MONET2

Project Full Title: Network of Excellence on Model Based Systems and Qualitative Reasoning.

Contract: Concerted Action / Thematic Network

Contract No: IST-33540

Deliverable TR1:

Technological Roadmap (Version Three)

Date: 30th June 2003
Version: 1.2
Status: Release

MONET Project Office
Department of Computer Science
University of Wales
Aberystwyth
Ceredigion
SY23 3DB

Contents

Authors

Co-ordinating Node	MONET Project
Automotive Task Group	MONET Project
Bio-Medical Task Group	MONET Project
Education and Training Task Group	MONET Project
Bridge Task Group	MONET Project

1 Introduction

1.1 Purpose of this Document

MONET is the European Network of Excellence in Model-based Systems and Qualitative Reasoning. In 1998, it produced two versions of a roadmap for the technologies covered by the Network. This document is the third attempt, five years later, to identify the likely industrial development of the technology, and the improvements to the technology needed to achieve the expected industrial development.

The first MONET roadmap attempted to map possible applications of the technology against time. This roadmap takes a different approach. MONET2 concentrates on four areas of application of model-based and qualitative reasoning (automotive, applied diagnostics, biomedical, and educational applications). A roadmap has been produced for each of the application areas, examining technology drivers in that application area, and matching them to the capability of model-based reasoning systems. The roadmaps for the individual application areas have identified technological barriers to applications of model-based reasoning. This document collects and collates those technological issues, presenting them in a more logical order. The issues identified from the four main application areas have been supplemented from a wide range of other application areas for model-based reasoning. It should be noted that there are differing views in the different roadmaps, both on timing and on the importance of different features. This is only to be expected, as the emphases and outlook varied between the groups. This overall roadmap takes a balanced view of the different emphases.

1.2 Scope

The roadmap arranges the technological issues in the following way:

Section	*Subject*	*Sub-areas*
Section 2	Improving Modelling Products. This section considers the improvements needed to be able to represent and reason about a wide variety of complex real world systems.	Ontologies – deciding how to model a specific domain in an effective and efficient manner. Model Integration – combining different kinds of models to produce a larger model-based reasoning system.
Section 3	Processes for Model-based Engineering. This section concentrates on issues related to the industrial deployment of model-based technology.	Model Generation – tools to produce models more efficiently. Model Conversion – tools to turn existing information into executable models. Model Management – tools to enable the sharing and maintenance of models. Standards – the basis for sharing and reuse of models.
Section 4	Task-oriented Challenges. This section concentrates on issues concerning the integration of model-based technologies within larger problem-solving systems.	Hybrid reasoning systems – the integration of model-based systems with other reasoning technologies. Learning as diagnosis - modelling learners in order to diagnose their misunderstandings. Interpreting real world data – interfacing between the model and the world.

Appendices A to D of this document contain the roadmaps for the individual areas which are the focus of the MONET2 Task Groups. These act as an application-oriented pull on the technology discussed in this roadmap. The topics in Sections 2 to 4 have been generated by considering the technological needs of the four individual areas, and then arranging the needs in a structured way. The issues listed have then been expanded by considering the needs of a wider range of application areas than just the four represented by the Task Groups.

2 Improving Modelling Products

2.1 Introduction

One of the main factors slowing down the adoption of model-based systems is the lack of easy-to-use, off-the-shelf models for a range of domains. This section considers the areas where modelling needs to improve if we are to be able to reason about real world systems in an efficient and effective way.

The subsection on ontologies discusses specific areas where the need for improvements in modelling is particularly evident. The subsection on model integration considers the issue of being able to combine models in a seamless way.

2.2 Ontologies

2.2.1 Models of Complex, Dynamic, Time-varying and Continuous Systems

This is a particular acute issue: many important real world applications fit this description, and continued improvements in this area should lead to adoption of model-based techniques for many applications.

What the Problem is

Many of the processes that we try to model evolve in time, happen in a particular space, are impossible to specify completely as not all relevant parameters can be determined (giving rise to uncertainty). In addition, lack of data makes it impossible to describe the system quantitatively. Many real-world systems are very complex; and while the exact nature of the complexity varies from system to system, the contributors to degree of complexity are: non-linearity, order, dimensionality, degree of coupling and non-determinism.

Examples of such systems are:

- Human brain, cardiovascular system
- Macro-economic systems
- Chemical process plant

State of the Art

A great deal of work has already been done in the engineering disciplines, but such work often relies on the availability of quantitative models. Model-based and qualitative reasoning methods exist to deal with situations where such quantitative models are not available, but are normally restricted to one aspect, such as dynamic behaviour, of a problem. They cannot combine various modalities, such as time, space and uncertainty, which may be needed for modelling the real world. In addition, the expressive power of many qualitative methods is restricted in the sense that even within one modality not all required behaviour can be modelled. In addition, many of the available methods do not scale up to real-world models.

Some examples of past and present approaches to dealing with the various aspects of complexity are:

Quantitative Modelling

- Identifying types of non-linearity (e.g. Michaelis-Mentin, Product, Power)
- Coupled system analysis – for loosely and tightly coupled systems
- Partial differential equations to deal with dimensionality
- Perturbation analysis to handle systems operating on different timescales (stiff systems)

Qualitative Reasoning

- Generalising linear / non-linear models
- Function free estimation (via fuzzy quantity spaces)
- Time scale abstraction (matches perturbation analysis)

In addition there are a number of approaches to categorising models which enable the identification of types of complexity:

- Model categorisation on the basis of model properties
- Multimodelling on tetrahedron of state (includes functional)

What Further Research is Needed

More powerful modelling languages:

- Allowing levels of abstraction in languages describing temporal processes
- Languages for the description of uncertain evolution of processes, possibly of a qualitative nature
- Integration of various modalities into the same language

Coupling of models:

- Development of tools to integrate types of complexity in appropriate models at varying levels of abstraction
- Modelling complexity by various means: simplification, abstraction, etc

Extended definition of a space of models from which desired model can be selected. Partial models. Integrated co-operative models dealing with single, or a small number of aspects of complexity

- Model definition framework
- Strategies for co-operative use

2.2.2 Models of Software

The modelling of the action and influence of software is an issue for almost any advanced man-made device or system. For example, in the automotive domain, electronic control units

(ECUs) containing many thousand of lines of software control the state of vehicle subsystems, and often perform monitoring, diagnosis and reconfiguration of systems. It is necessary to incorporate the actions performed by software in models, in order to understand the state of the device, and perform device-specific tasks. Similar issues occur in other domains, such as model-based reasoning about process control systems.

The deployed State of the Art in this area can be represented by the AutoSteve system. Abstract representations of the software in state charts are incorporated within ECU components to reason about the overall behaviour of electrical / electronic systems. State changes within the software are mapped onto electrical path changes in the electrical system, and vice versa. This is adequate for many simple representations of software, but fails to capture the complexity of the software, and so can mean that incorrect results are generated where the software does not exactly mirror the behaviour of the state chart description.

It should be noted that present commonly used software engineering modelling methods such as UML (Universal Modelling Language) do not provide a solution to this problem, as UML is not directly executable - it suffers from the same problem of not necessarily matching the actual computer program implemented.

There are three possible ways in which this situation can be ameliorated. The first is to have an executable description language from which programs can be automatically generated. The model-based system could then interpret the requirements, and use them in reasoning. This might be referred to as automatic programming from requirements, and has been pursued for the last 20 years without notable penetration of the software engineering industry. A second solution is to be able to abstract models from programs and use those models within a model-based system. The third solution is “software in the loop” where the model-based system interacts directly with the chip containing the software, and so the behaviour of the overall system can be modelled with the software that will run in the real system.

2.2.3 Models to Represent System Specifications and Requirements

One of the major advantages of model-based reasoning for problem-solving is that it can consider many more possible scenarios than a human could. One of the key concepts for qualitative model-based systems is that of an ‘envisionment’. An envisionment is a map of all of the possible states that a given system can reach, and how the system moves from one state to another. It is generated by exhaustive simulation from all states to see what other states can be reached.

For systems where safe operation is an issue, an envisionment provides important indications of the possibility of reaching unsafe states or situations. In other types of application, it might be possible to specify ‘interesting’ states of a different sort.

In order to identify interesting / unsafe states, two things are needed:

- Descriptions of what is interesting. This can involve capturing descriptions of the way in which the system should work, and might include issues of complex dynamic time varying and continuous systems, dealt with elsewhere in this section.

- Abstraction of state descriptions. It must be possible to abstract the results of an envisionment so that they can be compared with the descriptions of interesting states.

This area is in its infancy, but we would expect it to make a significant contribution to system safety and reliability as it becomes better developed.

2.2.4 User Modelling

Many applications of MBS require that features of the user are modelled and taken into account during the reasoning (problem solving) process.

For example:

- In automotive applications, the driver, her identity, her capabilities, her preferences about driving style should be taken into account in model based monitoring/control of the system. Alternative features of the user could be taken into account in the design phase to generate a personalised version of the car (again as regards controlled systems).
- In educational systems, the learner, her capabilities (e.g., cognitive abilities), background (what she already knows) should be taken into account in the definition of the process.
- In biomedical applications (in cases when the patient is not the subject of modelling), it can be useful to model patient features and preferences.

We can see from this that there are specific classes of users - driver, patient, learner - and what we need to model will depend on the type of user, and the task they are trying to accomplish. At present techniques for user modelling have been produced and widely used in other communities (see the User Modelling and Adaptive Systems community). Different techniques are being used in these fields to model different dimensions of a user, in relation to different goals to be achieved by an adaptive problem solving agent. Education systems are one of the main applications.

However, these techniques have to be integrated into MBS system, as a new dimension of modelling where the user in not the main target of problem solving, but one of the elements to be taken into account. For example, in a car, while reasoning on a model of the engine behaviour in order to monitor it and define optimal control strategies, a system may also take into account a model of the users preferences to tailor these strategies as much as possible (the main goal, however, still remain optimal monitoring and control).

Thus further research is need in order to study these forms of user modelling and, especially, the interaction between the user model (and thus the dimensions of the user to be modelled) and other models (e.g., the model of the car engine and of its behaviour) and to integrate aspects of reasoning about the user inside 'traditional' model-based reasoning.

The expected results of these research is the capability of integrated modelling of systems and of their (active or passive) users to achieve personalized forms of problem solving (e.g., monitoring, control, diagnosis) on the systems themselves. For example, in the automotive domain, the capability of taking into account the user in the design of a personalised car and then in the phases of controlling and monitoring and optimising the car behaviour.

2.2.5 Modelling the External World (and Reasoning on World Conditions)

Different aspects of the external world (context of application) could be taken into account in model-based problem solving and thus need to be modelled.

For example:

- In the automotive domain, the context of use of the car is relevant for various tasks (e.g., monitoring, control and diagnosis). This contextual information may include: the road conditions, terrain knowledge, weather conditions, etc.
- In biomedical systems, external factors such as the climate or other environmental factors may be relevant.
- In education system, the social context of a class or group of people may be relevant for defining learning strategies.

As in the case of user modelling, some research has been carried out in other communities, although the external context of the system has been explicitly taken into account in modelling less frequently, and thus the field is less mature. Model-based systems applications need special forms of context modelling to be integrated with models of the system (the main subject of problem solving). Specialised forms of reasoning about external context, integrated with 'traditional' model based reasoning and problem solving will achieve improved results.

Thus, in this case further research is needed, especially in the context of model-based systems, to understand which aspects of the external world need to be modelled, the way they need to be modelled, and the way they may be used in MBS&QR problem solving.

We expect that taking these aspects into account will produce a significant improvement in the problem solving capabilities of several applications of model-based reasoning. For example, the model-based control of a car may make decisions about optimal strategies (e.g., for controlling the gear box or the ignition), based on the engine status and behaviour, but also taking into account external condition such as the terrain. For example it would be useful to go to an lower gear just before a steep incline / decline or just before a dangerous bend, or be aware of the presence of a red traffic light just ahead of the car. Thus these forms of reasoning may lead to improved performance as regards for example, safety or environmental impact. We thus foresee intelligently managed systems, where the intelligence comes from reasoning on the system model but also, and in an integrated way, on a model of the external world in which the system is used (as well as on a model of the user of the system).

2.2.6 Organisational Modelling

Model based systems have concentrated mostly on the application to artefacts (e.g., electronic, automotive or aerospace systems) or biomedical systems (e.g., medicine, biological systems, etc.).

On the other hand, other areas of applications seem to be promising. In particular, it may be interesting to apply model-based reasoning to problem solving on different types of organisation, such as social, economic organisation or enterprises.

Some very preliminary attempts in these directions have been made in the past, e.g., for modelling economic systems and indeed they demonstrated interesting potential. For example, given models of economics, one may apply model-based problem solving to specific economic organisations to make predictions or interpret situations or monitor and control phenomena. Similarly, one may apply these techniques to social or socio-economic domains and processes.

I will require a lot of research in order to understand which dimensions of these domains can and need be modelled, also in relation to the different tasks that can be performed. This will require the definition of new ontologies and languages for modelling in these domains.

We envision a new area of model-based systems which will support problem solving and decision making in these areas and which thus may be used by managers and decision makers in different types of organizations (enterprises, communities, etc).

2.2.7 QR Methods Using More Sophisticated Mathematics

What the Problem is

In many cases, methods from model-based systems and qualitative reasoning build upon existing mathematical methods from calculus (e.g. differential equations), algebra (equations, functions and sets), and logic. The basic methods are geared towards the area of model-based systems and qualitative reasoning: (1) by restricting the domains and co-domains of functions to be discrete, possibly ordered, instead of being continuous, and the results are then still consistent with the underlying axioms, (2) by adding task-specific problem solving methods, such as methods for diagnosis, which are able to act on particular representations in a particular fashion. There are many mathematical methods which are restricted in their practical usefulness as qualitative versions of those are as yet not available.

Examples:

- in the area of uncertainty reasoning, where if data are not available or scarce, it is impossible to quantify a probability distribution reliably (this is because you can still use subjective estimates)
- in economics, structural equation models expect quantitative information, and the present state of qualitative mathematics is unable to deal with them

State of the Art

The theory underlying current model-based and qualitative reasoning systems is already firmly grounded on logic (e.g. logical abduction in abductive diagnosis and consistency checking and assumption-based reasoning in consistency-based diagnosis), algebra and calculus (e.g. QSIM). However, there are many open ends in this work, as the expressive power of the languages used for modelling is often restricted. For example, even though it is possible to provide a formal specification of dynamic behaviour in a qualitative fashion, it is as yet not possible to specify the uncertainties regarding these behaviours qualitatively in a probabilistic framework, as the mathematics to do so is still not powerful enough. The work going on in constraint logic programming can be taken as a starting point. With the exception of situations where uncertainty is involved where qualitative probabilistic networks can be taken. In both areas much progress has been made in the last three years, this process can usually be exploited in MBS & QR.

What Further Research is Needed

- The expressive power of many qualitative reasoning methods need to be extended in order to allow tackling a wider range of problems
- Ad hoc approaches in use due to the unavailability of mathematically sound approaches should be identified, and progress made in others areas should be incorporated into the field

- Mathematical methods allowing mixing qualitative and quantitative approaches within the same axiomatic framework should be developed

One consequence of this would be that in actual modelling of a problem, no choice has to be made between using either quantitative or qualitative methods.

2.3 Model Integration

2.3.1 Hybrid Modelling

Different modelling techniques allow the capture of different aspects of the same phenomenon. Hence, in order to include in one model different aspects of the same phenomenon or even different phenomena, you need to integrate models from different sources.

Examples:

- Pure qualitative models allow one to focus on significant behaviours, while pure numerical models allow one to detail each one of these behaviours or even to solve ambiguities related to the qualitative reasoning.
- Causal models and models based on quantitative differential equations provide two different views of the same phenomenon.

Few systems are capable of combining different modelling approaches. Currently, the main research effort is devoted to producing and to using semi-qualitative models.

A basic research issue in the QR community for the next decade is the mathematical aspects of the different kinds of models and their inter-relationships. In the future, model-based systems need to be able to combine different types of models to solve a given problem. The target to be achieved might be the kind of reasoning displayed by human experts, who seem able to combine information from different types of models seamlessly, and to combine information from partial models of each type.

2.3.2 Integration of Models from Different Domains: Electrical, Mechanical, Hydraulic, etc.

In many situations it is necessary to consider phenomena with different natures in order to reason about a system.

Examples:

- In the field of continuous industrial processes, many devices, such as pumps, comprise phenomena related to hydraulics and mechanics.
- In the automotive industry cars comprises different inter-related subsystems such as hydraulic, electric and electronic.

The integration of models is an open problem, and further research on this topic is closely related to research on ontologies. Some work has been done using domain independent ways of modelling, such as bond graphs, although that work has not been as successful as might have been expected. One of the reasons may be that bond graphs are well suited to simulation, but less adapted for the other tasks performed by model-based systems.

It may be that a combination of appropriate methodologies for individual domains, plus standards (see Section 3.5) for interfacing models in different domains may finesse this problem, but at present it is still an open problem.

2.3.3 Multi-level Modelling

It is necessary to combine models at different levels of abstraction to solve a particular problem; usually to cope with complexity.

Examples:

- In food industry, the evaporation station can be modelled, at least, at three different levels: simple mass balances (product conservation), detailed balances (mass and energy conservation) and detailed dynamical model for control purposes.
- In computer industry, the paradigmatic example by Davis: a computer can be seen at different levels, from high-level functional components to chips.

Currently, there are different theoretical proposals: automated handling of diagnosis hypotheses, multiple models considering available time for diagnosis, multiple levels of abstraction regarding the quality of the diagnosis. However, there is almost no application working on industrial systems capable of handling models at different levels of abstraction, because there is no systematic way to share results from different models within the same task.

A real applicable methodology needs to be proposed to change smoothly from one level to another, exploiting results from different levels.

Eventually the reasoning system should be able to select the adequate level of abstraction automatically, and to switch from one to another as required.

2.3.4 Combining Qualitative and Functional Models

Much qualitative research has concentrated solely on reasoning about the structure and behaviour of systems. For many applications, it is necessary to abstract the results in terms of the function or teleology of the system. That implies being able to represent teleological knowledge, to reason about it, and to map behavioural knowledge to it. This has been done for systems with fairly static behaviour. That work needs to be extended to cover complex, dynamic time varying functions.

3 Processes for Model-based Engineering

3.1 Introduction

As model-based reasoning increasingly enters the industrial arena, issues concerning the long term commercial use of models are becoming more important. Some of those issues are in common with any kind of software, and are covered by general software engineering considerations. However some of them are more specific to models and modelling. This section deals with:

- Model Generation. How do we build appropriate models in a cost effective way?
- Model Conversion. Where models already exist for other purposes; how can that information be reused for model based reasoning?

- Model Management. How can a company keep track of what models it has, the ways those models should be used, and when a specific version of a model is the right one?
- Standards. At present, most people using model-based reasoning develop their own model descriptions. As models are used for more purposes, common standards for models will need to address a number of issues, so that models can be exchanged between component suppliers and system builders. These kinds of modelling issues are not addressed sufficiently at present by existing standards such as STEP.

3.2 Model Generation

Each attempt to build a model-based problem solver for some industrial application faces the problem of creating the appropriate model. It is not sufficient to merely deliver such a model, it is important that there are methodologies and tools that allow us to calculate and restrict the costs for model generation. Moreover, it is also not sufficient to develop arbitrary methodologies and tools, it is essential that they are related to the current practice, education, work processes, and tools in the respective engineering domain. That means considering what data and knowledge is already available in the domain, and use it as efficiently as possible when creating models.

3.2.1 Automated Model Generation from Simulation Models

The exploitation of model-based systems in industry will greatly depend on the (additional) modelling efforts they require. This led us to the attempt of reducing these efforts by automated conversion of existing simulation models into abstract models suited for model-based problem solvers.

Simulation models of a system are often created for control purposes. However, for diagnosis, for example, specific properties are needed from a model:

- *Needed: a component-oriented model.* For achieving the simulation of the system behaviour, the component structure of the respective device is fairly irrelevant. As a result, the subsystem structure of the model does not necessarily reflect the component structure of the device. For instance, a certain pipe might not occur at all in this model. But if diagnosis has to consider the possibility of a leakage or clogging, the component has to represented and modelled.
- *Needed: Preservation of the physical structure.* A typical example of a violation of this requirement is that input and output flow of an aggregate device might be identified by an equation which, again, is based on the assumption of normal behaviour (no leakage occurring).
- *Needed: models of faulty behaviour.* These are required if we are not only interested in fault detection, but fault (class) identification (as in on-board diagnostics), diagnosability analysis, and FMEA. As long as ‘control’ is considered as ‘controlling the device under normal conditions’, faults are not considered in the development of the control algorithms. As a result, they are not part of the respective simulation model. Extending this to include fault models is not always trivial. If faults correspond to deviating parameters, it is fairly straightforward, but in the general case, faults may change the structure of the model radically. For instance, introducing a pipe with the potential to have a leakage means introducing another state variable and affecting the possibility to simulate the model.
- *Needed: a physically correct simulation model.* Since the diagnosis approach is based on identifying discrepancies between a certain behaviour mode (OK mode or

some fault) and the respective model, it is crucial that the real physical behaviour is actually covered by (the envelope of) the model. If this is not the case, e.g. because the error functions ε-, ε+ are difficult to estimate, diagnosis runs the risk of detecting model faults rather than component faults and, hence, of generating wrong diagnoses. While we may assume that the normal behaviour is properly covered if the model satisfies the needs of control, it also has to correctly model the behaviour if a fault occurs. In many cases, the models of components are based on an implicit assumption about overall normal behaviour. This may be addressed by an appropriate modelling methodology. However, there is a serious limitation: in particular for complex components, we may lack first principles models, and the simulation model contains characteristic maps that contain empirical data. In this case, the conditions under which these date were obtained (typically normal conditions) are compiled into the model in a way that is hard or impossible to detect.

We should note that only a few of these difficulties really stem from an inappropriate modelling process or modelling faults. Rather, it is the purpose of the simulation models, namely simulating correct behaviour for control purposes that is in conflict with the diagnostic requirements. Without integrating the views and the work processes concerning system development for control and diagnosis, this will be difficult to change.

There is a need for methodologies that ensure that the models are built correctly in the first place for use in other tasks that simulation, and techniques that make the process of converting those models to ones appropriate for diagnosis and other tasks is a painless process.

3.2.2 Derivation of Qualitative Models from Requirements

During the design process, the correct operation of a system is often described at a high level, perhaps in terms of state charts. Such information is often very useful when performing model-based reasoning. Better methods are needed of integrating it into the construction of model-based systems.

3.2.3 Automated Modelling

Automated model building and model transformation needs continued theoretical work and more effective and efficient algorithms. This is emphasised by application requirements. Much of the expected gain depends on fast and economic creation of models from a library. Since different tasks may require models at different levels of abstraction, there is a tension between the desired compositionality and generality (and, hence reusability) of models and the necessity of task-oriented models. QR needs to develop techniques to generate task-oriented models from generic ones.

This also touches upon a more general goal, namely integrating QR results and techniques with standard engineering practice and tools. The lack of integration presents a major obstacle to transferring QR technologies into industrial practice. Deriving qualitative models from numerical ones that have been developed, for instance, in the phase of design verification, is of high practical importance. However, it may require changes in current modelling practice towards modular, component-oriented models. The need to blend in with current practice also applies to other domains, such as medicine, economy, biology and ecology.

3.2.4 Model-based System Identification

Model building is a difficult and time consuming process. A much more efficient alternative to building models by hand would be to learn models from observed data. This is still a very difficult machine learning challenge for complex domains. Qualitative reasoning can help with this in two ways.

> *Learning qualitative models*. In domains such as some areas of biology, where the underlying models may not be known, it will be possible to learn qualitative models from data. Early research in this area indicates that it is more useful to build qualitative models rather than numerical models at this stage, in order to facilitate understanding by domain experts.

> *Deriving quantitative models from qualitative models*. In domains where qualitative models are known, but are not executable, qualitative reasoning provides graphical ways of building executable models, and makes clear the assumptions behind the models, enabling domain experts to compare their models on a like-for-like basis. Where models are known and data is available, it should make it possible to develop accurate numerical models with known assumptions and limitations.

The main impact of these techniques may well be in science rather than in engineering, providing tools for scientists to understand the world better, and having a dramatic impact on the way that we carry out scientific research.

3.3 Model Conversion

3.3.1 Conversion of Qualitative Models

One issue concerns the development of better engineered and easy-to-use tools that facilitate the exchange of results among researchers and make QR techniques available to potential users in other areas and application work. The field, so far, has developed a variety of theories, formalisms, and techniques with different degrees of generality and is still far from delivering a small set of uniform principles and systems. If the field can make progress on this, it will become easier to create and exchange libraries of reusable models.

3.3.2 Conversion of Design Models

Section 3.2.1 has discussed the need for closer links between simulation models and model-based reasoning models capable of a range of problem solving tasks.

3.3.3 Automatic Model Selection

When performing a range of reasoning tasks, it is unlikely that ‘one size fits all’ modelling can be achieved. What is needed will be a range of models, perhaps sharing some features and information, with the ability to move between models seamlessly, in the way that a human expert might do. We need to develop clearer representation techniques, better methods for showing the relationships between different types of models, and robust techniques for reasoning with a range of models.

3.4 Model Management

3.4.1 Model Maintenance

Like many new technologies, model-based reasoning needs to impose the lessons learned from other fields of engineering. In particular, it is necessary to be able to manage different versions of a model, and know which versions make up a specific simulation. The integration of configuration management tools into model-based reasoning will become important for safety critical applications, and has not been properly addressed.

3.4.2 Model Warehouse to Allow Knowledge of the Variants Present for Diagnosis

One specific issue for model-based diagnosis is the management of models for doing diagnosis of different variants of a vehicle. This involves integration of model-based diagnosis with the other data warehousing facilities of an organisation, so that all appropriate modelling information is extracted from the warehouse automatically in order to produce the correct diagnostic / design information for the variant.

3.4.3 Modelling Tools and Environments (to Support the Whole Lifecycle)

The average expected lifespan of a vehicle is 8 to 12 years. When the design time and extended support of long-lived vehicles is added in, the lifespan of a typical model is nearer to 30 years. A manufacturer needs to support information about a vehicle across that kind of span of time. If model-based reasoning tools are available to support tasks throughout the vehicle's lifetime, from design through diagnosis to eventual recycling, then there is a need for environments which are capable of supporting the modelling and reasoning tasks through that length of time. Some of the challenges are in the model warehouse area outlined above; others involve developing and supporting generic model structures that will be able to support modelling as the nature of vehicles changes over such a long period of time.

3.5 Standards

3.5.1 Standards for Modelling Tools (for Knowledge Management of Model Libraries)

What the Problem is

With the increasing availability of tools that assist in the development of deployment of model-based systems, it becomes necessary to be able to exchange models and modelling templates between various tools from different tool vendors. It may also be necessary to actually import model-pattern libraries into a software tool.

State of the Art

Current model-based and qualitative reasoning tools are unable to take advantage of what exists in the area of model-based systems and qualitative reasoning in general, as each system has its own representation formats, model library, and often specific choice of problem-solving methods.

What Further Research is Needed

Research into the principles underlying model libraries and what should actually be in model libraries is needed:

- General patterns of models that can be used in model-pattern libraries
- Shared formats (e.g. based on XML) for model libraries
- Techniques for investigating the consistency of a model library
- Techniques for integrating model libraries

3.5.2 Standards for Model Representation and Model Meta-knowledge

What the Problem is

The area of model-based systems and qualitative reasoning distinguishes many different types of model, and various software tools offer support for the development of specific types of model. However, even when considering the same type of model, model representation will vary among different software tools. In order to be able to transfer models that have been developed to the user community it is necessary that a common format is supported by all the software tool vendors. This does include the possibility to transfer meta-knowledge about models that have been developed between various tools.

State of the Art

Available software tools have their own formats, although increasingly based on XML-based formats. The capability of transferring models, i.e. meta-knowledge about models, has so far not been an issue for software developers.

What Further Research is Needed

A shared representation formalism with fixed syntax and semantics for both models and meta-knowledge about models needs to be developed. This may include the need for special purpose parsers and checkers that can be exploited in tools already available.

Final Capability

Integration with versioning software and other software, e.g. for visualisation.

3.5.3 Standards for Diagnostic Knowledge Representation and Distributed Mitigation Protocols

What the Problem is

Many different methods for developing diagnostic models exist, varying from qualitative methods from consistency-based diagnosis, abductive diagnosis, FDI to probabilistic methods used by the Bayesian network community and adaptive system community.

State of the Art

Various theoretical frameworks for the construction of diagnostic systems have been proposed in the literature, based on logic, set theory, and probability theory, and partial tool support for parts of those frameworks have been developed within research projects. However, this work has not given rise to the development of standards in the area. The development of standards for diagnostic models is important if diagnostic models are to be used within industry.

4 Task Oriented Challenges

4.1 Integration with Problem Solving

4.1.1 Hybrid Reasoning Systems (Integration of Different Reasoning Technologies)

Different aspects of a given task can be solved more efficiently using different reasoning techniques, such as model-based reasoning, case-based reasoning or knowledge-based reasoning. This fact is even clearer when the reasoning system has more than one task at hand, for instance diagnosis together with monitoring and/or re-configuration.

Examples:

- Diagnosis in industrial processes can be done knowledge-based, model-based or combining results from both.
- Reasoning about socio-economic systems usually requires the combination of quantitative, qualitative and experience-based models.

Few systems had used hybrid solutions in the past to solve an specific task. Nevertheless, currently there is an increasing interest in this subject in different research communities, not just in MBR. Nowadays research is being done in combining model-based and knowledge based systems, knowledge-based and machine learning, and model-based and machine learning among others. Unfortunately, there is no general proposal yet, merely a collection of *ad hoc* technological solutions.

Additional research needs to be carried out to bring a formal framework for sharing knowledge among different reasoning technologies. For instance, results provided by a neural classifier should be smoothly interpreted to select an appropriate level of abstraction in a model-based reasoning system. In the long term, these issues could be answered by means of a General Knowledge Theory. Nonetheless, this seems to be a basic research issue for years to come. Hence, current and future research should provide different technological proposals which should be tested in different contexts.

Finally, reasoning systems should be capable of automatically deciding the best technique to solve a given problem.

4.1.2 Integration with Planning, Re-configuration, State Assessment and Other Tasks

Increasing complexity (in size, behaviour and/or configuration) and continuous operation demands in real applications has caused new requirements on reasoning systems performance, which must now run simultaneously more than one task to accomplish these goals. Generally speaking, solving complex problems eventually requires the collaborative work among different sub-tasks. However, finding the set of lower-level sub-tasks is just a part of the problem. It should be stated how each task could improve its own performance through collaborative work with other tasks.

For instance:

- Diagnosis systems running in satellites must be in continuous operation, adapting themselves to their own results. In such sense, they must work together with re-configuration or planning systems to avoid the effect of the failure.
- Another example can be found in many industrial processes, where the whole factory must be commanded with different protocols according to the different states the

factory could be in. In this case, a state assessment task should inform the monitoring task what the current set of working conditions are. This information would produce changes in the models used by the monitoring task.

Currently, many research communities are approaching this problem with different perspectives. In fact, several taxonomies have been proposed to characterise the set of sub-tasks involved in solving a more complex task. Nevertheless, there is still work to be done in that area.

There should be a complete and exhaustive study of the dimensions within each sub-task, and their functionalities should be defined without considering the domain in which they are to be applied. Moreover, this research topic is closely related to the one mentioned above (hybrid reasoning systems), since each sub-task could be approached in different ways through different methodologies. In our opinion, this kind of research should be carried out by multi-disciplinary groups with different backgrounds on technical, social or educational issues.

Eventually, model-based systems (in general, reasoning systems) should be able to perform high-level tasks by smoothly integrating results from different sub-tasks. The final goal would be to maintain along time the desired functionality of the application.

4.1.3 Learning as diagnosis

Some interesting work has been done on using the well developed model-based reasoning methods in diagnosis in order to assist learners. Essentially, a model is kept of what the learner understands, and what they need to understand. Problems with the learner's model are diagnosed, and learning exchanges are entered into in order to try to fix the errors.

Improvements in this area depend on several factors:

- Improved models of user understanding.
- Better diagnosis of those models.
- Increased understanding of how to repair discrepancies between the user's model and the ‘ideal’ model.

This type of work is at an early stage, but is significant, not least because of its potential for supporting e-learning.

4.1.4 Interfacing Model-based Reasoning to the World

This issue was highlighted by the Automotive Roadmap, where it called for ‘reliable, efficient and inexpensive bi-directional communication with the vehicle’. However, the issue is a wider one. Many applications depend on information or data from outside of the model based system in order to be able to perform reasoning. Linking from outside data to the model-based system is often presently done in an ad hoc method.

What is needed is operational abstractions of what is being done. There has been some work in this area, but further research is needed, both to formalise the kinds of techniques that have been employed, and in order to make such integration a painless process.

5 Further work

The MONET2 project has a further version of the Roadmap planned. Some issues have emerged from this version of the Roadmap which there has not been enough time to explore in detail.

Higher level cross-domain technological issues emerge from consideration of the individual Task Group roadmaps. This section lists those issues for further thought and discussion within the Task Groups.

- Higher levels of autonomy: Model-based systems increasingly need to be able to reason about control systems, and to make decisions in the absence of human input.

- Tool support within a work process. The present applications of model-based technology implement tools that provide partial solutions for specific tasks. In the future, they will need to be incorporated into complex work processes.

- Overcoming barriers to industrial progress. Full acceptance of model-based tools in industry will need improved interfacing with users. For example in engineering, it will mean building systems that communicate in engineering terms, that conform to the users' way of thinking, that also work with other tools in that domain.

It is hoped that the importance of these themes can be explored in the next version of the Roadmap.

6 Appendix A - Automotive Roadmap

7 Appendix B - BRIDGE Diagnostic Roadmap

8 Appendix C - Education Roadmap

9 Appendix D – Bio-Medical Roadmap

10 References

N/A

11 Document History

Version	*Date*	*Changes made to document*	*Changed by*
1.1	30th June 2003	Co-ordinating Node and Task Groups Produce Document	MONET
1.2	3rd October 2003	Updated with Standard format, text unchanged so release date left at June 03	RIR