FORMENTOR REAL-TIME DECISION SUPPORT FOR RISK MANAGEMENT

Roland Pennings, Gilles Gerlinger and Mylène Ponamalé
Cap Gemini Innovation
86-90, rue Thiers.92513 Boulogne-Billancourt. France
Voice: 33-1-49105389
Fax: 33-1-49100615
E-mail: Roland.Pennings@capsogeti.fr

KEYWORDS: risk management, decision support,methodology, process control.

ABSTRACT

The supervision of complex industrial processes, such as in the chemical, nuclear or aerospace industries, is a difficult task. The FORMENTOR methodology allows the development of supervision support systems to aid operators in their tasks, especially when dealing with perturbations and hazardous situations. A FORMENTOR system presents a synthetic plant-wide view of the current situation, diagnoses the underlying causes of perturbations, predicts possible future evolutions and proposes remedial actions. For this the methodology integrates models, techniques and tools from four domains namely artificial intelligence, safety analysis, real-time computing and ergonomics. The methodological approach provides firm control over the development process while meeting the quality and technical requirements of the client; furthermore it ensures the reliability and maintainability of the obtained system

INTRODUCTION

The supervision of hazardous industrial processes is a difficult activity. Traditional process control techniques partly support the work of the operator. They can help in regulating and optimizing the process when the process is within known bounds. They do not give support however in the case of perturbations. The aim of the FORMENTOR project, which is part of the EUREKA program of cooperative european R&D projects, is to construct systems which help the operator when the process goes out of bound and perturbations or even hazardous situations occur. The ultimate objective of the project is to avoid perturbations that may lead to any type of loss: loss of production, start-up costs related to shutdown and in particular accident losses (Wilikens et al., 1993).

The development of a FORMENTOR system draws on techniques from four domains:

Safety analysis: safety studies are performed during the design stages of hazardous systems to show the relationships between failures such as component breakdowns, external events which include human actions, and feared events which could cause a certain loss. Until now the results of these studies are only implicitly transferred to the operators through a set of operating procedures and alarm management systems. The aim of FORMENTOR is to make this safety knowledge explicitly available to the operator.

Artificial intelligence: given the complexity of the plants and the number of potential perturbations AI techniques, such as heuristics and model-based approaches, are the most appropriate to provide support on time. A FORMENTOR system has to incorporate different types of knowledge, such as diagnosis knowledge to find the causes of observed symptoms and safety knowledge to detect and evaluate the current threats. We have chosen a knowledge based system approach, because it clearly seperates this knowledge from the reasoning processes using it.

Real-time computing: the supervisory support system is connected on-line to the target plant and has to take into account real-time constraints, such as the continuously changing plant conditions, deadlines and reactivity to external events.

Ergonomics: for the adaptation of the advisory system to the activity of the operators and for the acceptance of the system by them, ergonomics techniques play of course a crucial role.

The majority of currently available commercial products provides limited and partial support for the construction of advisory systems for operators. For example, AI process control tools are on the market which provide a diagnosis and simulation capability (see for example Årzen, 1992 for an overview). Common among these tools is a graphical, object-oriented and rule based programming

environment. They provide a tool for defining the structure of the plant graphically, an object-oriented and rule-based programming environment, and a built-in simulator. One has to keep in mind however that these tools only support the design and implementation stages. They do not support the earlier stages of the development such as the analysis of the problem domain.

The methodology presented in this paper supports the construction of an industrial quality advisory system during the whole development process. For this it provides to a development team a complete set of models, techniques and tools. The way in which the methodology is constructed can be represented as a pyramid, as shown in Figure 1.

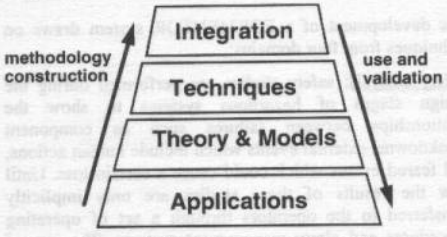

Figure 1: The Methodology Pyramid

First, we have identified pilot applications to obtain sufficient knowledge of the safety requirements in the supervision of hazardous systems (Section 2). We proceeded to develop analysis and design models to satisfy these requirements (Section 3). The models in their turn have been the basis for a set of techniques to support the construction process of an application (Section 4). Then we have integrated the models, techniques and tools in a methodological guide (Section 5). Finally, the pilot applications have served to validate the obtained results.

THE PILOT APPLICATIONS

Within the FORMENTOR project, one case study and two pilot applications have been constructed within the nuclear, aerospace and process industry domains:

The case study was concerned with the monitoring of a simulated Auxiliary Feedwater System (AFWS) of a nuclear plant. A prototype has been constructed which helps an operator to keep the plant in hot standby for as long as possible before proceeding to cold shutdown.

The first pilot application has as target process the on-ground filling process with liquid helium of an ISO-Satellite (Infrared Space Observatory). During the filling several hazardous situations can occur, such as the explosion of the vacuum vessel because of an overpressure or damage to instruments because of unacceptable temperature gradients imposed on the optics. An application has been developed for Aérospatiale which supports an operator in charge of the filling process.

The second pilot application has as target process a butadiene extraction plant. The plant is equipped with a process control computer, based upon feedback and feed-forward mechanisms which control the process when it is within known bounds. An application has been developed for British Petroleum which supports an operator when, as a result of a disturbance, the process deviates from the operating constraints. The system offers advice on the underlying causes and how to return to a state where normal control can be resumed.

Based upon the experiences with the pilot applications the following generic functions have been identified for a FORMENTOR system:

- monitoring: it validates sensor readings and detects symptoms which indicate an abnormal situation;
- diagnosis: it diagnoses the symptoms to deduce the underlying causes which have given rise to the abnormal situation;
- situation assessment: it assesses the current state of the target plant in terms of achievement of the plant goals and functions;
- consequence assessment: it predicts future situations and assesses the criticality of these situations; this also allows the operator to ask "what if" questions; and
- action planning: it determines the most appropriate sequence of actions to bring the plant back to a safer state.

The next section describes the theory and models which incorporate the underlying knowledge to provide the functions listed above.

THEORY AND MODELS

A safety-oriented supervision support system has to incorporate different types of knowledge of safety and plant experts. To integrate this knowledge efficiently in an industrial quality system it is necessary to use a structured approach. As a framework we have chosen the CommonKADS knowledge engineering methodology (Schreiber et al., 1993). According to CommonKADS, the development of a knowledge based system consists of the transformation of intermediary models. This breakdown diminishes the complexity of the development process. Each model corresponds to a specific concern, such as the organization, the tasks required, the agents involved, the experts' knowledge, and the design of the final system.

A distinction is made between the analysis and design activity. The analysis activity describes expertise at a conceptual level; the design activity on the other hand describes how this expertise can be realized as a

knowledge based system. The advantage of having this distinction is that the expertise can be clearly expressed independently of implementation issues, which helps in constructing an understandable and maintainable system.

The Analysis Activity

One of the main outputs of the CommonKADS analysis activity is the expertise model, which models the expert's knowledge. The model makes a distinction between three different types of knowledge: domain, inference and task knowledge. The domain layer captures the knowledge of the application domain, such as the concepts, their properties, and the relationships between concepts. This layer also contains models of the target plant. The inference layer represents the reasoning process of the expert performing his/her task. Finally, the task layer specifies the reasoning tasks, their goals and the control over the associated reasoning processes.

To support the knowledge engineer constructing the expertise model we have developed the FORMENTOR task library. This library permits the speed-up of the analysis activity, because the tasks give the knowledge engineer a starting point to develop the expertise model for a new application. The library also provides the means of capturing the experiences gained in developing FORMENTOR systems for later reuse: when new applications are developed, the library is enriched accordingly. As an example, Figure 2 shows the FORMENTOR inference structure for the overall supervision support task.

The task library has been developed on KADS-TOOL workbench, an industrial workbench for the CommonKADS methodology, which supports a knowledge engineer in knowledge acquisition and the construction of an expertise model. The workbench allows the integration of the interviews with the experts and from these the extraction of concepts, attributes and relations. Based upon the interviews and the generic task library, the tool aids in the construction of the inference structures and the task layer of the expertise model.

To give to the operator diagnosis, situation assessment and prediction capability, we need to incorporate knowledge about the plant. A multi-model approach of plant models has been chosen to gather and represent this knowledge. This approach allows the knowledge engineer each time to focus in turn on each particular aspect of the plant. It also makes the obtained system easier to maintain because when changes are made to the plant, only the plant models have to be adapted. The models present different viewpoints of the plant, such as a functional safety-oriented view, a hierarchical component view, and behavioral views in normal and degraded conditions.

The main model which FORMENTOR proposes is the Goal Tree - Success Tree (GTST), which provides a functional safety-oriented view of the plant, relating high-level goals to hardware and process dependent functionalities. This model is used to show the criticality of the current situation by dynamically evaluating the plant goals achievement. The model also serves in selecting the most appropriate countermeasures to repair malfunctions related to the unachieved goals. The GTST originates from the work of Kim and Modarres (1987). Within the pilot-applications the model has been adapted to make it suitable for safety-oriented supervision.

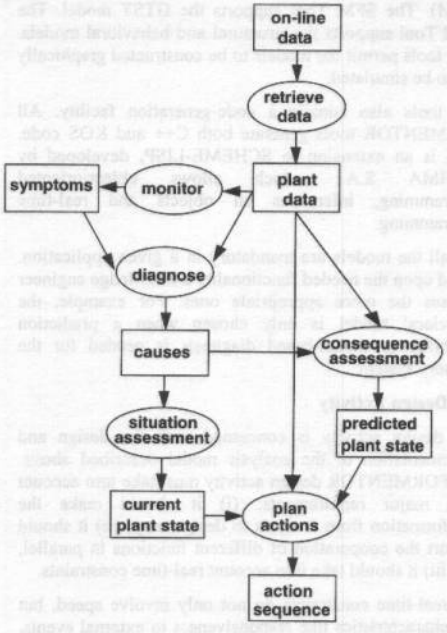

Figure 2: The inference structure for the overall risk management

Other models that are used are the plant structural model, the plant behavioral model, and the plant causal model. The plant structural model gives a component decomposition of the plant. This model presents the diagnosis results and is used as the underlying structure for the behavioral and causal models. The plant behavioral model describes the behavior of the plant. This model is used for model-based diagnosis and for the prediction of the future behavior of the plant. Finally, the plant causal models are used for a heuristical diagnosis based upon causal relationships. The models are inter-related. For example, a failed component of the structural model can be used as input for the evaluation of the achievement of a goal in the GTST.

The plant models have an analysis part and a design part. At the analysis level the plant models correspond to the domain models of CommonKADS, because they give to the knowledge engineer a certain viewpoint on how the domain knowledge should be structured. Unlike the domain models however, the plant models also have a design part, because they allow direct code-generation and they incorporate algorithms which allow the models to be manipulated in real-time. For this we have developed two software tools: the Safety and Functional Model Tool (SFM) and the Structural and Behavioral Model Tool (SBM). The SFM Tool supports the GTST model. The SBM Tool supports the structural and behavioral models. Both tools permit the models to be constructed graphically and to be simulated.

The tools also contain a code-generation facility. All FORMENTOR tools generate both C++ and KOS code. KOS is an extension to SCHEME-LISP, developed by SODIMA S.A., which allows object-oriented programming, inferences on objects and real-time programming.

Not all the models are mandatory in a given application. Based upon the needed functionality a knowledge engineer chooses the more appropriate ones. For example, the behavioral model is only chosen when a prediction capability or model-based diagnosis is needed for the advisory system.

The Design Activity

The design activity is concerned with the design and implementation of the analysis model described above. The FORMENTOR design activity must take into account three major requirements: (i) it should make the transformation from analysis to design easy, (ii) it should support the cooperation of different functions in parallel, and (iii) it should take into account real-time constraints.

The real-time constraints do not only involve speed, but also characteristics like responsiveness to external events, non-monotonicity and graceful adaptation. Though FORMENTOR systems respond to the characteristics of real-time systems, the time constraints are rather soft. Because the system is advising an operator there are always a number of seconds before a response is called for. The key problems are more in reasoning and adapting the advice in a changing world than in giving a fast response.

We have chosen an object-oriented approach as a basis of the design model, because features like encapsulation, code reuse and inheritance enable the rapid development of application modules which are understandable and maintainable (Schlaer and Mellor, 1992). Special constructs are needed however to take into account the real-time constraints. For this we have developed a data model which is based upon information propagation.

Information propagation is related to the need for a task to determine rapidly which objects are concerned by the modification of a given piece of information. This avoids reconsideration of all the objects after the modification of one of them. This concept is realized by structuring the objects in a fixed network, so that, after the modification of an object, information can be propagated directly to the objects which are concerned by the change.

The cooperation of different functions and the real-time constraints lead to a software architecture composed of independent modules which may run concurrently, if necessary on different processors. A module is then considered as a collection of objects and a local controller which manages the activity within the module. The overall system is in turn composed of a collection of modules and a global controller which manages the concerted activity of the modules.

The controllers have to ensure a reactive behavior of the system. To model this we have chosen the Statechart formalism which support a graphical specification of event-driven control (Harel et al., 1990). This makes the control accessible and easily modifiable. For example, a model-based diagnosis can take quite some time. When in between a more urgent problem shows up however, we want to be able to directly treat the new problem. Statecharts are very efficient for modelling this. They represent the control as state-transition diagrams with three additional features, depth, orthogonality and broadcast communication. These features ensure that the statechart specifications remain clear and concise for the description of a complex system.

To support the design activity we have developed two other software tools: the Data and Communication Management Tool (DCM) and the Statechart Design and Simulation Tool (SDS). The DCM Tool supports the graphical specification of modules, classes, objects, and information propagation structures. The SDS Tool supports the graphical specification of a statechart, provides consistency checking (such as non determinism and unreachable states) as well as a simulation environment.

TECHNIQUES

Until now we have described the theory and models part of the methodology, which explains in what way a FORMENTOR system should be modelled; another important aspect of a methodology however is to explain how this should be done. In the FORMENTOR methodology three different types of techniques have been developed: for the construction of the different models, for the re-use of safety analysis results and for the ergonomic activities.

The model construction techniques support the construction of the expertise model and the transformation from expertise to design model. Some of the transformations can be automated. The plant models for example can be directly translated into software code. This means that for these models a separate design representation is not necessary. For other knowledge in the expertise model this is not so simple however. An inference structure, for example, describes the reasoning process of an expert without taking into account implementation issues. This makes it impossible to automatically transform these into information propagation structures. We have therefore developed a design library of information propagation structures. The inference structures of the analysis library contains indexes to corresponding information propagation structures of the design library. For example, in the monitoring inference structure the inference step validate gives an index to information propagation structures of all sorts of sensor validation techniques, such as checking of process limits and stuck values.

The safety analysis techniques allow the re-use of the results of existing safety and reliability studies, such as Failure Mode Effect & Criticality Analysis (FMECA), HAZards & OPerability (HAZOP) studies, Fault-Tree analysis and Functional Block Diagrams. These studies incorporate a lot of information which can be used to construct the different plant models. The safety analysis techniques present which parts of the studies can be re-used and how.

Finally, the ergonomics techniques have been developed to incorporate a human-factors approach. Indeed, ergonomics plays a crucial role for the adaptation of the advisory system to the activity of the operators and for the acceptance by them. Having the right information available at the right moment is vital in order that the operator takes the appropriate actions as soon as possible when the process does not behave as prescribed. The methodology therefore integrates ergonomics techniques, which takes into account the needs of the operators, their activities, their characteristics, and their environment. The techniques encourage the active participation of the operators during the development. In particular, several prototypes are constructed during the development process, started as early as the requirements definition phase. These are used and evaluated by the operators themselves with the help of an ergonomist.

INTEGRATION: THE METHODOLOGICAL GUIDE

As the final step in the construction of the methodology we have integrated the developed models, techniques and tools into a guide. The aim of the methodological guide is to support a development team during the complete development process. It contains the following parts:

- a management part which aids the development team during the development process;
- a techniques library which contains the techniques for model construction, re-use of safety studies and ergonomics;
- a FORMENTOR task library which contains the inference structures for the analysis activity;
- a design library which contains the information propagation structures for the design activity;
- a plant model set which describes the different plant models: the GTST, structural, behavioral and causal models;
- a tool set which presents the developed FORMENTOR software tools;
- a living experiences part which describes the experiences with the developed pilot applications.

The management part shows a development team how to proceed in developing a new application. To support this it contains: a development task library and a management aid based upon the spiral life-cycle model.

The development task library supports the development team with a directory of all tasks which must be performed to ensure the quality of the system. The library is sub-divided into nine phases (such as requirements definition, functional specification, integration and validation) which themselves are sub-divided into about fifty tasks. For each task we define: an overview of the task, the activities needed, the inputs and outputs (such as documents and software code), the actors who intervene, and the models, techniques and tools which can be used. The directory gives an exhaustive list: for the development of a specific system not all tasks may be necessary.

Special care has been taken concerning the validation tasks of a FORMENTOR system. There are two possible approaches to this, the first one is using a dynamic simulator of the target process and the second one is to use scenarios of the real plant. The advantage of a simulator is that it can model many different sorts of perturbations without endangering the actual plant. A simulator has disadvantages however. First of all, the development of a simulator which simulates many different fault conditions is elaborate and costly. Secondly there is also the question of the validation of the simulator itself. It is difficult to guarantee that the simulator has the same behavior as the plant. This means that a FORMENTOR system which is validated against a simulator has to be re-tuned for the actual plant. At the moment we believe that it is better to validate against real scenarios of the plant. If needed these scenarios can be worked upon to simulate perturbations which cannot be performed in the real plant.

Finally, the spiral life-cycle model has been chosen to manage the development process. The spiral model breaks down the development process into several cycles which facilitates the construction of a complex system [Boe88]. At the beginning of each cycle the technical results which must be obtained at the end of the cycle are defined. Based upon these results a task-planning is constructed for the cycle. The tasks for this planning are then chosen from the development task library. The spiral model provides firm control over the development process and assures that the requirements of the client are met.

Together with the software tools the methological guide permits the development of applications with a minimum of delay and risk. The guide is backed-up with seven training modules, which cover all aspects of the methodology.

CONCLUSION

This paper presented the FORMENTOR methodology, which integrates models, techniques and tools for the construction of supervisory support systems of hazardous processes. The methodology is original in that it supports all aspects of the development process: theory and models, techniques, tools and the integration of these in a methodological guide.

For the knowledge engineering aspects the methodology is based upon CommonKADS, enriched with a FORMENTOR task library and specific plant models. The plant model approach is necessary for the development of a maintainable system, because when changes are made to the target plant, only the corresponding parts in the models have to be changed.

The design part of the methodology is specifically developed to take into account the real-time constraints. It is based upon an object-oriented approach, enriched with information propagation and propagation control. A design library has been developed to facilitate the transformation from expertise model to design.

Four software tools support the construction of the plant models, the propagation model and the statechart controllers. All tools have a code generation facility. This greatly increases the speed at which the system can be delivered.

Finally a methodological guide is constructed which integrates the developed models, techniques and tools. Furthermore it supports the management of the development process with a development task library and a spiral life-cycle model. The guide is backed-up with a FORMENTOR training.

Three pilot applications have validated the methodological approach. The project has now entered the industrialization phase and the methodology is currently applied in an industrial petrochemical application: a supervision support system for an industrial cracker.

ACKNOWLEDGEMENTS

We would like to thank all the members of the FORMENTOR project who contributed to the methodology.

REFERENCES

Årzén K.E., 1992. A Survey of Commercial Real-time Expert System Environments. IFAC Symposium on Artificial Intelligence in Real-Time Control, Delft.

Boehm B., 1988. A Spiral Model of Software Development and Enhancement. IEEE Computer.

Harel D., Lachover H., Naamad A., Pnuelli A.A., Politi M., Sherman R., Shtulltrauring A. and Trakhtenbrot M., 1990. Statemate: a Working Environment for the Development of Complex Reactive Systems. IEEE Transactions on Software Engineering, 16 (4): 403-414.

Kim I.S. and Modarres M., 1987. Application of the Goal Tree - Success Tree Model as the Knowledge Base of Operator Advisory Systems. Nuclear Engineering and Design, 104: 67-81.

Schlaer S. and Mellor S.J., 1992. Object Lifecycles: Modelling the World in States. Yourdon Press Computing Series, Prentice-Hall, Inc., New-Jersey.

Schreiber G., Wielinga B. and Breuker J., 1993. KADS A Principled Approach to Knowledge-based System Development. Academic Press, London.

Wilikens N., Nordvik J.P., and Mitchison N., 1993. FORMENTOR: Real-Time Expert Systems for Loss Control, ISSA 93 Conference: "Safety Pays!", Lugano.