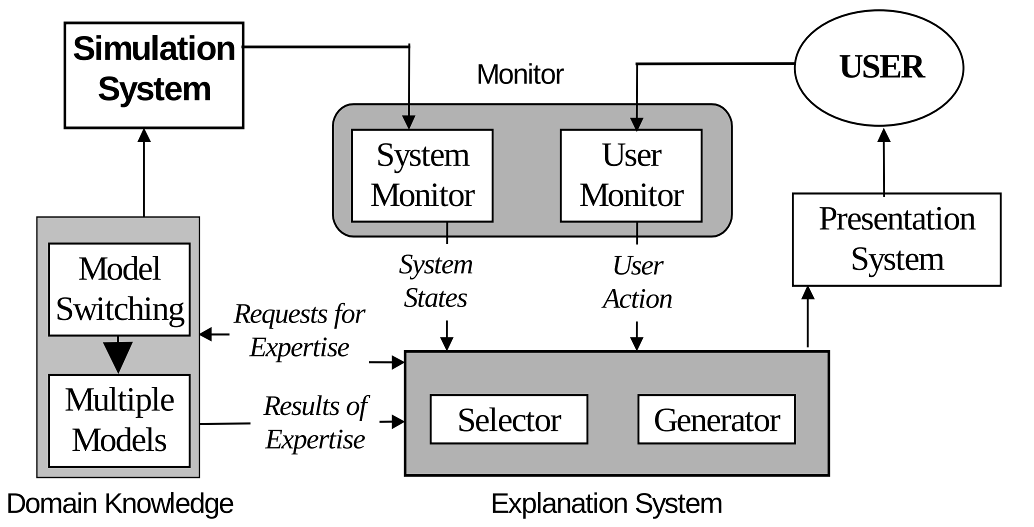

Figure 4: Architecture of Toolkit

7.1 Main Components of Knowledge Management System

The components of the architecture that are of most interest from a research perspective are the domain knowledge, monitor and explanation modules. Numerical Simulations are also an important part, however, novelty lies in their incorporation in the architecture rather than in their design, which is purposely meant to follow industrial standards. Each module comprises two main sub-modules, which have distinct functions. Within the domain knowledge module a multiple model module houses the set of domain models of the industrial system in an object oriented database. Selection of models is performed by the model switching module, which is held as a rule-base (i.e. a set of heuristics).

Explanations are produced by the collaboration of the selector module, which is another rule-base designed to identify the model needed for an explanation, and the generator module, which is responsible for locating and retrieving the chosen model. Information needed in the selection process is provided by the monitor module through the system monitor, which passes on states of the system, and the user monitor, whose role is to indicate what actions the user has performed and to interpret the reasons for any errors. Cooperation of all modules leads to an explanation being produced from an appropriately chosen model, which is displayed as text (for the moment) through a presentation component. For the current version of the application, presentation was not a major research issue, although future work is intended to consider alternative modes of presentation, such as graphical and animation. However, provided a suitable interface is available, the numerical simulation or any of the domain models can be executed to illustrate the effects of actions.

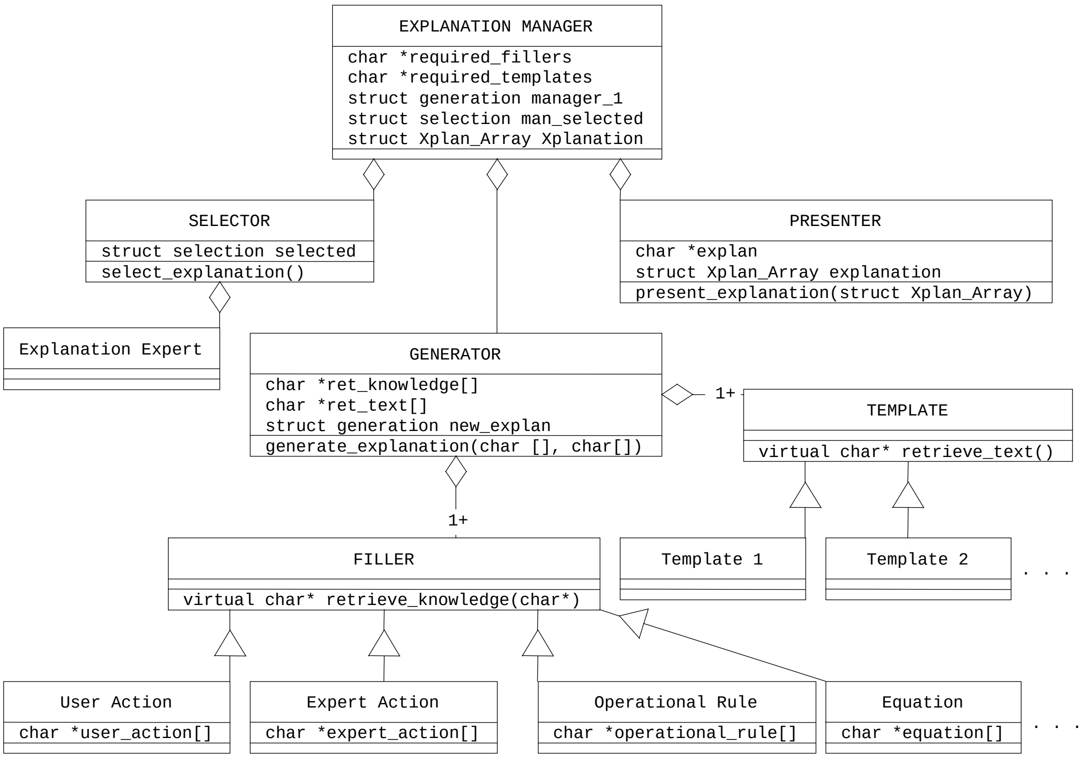

Figure 5: Object model of explanation system

7.2 Explanation

A strategy has been developed in a set of rules, which can choose an explanation for a situation. Types of error committed, the current task, and previous explanations provide the context for selecting explanations. The explanation subsystem, shown in Figure 5 (an object model), has an explanation manager containing several explanation types. Each explanation type is a separate object that instantiates its own version of inherited operations from the virtual Explanation Type object. (The implementation was done in C++ following the OMT approach).

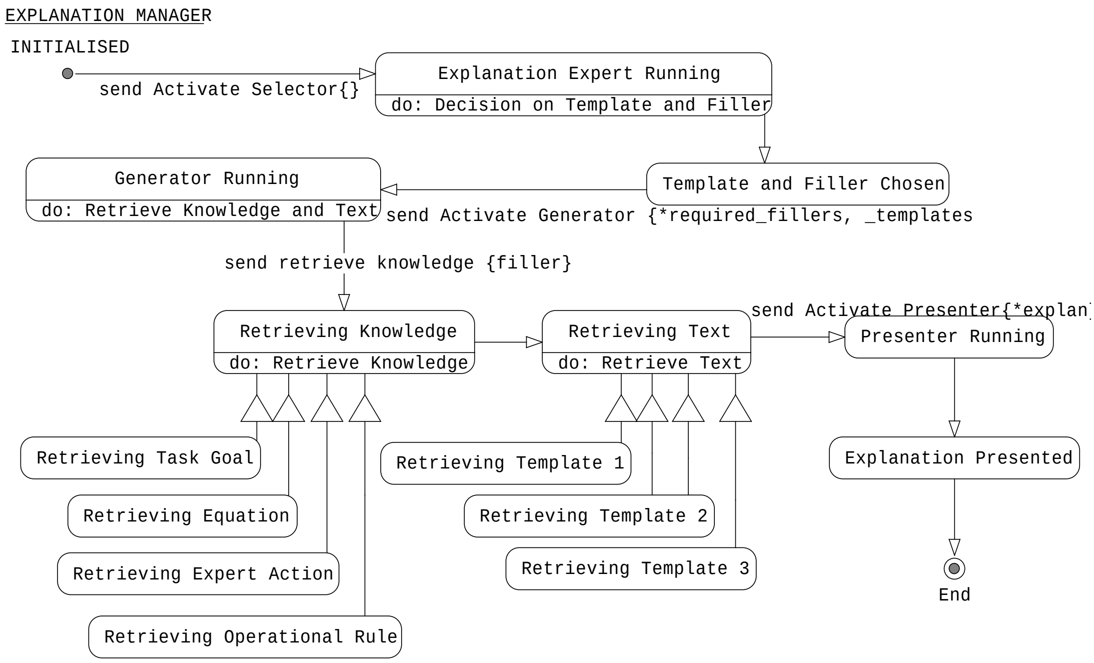

Figure 6: Dynamic model of explanation subsystem

The *explanation manager* is responsible for selecting the type of explanation needed, e.g. one with causal information or with procedural knowledge. The cause of error is used in this decision. Each explanation type is an autonomous object comprising a *template* and operations for retrieving knowledge needed to fill the template with *fillers*. More than one explanation object can be active at a time, and successive explanations involve successive activation of explanation objects.

Figure 6 shows the dynamic model for the virtual class explanation type; this model can be instantiated for any of the explanation types. Furthermore, each explanation type has several text templates available which each contain one or more fillers, Table 2. Activation of an explanation type, results in a template being chosen based on appropriate criteria, e.g., error type, and one or more fillers being activated. When a filler object is activated it requests the model manager, or either of the monitors to supply it with relevant knowledge, e.g., expert action, user action, etc. This knowledge is implanted in the chosen template to produce a complete explanation, which is then displayed. The choice of explanation object leads to selection of multiple models needed to construct the knowledge for each template, hence, this amounts to *model switching*.

Switching models here is not performed to track the user in order to evaluate performance, rather it is designed to *lead* the user to a different model. The pedagogical implications for this are strong because in this approach there is far more direction asserted on the interaction, so it is possible for a specific instructional goal to be supported. For example, if the instructional goal is to encourage the user to reason causally about the plant, an explanation can be chosen that contains compiled causality as rules. The user is required to *induce* the causality assumed in the associations. Alternatively, principles can be presented that require the user to *deduce* the causality. A third option is to present the causality explicitly, which is an example of *rote* learning. Induction, deduction and rote are examples of learning styles that were implemented in the MOBIT project (Mitchell *et al*, 1996) and can be supported by EXTRAS.

Each explanation is composed of a combination of knowledge and data fillers (Khan *et al*, 1997), shown in Table 2. These are the basic elements of the organisational memory and can be brought together to produce meaningful relationships. For example, an explanation that contains just *user action* requires the user to find the correct action to perform in a situation; a follow-on explanation could give the solution through the *expert action.* Another explanation could provide justifications of this action in the particular situation by presenting an appropriate rule or causal relationship. And yet another follow-up explanation could provide theoretical grounding for the rule by presenting a set of basic principles.

Table 2: Knowledge and data fillers used in explanations

Filler	Description
Task Goal	the current goal to be achieved by a procedure, e.g., raise electrical load.
Expert Action	the recommended action to be performed according to a normative model of the procedure, e.g., pressing a button, watching a dial, etc.
User Action	an action performed by the user in following a procedure, e.g., pressing a button, watching a dial, etc.
Error Type	a classification of any discrepancies between Expert Action and User Action, e.g., omission, reversal, intrusion, etc.
Expert Rule	a heuristic connected with a recommended Expert Action.
Causal Relation	a cause-effect relationship between an action and a state change.
Basic Law	a constraint connected with an Expert Rule or Causal Relation.
System State	the value of any relevant variable in the system.

8 DESIGN ISSUES IN PRACTICE

A multiple model architecture is employed (*issue 3*) in which there are three levels of knowledge to be learned: principles, associations, and procedures. Associations and procedures relate to developing skills, since they allow interaction with the simulation. For example, a procedure for increasing the set point of a valve includes *direct* reference to controls on the panel: “press the valve set point raise button, wait until indicator shows value has increased by 10%, then release”. This is a very simple procedure, but it demonstrates the direct relevance of a procedure to the controls and dials on a simulated panel. Associations (rules), e.g. “IF water flow rate is too low THEN increase the valve set point,” contain conditions that describe a situation (“water flow rate is too low”) and *goals* that procedures achieve (“increase the valve set point”), so there is an *indirect* reference to the controls through goals. An association is used to decide which goal is appropriate in a situation. The final level of knowledge is principles, which are basic equations for the domain, such as differential equations relating water flow rate to water level in a tank. These do not have any direct connection to procedures, hence principles have no explicit relevance to controls; instead relationships between measurable variables (e.g. temperature) and parameters to be calculated (e.g. mass) are contained.

Learning (*issue 4*) with the environment involves more than just learning a set of rules, rather it involves learning procedures and equations too, plus importantly, the strategies needed for knowing when it is appropriate to move from one kind of knowledge to another. This is a crucial aspect of expertise. The user must learn how to manipulate controls directly when given a procedure, know how to relate procedures to goals selected by rules, and determine the relevance of principles to operating controls and achieving goals.

MOBIT allows learning of skills through a cognitive apprenticeship mode (Brown *et al*., 1989), and conceptual knowledge through conventional tutoring. Skills are encouraged by first observing an expert operate a simulation of a boiler plant through a simulated control panel. There is no attempt to introduce authenticity beyond visual and behavioural fidelity with the panel (*issue 1*). The intention is to provide a simplified representation of the real control panel, so that learning is made easier. Training with the simulation is not meant to be a replacement for on-the-job training, rather it is preparatory to make the transition from classroom to control room easier.

The expert component in the architecture executes a procedure that involves altering particular controls in a set sequence. The user is then required to replicate this behaviour. Another level of knowledge is employed when the expert reasons about which controls to change when the sequence is not fixed. This involves using a rule-based model to determine goals that must be achieved. These goals are then attained by executing actions on the controls. Again the user is required to follow a similar line of reasoning by selecting rules and procedures.

Conceptual knowledge can be taught by an expository method in which the tutor selects instructional plans based on a profile of the user (issue 2). The plan introduces concepts related to the boiler plant. The emphasis here is not to develop skills in controlling the plant but to learn about how the plant behaves in a causal manner. In EXTRAS (Khan *et al*, 1997/1998a) this is supported by explanations (issue 6), which retrieve and present different kinds of knowledge, such as rules, causal relations, and recommended actions of a procedure. Thus the purpose (issue 5) of EXTRAS is to support both knowledge acquisition and knowledge application, and it does this through a combination of expert demonstrations, instructional planning and explanations.

9 CONCLUSION

The EXTRAS and MOBIT systems are an attempt to employ simulations in knowledge management to support dissemination of tacit and dynamic knowledge, which usually are not included in databases or knowledge bases. We discussed several important issues in the design of these simulation-based knowledge management systems and highlighted decision support and training systems as vital parts of a knowledge capitalisation strategy. It is important for system designers to be aware of these issues because they can affect how well any implementation satisfies the end-user's goals. For instance, fidelity is important in ensuring that the user has a smooth transition from the computer-based depiction of a physical system to the real system; however, the measure of similarity can vary from superficial visual and behavioural fidelity, to deeper and more subtle authenticity. The level of guidance dictates whether learning is geared towards self-development of subjective knowledge or towards objective knowledge through expository methods. It is desirable to support both modes to ensure all aspects of knowledge can be learned. There are several distinctions in expertise that must be considered when developing a domain model capable of solving problems; these include skills and concepts, and specific and general, among many. It is crucial to match the choice of knowledge with the instructional goals and organisation objectives. Furthermore, learning can be seen as the development of a single model of knowledge or the development of several domain models containing different kinds of knowledge. Explanations can be used to aid learning by presenting knowledge from these multiple models to lead the user along the path of an instructional strategy, which should be formulated in terms of kinds of knowledge to be learned. Finally, the purpose of the simulation-based system is key to matching the user's needs; for a system can be used to teach what to do, or how to do it.

References:

Anderson, J.R. (1990) Cognitive modeling and intelligent tutoring. *Artificial Intelligence* 42. 7-49.

Ausubel, D. (1985) Learning as Constructing Meaning. In *New Directions in Educational Psychology 1. Learning and Teaching*. (ed. N. Entwistle) pp. 71-82 The Falmer Press, London & Philadelphia, pp. 71-82.

Bell, B.L. & Bariess, R. (1993) Sickle Cell Counselor: Using a Goal-Based Scenario to Motivate Exploration of Knowledge in a Museum Context. *Proceedings of Artificial Intelligence in Education, AI-ED 93*, pp:153-160. Association for the Advancement of Computing in Education.

Bruner, J.S. (1966) *Toward a theory of instruction*. W.W. Norton & Company, New York.

Brown, J.S., Collins, A. & Duiguid, P. (1989) Situated cognition and the culture of learning. *Educational Researcher*, 18(1).

Cawsey, A. (1993) Planning Interactive Explanations. *International Journal of Man-Machine Studies* 38. 169-199.

Chandrasekaran, B., Tanner, M.C. & Josephson, J.R. (1989) Explaining Control Strategies in Problem Solving, *IEEE Expert* Spring. 9-24.

Chi, M.T.H., Bassok, M., Lewis, M.W., Reimann, P. & Glaser, R. (1989) Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science* 13. 145-182.

Clancey, W.J. (1987) *Knowledge Based Tutoring: The Guidon Program*. MIT Press.

Elsom-Cook, M.T. (1990) Guided Discovery Tutoring. In *Guided Discovery Tutoring - A framework for ICAI research*, (ed. M.T. Elsom-Cook) pp. 3-23. Paul Chapman Publishing.

Forbus, K. & Whalley, P. B. (1994) Using qualitative physics to build articulate software for thermodynamics education. *Proceedings of the 12th National Conference on Artificial Intelligence*.

Gagné, R.M., Briggs, L.J. & Wager, W.W. (1992) *Principles of instructional design*. Harcourt Brace Jovanovich.

Jonnassen, D. & Wang, S. (1993) Acquiring Structural Knowledge from Semantically Structured Hypertext. *Journal of Computer-Based Instruction*, 20(1). 1-8.

Khan, T. & Yip, Y.J. (1996) Pedagogic principles of case-based CAL. *Journal of Computer Assisted Learning*, 12, 172-192.

Khan, T.M., Brown, K.E. and Leitch, R.R. (1997). Didactic and Informational Explanation in Simulations with Multiple Models, in B du Boulay and R Mizoguchi (Eds.) *Artificial Intelligence in Education, Knowledge and Media in Learning Systems*, 355-363.

Khan, T.M., Paul, S.J., Brown, K.E. and Leitch, R.R (1998a). Model Based Explanations in Simulation Based Training, *4th International Conference on Tutoring Systems, ITS'98. San Antonio, Texas.*

Khan, T.M., Mitchell, J.E.M, Brown, K.E. and Leitch, R.R. (1998b) Situated learning using descriptive models. *Int. J. Human-Computer Studies* (49), 771-756.

Khan, T.M., Brown, K.E. and Leitch, R.R. (1999) Managing organizational memory with a methodology based on multiple domain models. *The Second International Conference and Exhibition on The Practical Application of Knowledge Management, PAKeM'99,* pp:57-76, London.
Krathwohl, D.R., Bloom, B.S. & Masia, B.B. (1964) *Taxonomy of educational objectives: the classification of educational goals: Handbook 2 - Affective domain*. London: Longman Group.
Leitch, R.R. *et al.*, (1995) Modeling choices in intelligent systems, *AISB Quarterly* 93, Autumn. 54-60.
Mitchell, J.E.M., Liddle, J., Brown, K. & Leitch, R. (1996) Integrating Simulations into Intelligent Training Systems, *EuroAI-ed '96*, Lisbon, September. pp. 80-87.
Newell, A. (1990) *Unified Theories of Cognition*. Harvard University Press, London.
Norman, D.A (1993). Things that make us smart. Addison-Wesley.
(Pěchouček, M. (1999) Met-knowledge representation for industrial decision making, in *The Second International Conference and Exhibition on The Practical Application of Knowledge Management, PAKeM'99,* pp:115-133, London.
Piaget, J. (1926) *The language and thought of the child*. London: Routledge & Kenan Paul.
Picard, S; Ermine, J-L & Scheurer, B (1999) *The Second International Conference and Exhibition on The Practical Application of Knowledge Management, PAKeM'99,* pp:93-114, London.
Rassmussen, J. (1983) Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics*, Vol SMC-13 No.3. 257-266.
Reigeluth, C.M. & Schwartz, E. (1989) An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction*, 16, 1. 1-10.
Riesbeck, C.K. & Schank, R.C. (1989) *Inside Case-Based Reasoning*. Lawrence Erlbaum.
Schank, R.C. (1991) Case-Based Teaching: Four Experiences in Educational Software Design. *Technical Report No. 7* Institute for the Learning Sciences, Northwestern University.
Vanlehn, K. (1987) Learning one subprocedure per lesson. *Artificial Intelligence* 31. 1-40.
Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos.
White, B.Y. & Frederiksen, J.R. (1989) Causal models as intelligent learning environments for science and engineering education. *Applied Artificial Intelligence* 3 (2-3). 83-106.
White, B.Y. & Frederiksen, J.R. (1990) Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence* 42. 99-157.

Acknowledgements:

The work described in this paper has been undertaken in Brite/Euram III Project BE-1245 (EXTRAS: Explanation in On-line Help and Off-line Training Based on Continuous Simulation) by the following partners: Alenia Marconi Systems (formerly GEC Marconi RDS), CISE, ENEL, Heriot-Watt University, Iberdrola, KCL, LABEIN, TECNATOM, United Paper Mills. The authors wish to acknowledge the contribution of all members of the project team to the ideas presented in this paper, whilst taking full responsibility for the way they are expressed.