MONET2

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

Contract: Concerted Action / Thematic Network

Contract No: IST-33540

Deliverable MBD7:

Fault Detection and Diagnosis (BRIDGE) Technological Roadmap (Version Two)

Date: 18th November 2004
Version: 4.2
Status: Release

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

Contents

Authors

Joaquim Armengol	University of Girona
Marie-Odile Cordier	Institut National de Recherche'en Informatique et en Automatique Unite de Recherche IRISA
Olivier DeMouzon,	LAAS-CNRS
Teresa Escobet	Uni Poly Catalonia
Marc le Goc	LSIS-CNRS
Anne-Lise Gehin	Lille University
Carlos Alonso Gonzales,	Valladolid University
Micheal Hofbaur	Graz University of Technology
Robert Milne	Intelligent Applications
Xavier Olive	LAAS-CNRS
Belarmino Pulido	Valladolid University
Benny Raphael	EPFL
Iain Russell	MONET Project
Sandro Saita	EPFL
Peter Struss	O'CCM GmbH and TUM
Janet Thomas	MONET Project
Louise Travé-Massuyès	LAAS-CNRS
Cecillia Zanni	LSIS-CNRS
Marina Zanella	Universita degli Studi di Brescia

1 Introduction

The Bridge Task Group is aimed at further developing, relating, or converging the various approaches of DX (Diagnosis - Computer Science / Artificial Intelligence field) and FDI (Fault Detection & Identification - Engineering / Control field) to the Diagnosis of technical systems, in combination with participation in relevant conferences and industrial participation. The Task Group has defined a set of problems extracted from industrial concerns supplied by industrial participants, and works by analysing the application of the respective technologies to the formation of solutions to industrial problems, thereby assessing their strengths, weaknesses and also their general complementarities.

The Task Group has the following objectives;

- To support the transfer of Model-based Diagnosis (MBD) technology into industry related disciplines.

- Through industrial participation, to help identify for participants from different backgrounds, the requirements of the diagnostic processes in industry, and to relate the existing theories, methods, techniques and systems to these requirements, by characterising the possible solutions for a given diagnosis problem.

- To determine and define problems extracted from industrial challenges, and to use these as tests for measuring and comparing the value and the results of applying various techniques or combinations thereof in the search for solutions. The accurate characterisation of diagnosis problems and solutions is, then, crucial.

- To compare and characterise the presumptions and scope of applicability of MBD technologies developed in AI (DX) and control theory (FDI).

- To identify competing and complementary methods and solutions using the DX MBD technology alongside the FDI MBD technology.

- To identify and define useful interfaces and protocols for beneficial combinations of techniques coming form these two fields.

This Roadmap is aimed both at producing what the Task Group feels is an accurate vision of the evolution of Diagnosis in the real world over the next ten years and also a vision of what researchers will be doing and what they would like their students to be achieving over the same timeframe.

The Roadmap will also weigh the vision of ‘Diagnosis’ as a whole but will utilize the vision of specific application domains where necessary. If this document were to focus on only one specific domain we would be able to show the impact these technologies would have on the everyday lives of citizens, however the Roadmap would then be very different for each application domain. Therefore this Roadmap is focused at Diagnosis as a whole and the convergence of DX and FDI techniques specifically, and will include application specific examples where they assist in the clarity of the document.

2 Present Situation of Technologies

Technologies, such as those used in DX or FDI, have always faced issues when it comes to the realisation of real world applications.

End Users often do not have a clear understanding of the technologies[1]. Hence, it should be our responsibility to ease the transfer from our research field to 'Technology Supply Companies'. At this time, and in the Bridge context, it is clear that FDI techniques and methodologies are more established and are better known by engineers than their DX equivalents. Hence, one of the main long term goals of Bridge should be to provide those companies with prototypes / products combining both technologies.

This situation is also a consequence of the current state of the research world; where there could be conflicting, or even opposing, ideas or understanding of the same issues. It has been the task of Bridge to attempt to bring these people and ideas together in order to find common ground, although it has struggled against the lack of momentum from the FDI community to accept DX technologies. This is, however, a situation that can often occur when researchers focus closely on their own particular fields of expertise and do not work in wide collaboration with researchers from different backgrounds. This degree of research specificity is required for important aspects of complex problem solving, however ,certain approaches can lead to better results that others. However, the critical importance of bringing different, but associated, research communities together can be seen when there are issues in one community that could be better solved by the methods in another. Examples do exist which can clearly demonstrate that this is in fact the case, for example:

- The successful introduction of fuzzy logic or neural networks in control technology
- The use of statistical techniques in FDI
- The use of dynamic systems theory results in qualitative reasoning, etc

Fortunately, there are now more researchers who are aware of the variety of available diagnostic techniques and it is Bridge's intention to ensure that this spectrum of techniques should be applied more often in the future. Many researchers in the DX field already benefit from the utilisation of some FDI technologies but the reverse is not generally true.

Another primary challenge which the MBD (and any other diagnostic) community is facing, is that the complexity of modern systems is increasing vastly quicker than its current ability to model these systems. One source of complexity is that there are systems with a large number of components / distributed systems, etc. Although there are current efforts devoted to these issues (works based on agents, or distributed aspects of Diagnosis), additional research is needed. Another issue is that larger systems will have more kinds of uncertainty, noise and unknown disturbances related to model building issues which exist in both FDI and DX. If these issues are to be solved, a combined (hybridised) approach must be adopted. It has been shown that one technology can be better suited for a given kind of problem than another, and vice versa. For instance, FDI is more suited and has larger experience of fault detection than fault localisation, while in DX fault detection is usually taken for granted and emphasis is placed upon fault localisation. Hence, many Diagnosis applications would require the

---

[1] It must be noted that within the field the term 'Users' is split into two distinct groups. Business / End Users are the people who utilize the applications that are produced using this technology. It is the suppliers of the applications, the 'Technology Supply Companies', who are the ones who make the transfer of the technology from development to product and it is these companies to whom the 'technology' itself must be justified. The Business / End Users often do not care, or usually even know, what technology underlies the products they are using.

complementary use of both approaches. Nevertheless, in order to share results and techniques, there should be a common framework and a common understanding. This is another major goal of Bridge.

Both FDI & DX techniques have restrictions when facing real, complex problems and thus have to make assumptions (single fault, non-compensating faults, etc.). This is less of an issue in DX, which means there is the potential for FDI to benefit from this approach. However, if we are to demonstrate the compatibility of these techniques, there is a need for a product(s) that will show that DX and FDI technologies can be combined and applied in real world applications. Identifying what problems exist on the road to generating this application would lend considerable weight to the efforts to combine the technologies.

Another factor that would be of considerable value would be to show how Model-based Reasoning (MBR) can benefit from complementary techniques or methods in order to cope with Diagnosis of complex systems. It is also important that the techniques or methods ‘above’ MBR are able to extract relevant pieces of information, data, parts of models, etc. and fully understand and interpret them. However, it is likely that when Diagnosis is difficult, it will always be difficult and that there will always be some systems which are too complex and need to be diagnosed using simplified models. It is very hard to say if this will always be the case or not.

Summarising, Bridge’s aims have been to provide a better understanding of each other for both DX and FDI communities; then bring in a common framework which would allow future sharing of the best from both worlds (results, methods and techniques); and helping in the transfer of these results, with new products (for instance, ToolKit Products) that can smoothly combine both approaches.

3 Roadmap - Drivers

Originally this section (and the Roadmap as a whole) was focused on developing the Bridge between the FDI and DX Communities. It is the intention of the Task Group, and this version of the document, to broaden that focus to encompass Diagnosis as a whole.

That said, a major part of the work of the Bridge Task Group does include ‘bridging the gaps’ between the FDI and DX communities. In order to do this, as clear a picture of these gaps as possible must be drawn up, but a clearer view of the Diagnosis community as a whole will aid this task greatly, and therefore to ensure that the Technological Roadmap can successfully assist with this task it must be more general in application. To do this the Roadmap must concern itself with the whole technology of Diagnosis and not just with one particular area. The Bridge Task Group can then utilise it to forward the aims of the Diagnostic community as a whole by using the best of both techniques in the future under a common framework.

Initially, however, the Roadmap will examine the State of the Art in the sector and draw upon the current position of the Technologies in order to attempt to predict future trends. It should be considered that within the Diagnostic community, FDI is more established than DX. Nevertheless, we cannot be certain as to whether there will be a time when the two are at the same stage of development or whether FDI will continue to develop at the same pace and will therefore always be ahead of DX.

The technological aspects of the field can be divided into four main issues:

- Modelling Issues

- Temporal issues
- Diagnosis and Product Life Cycle
- Autonomy versus human interface in applications

In the first part of the document, the vision is assumed and we have worked towards outlining the present limitations of our technology. This is then projected in time, in the second part of the document (tables), to obtain a scheduled vision of the research and tasks that will be undertaken.

To aid in the clarity of this next part of the Roadmap, bullet points have been used at the start of each section to give a clear indication of the topics that will be covered in that section. The text then goes on to discuss these points in finer detail.

3.1 Modelling Issues

- Modelling is difficult - need automated modelling tools
- Take advantage of design or existing models
- Few model libraries exist
- Every Application calls for specifically tailored modules

Modelling is obviously the core of Model-based Diagnosis systems and hence of prime importance.

The importance of domain libraries has been a topic of discussion for many years now and some experts question if the lack of generic model libraries is actually holding back the development of the technology or not. Currently very few domain libraries exist and the ones that do exist only in a small number of application domains. In the near future however, more domain libraries should be available. But it is the opinion of the Task Group that the lack of this resource has not had a major negative effect on the transfer of Diagnosis technologies into the industrial sector, primarily because industry uses very different models to the ones used in DX research. What we require to achieve better industrial technology transfer is a set of core models that are actually used by industry rather than a general library of models or model fragments. What might prove more useful again is to investigate how we can help industry to improve the models that they already have and that they are used to (even if just a little) to ensure that they are more usable and not just for their intended diagnosis purpose. If we continue to develop 'our' models 'our' way then we risk becoming distanced from what industry is actually doing and this will result in the bridge between research and industry becoming harder to achieve rather than easier. A side issue to this is to focus our technology transfer efforts towards what industry is 'used to'. If we can show them a better way to apply their technologies that they already utilize they will be more willing to discuss this rather than us having to 'sell', what is for them, a completely foreign concept.

One problem with the way Model-based Diagnosis is seen by the industry is that they believe that it demands more effort to operate in comparison with data driven techniques due to the continuing requirement to update the model. In fact, data driven approaches have the same problem because data becomes old and out of date quite soon. However, industry still appears to be of the opinion that updating a model is considerably more time consuming than updating data. It would considerably benefit our Bridging activities if we could show this is not the case, for instance providing tools to automate this task.

This issue is very closely related to automated modelling and a considerable effort should also be devoted to improving this work. Automated modelling diminishes the dependency on human experts paving the way to solve all the variants of a problem instead of a specific problem alone. Over the last five years, industry has done a great deal of their work fully automatically (using CAD Tools etc.). This means that they now have a large investment in modelling, modelling techniques and designs in electronic form and these standards are very firmly integrated into their working practices. We therefore have to fit in with the industry perspective and not work independently. For this trend to continue successfully there would also be a need to combine models from both MBR and other technological approaches such as machine learning.

More recently, industry has been beginning to understand the potential benefit of considering diagnosability and maintainability at the design stage. This means that the models are created at the design stage and are then used throughout the product life cycle. This has a major impact on Bridge activities, because if we can build a bridge between the design department and the maintenance department (for example) the models will be shared more easily. This can not happen unless the same models are used by different stages of the lifecycle, i.e. by different departments.

Some areas of industry are also attempting to define Standards in modelling (modelling languages and techniques, object orientated modelling, etc.) and it may be useful for us to ‘piggy-back’ our ideas onto this. Some attempts at industry standards have been made, for example the IDD Project [1] investigated industrial transfer but their results are not particularly applicable to our situation. They worked on generating qualitative models from Simulink models and outputs in a form that could be used by a Model-based Diagnosis Engine. For our purposes it may be of use to cross link this with the work currently underway in the machine learning community on extracting symbolic (qualitative) models from numerical ones. One of the major issues of both of these approaches is the difficulty in generating the thresholds of the qualitative model, in other words, discriminating between the data which corresponds to normal and the data which corresponds to a fault.

This point only goes to emphasize that there is no way in which we can ignore what tools are used in industry and so we need to orient our work towards them, possibly focusing on modelling languages that are becoming more widely used in industry, for example industry foundation classes (IFC).

Industry is also very segregated (designers, architects, etc.), so we must watch the language that we use and focus this on the target audience. Ontology will help us to understand these links and identify what can be translated from one world to another. We do understand that it may not be practical to have a single language for all areas, and even if it is, it would probably only be applicable to a particular part of any project. There have been attempts to automatically translate one modelling language to another but these have not been very successful in practical applications. Even if translation is not possible it may be useful to have a system that shows if models are still compatible, i.e. if there are several departments working on the same (or a similar) system and one department changes something in their plan, is the model still compatible with the model in the department next door.

Another possible way to achieve the same goal is to take the MATLAB / Simulink models, etc. that are being used in industry and make them more transparent and usable. This, at least partly, feeds back to us teaching students to build models in such a way that they can be easily converted, are re-usable and are also clear to other people who may be using them.

‘Model-based Design’ as a concept has a different understanding in industry than it does in research. In our case the model is the basis by which we programme the system, whereas industry sees it as the basis by which the design engineer does Diagnosis in a ‘hands-on’ manner. We could demonstrate the benefit of our approach by showing them what can be achieved if they use the model as more than just as a tool to build the simulation. They may realise that if they have a good model from the start and automate everything else, this would save them a considerable amount of time. This time saving may leverage the belief that they should work on good models, write them well and integrate them with all the other stages of the process.

However, one factor which works against this is the time that is required to create good models. For example, it might take only half a day for a skilled architect to make a design drawing that has sufficient details for a builder to construct a garage. It may then take two or three weeks, however, to create a complete three dimensional model of that same garage. The question we face is then, why should someone spend three weeks building a model of a garage that can be drawn in three hours. In order to encourage designers to create electronic models, we must show added value later in lifecycle; maintenance and fault prevention etc. If the result of the work is merely construction then there is no need for a model. This is also why it is important that designers take a long-term view when they build an artefact (car / bridge / etc.) so that it is done in a way which reduces future effort. It is important for us to make industry aware of the fact that for operational Diagnosis it is often necessary to have the original model so that we have a baseline with which to compare the current observations. If no such model exists then it is hard, if not impossible to state whether a variable is normal, within operating thresholds or at fault.

There are still several Diagnosis solutions which are based on simplifying assumptions (such as single fault or non-compensation assumptions) in order to be effective and / or efficient enough for real world applications.

Furthermore, the ever increasing complexity in systems pushes the need for new diagnostic solutions. For instance, there are problems associated with the distribution of Diagnosis in complex systems. FDI approaches the system as a whole, and therefore misses concepts so that tasks are shared. Current diagnostic technology only utilises one core part, to be in a position to work on distribution may take another five years.

3.1.1 Genericity of Models

- Generic model libraries or specific models?
- Need automated modelling tools.
- Reusable components versus system models.
- What level of abstraction in models?

There is some debate as to whether we will be producing more generic models in the future, or whether this is an almost impossible task and time and effort should not be wasted on it. This depends on what definition we have for a model and is affected by the way we describe the system, which depends on what you want to do with the model. Even when you have the system and you know what you want to do with the model, you still want to have a model that is as simple as possible. If you then want to use the same model for different purposes then the model must be more complex. This makes models difficult to understand and time consuming to work with.

One possible compromise is to produce a model which is re-usable for all of the components of one type, which is a central idea in object oriented modelling. For example a Simulink model has an input and an output. It is not possible to apply some signal to the output and observe the input. With object oriented modelling the goal is to restructure the model depending on what the input / output is at each stage.

This is not necessarily solely a question of Model-based Diagnosis, but a question of what models are available in industry and to what uses they are being put.

The basic premise is that the more abstract the model is, the more generic it will be and therefore the more re-usable. This may apply in principle but does not necessarily apply in practice. For example, even if a company supplies the model to be used for Diagnosis, you still have to go inside the model and ensure that the model will do what is expected, because models are usually built to understand the behaviour of the system (by simulation), not to diagnose it.

3.1.2 Multiple Models

- Many systems need models at different levels of abstraction
- Design models are not directly usable for Diagnosis
- Different goals may require different models

One way to cope with the complexity of systems is to produce models at different levels of abstraction, or multiple models of the same thing depending on the situation and level of detail. This must, however be viewed as the end result, currently industry will not be willing to commit the time / money to produce all of these different levels of models. When we experience difficulty in getting industry to build one complete model, to expect them to produce a series of models is unreasonable.

In the engineering world a model is often broken down to simply ‘input / output’. Within the car industry when the models are transferred, they are often ‘closed’ so they can be used but not opened. This can lead to a lack of any clear understanding as to what is a fault. Often when a driver arrives at a service garage and reports a problem, the garage does not know whether that is the way the system has been designed to operate or whether this is actually a fault. One example of this occurs with windscreen wipers. On one vehicle the windscreen wipers stop at a particular speed and drivers have been reporting this as a fault as they believed something was going wrong. However this was actually a design specification of which the garage was not aware and so when they investigate the problem they (rightly) could not find any faults. This could even have led to them changing the ECU, which is controlling this function, for a new one to solve the ‘fault’, but of course, the new ECU would perform the same way. This is also true of the telecommunications networks, as the operator who is monitoring the systems often does not know how the components are built and the companies who build the components do not need to diagnose problems.

This is an important driver for our technologies. When you get to the level of complex systems, they are built up from components that are constructed by different companies and no one person has a complete view of the operation of the system and all of its components. This is also a problem for people who are actually trying to construct models of complete systems because they may not have models of the components from which they are constructing their artefact. The model-based approach could usefully provide a method for aggregating this information together. Currently industry has no way to handle these problems, they merely isolate the ECU containing the fault signal, they are not capable of doing hierarchical Diagnosis.

3.1.3 Scope of our Models - Internal versus External Faults

- Do we need fault models or only a model of the correct system?
- We need the goals to construct the models

It is argued that for some approaches it is not necessary to make a distinction between internal and external faults. If you have a model that is basically a network of constraints, then it all comes down to whether or not you are able to characterise either the proper input or the proper output of the behaviour by the set of constraints. For consistency based Diagnosis, often, it does not matter what the fault actually is. It could be that 'the pressure is too high' or the 'valve does not open', it does not actually matter. It all comes down to what we would consider a 'wrong' or 'bad' input for the model.

The consistency based Diagnosis perspective suggests that for fault detection we start the Diagnosis from the point where we have a model of the correct behaviour and all of the components are working. Then what we have to do is to check for consistency with the observations. However, we must question if this is enough, after all what we actually want is to know whether the system (i.e. plan or device) that we are looking at is indeed fulfilling the purpose for which it was designed. The essential element of deciding whether something is going wrong is therefore that 'the device is not (fully) fulfilling its purpose', rather than 'there are no detectable faults'. This is a very recent but very important shift in Diagnosis.

Thus we need to take the model where all the components are operating correctly and then take the observations to see what the current state of the system is and then check whether this is consistent with the goals of the specifications of the behaviour. Previously the theory was, at least in component based Diagnosis, that if all the components work correctly then this means that all the goals are being met. Thus they are assuming that it is a well designed system and so if the components don't fail then the goals are achieved. However with modern Diagnosis we are required to make the goals explicit as the aforementioned belief does not necessarily hold true as we are aware of conditions where, even when all of the components are working, the goals are not fulfilled.

This gives us two sources for identifying a problem. Firstly, we have the model of the components which tells us that a component is faulty if it exceeds a certain specification or is working outside a certain set of parameters, or indeed not working at all. The second source is the specifications of the goals themselves and this means that it may even be impossible to satisfy the goals under certain conditions even if all the components are operating optimally. A sub-set of these goals are financial goals, this may mean that we have no interest (or will) to fix the fault but we want to know if the system will keep working, even if this is only for a limited time frame. The main difference is conception of the model. The model is constructed from known behaviours of component parts, this is physics and can not be altered, the goals are not 'physics', they are an expression in physical terms of what you would expect the device to do. Although this later definition does not change the physics, it imposes a restriction on what you would expect from the behaviour and thus has a different focus.

The major problem with this goal explicit approach is that it rapidly becomes very difficult to construct anything that we can check against the physical model of the components. This is also related to distinguishing between the presence of faults and deciding which of the faults discovered is interesting at any one moment.

We need, therefore, to make the distinction between model faults and specification / goal faults. When we ask the question of how we can define the bounds or the intervals, then we need to think of it in terms of where they come from, do they come from the intent of the specifications or a faulty component and what does this distinction mean to the way we view the fault.

Further to this, the constraints themselves must be thought of as part of the process. First there is the model of the components; this is the system that you have built and this system has been built in order to achieve some goals. The system, however, can only do this by working within a set of prescribed constraints. This gives us three distinct areas to the process; Models, Constraints, and Goals. The constraints themselves are connected with the environment, this makes them distinct from the Goals which are not connected with the environment, only with how you want your system to perform.

Goals can potentially be described within the model itself. One weakness of this type of approach from the FDI perspective is that they do not describe the components, but one strength is that they can describe ‘everything’. For example, we have a set of equations that describe the components and also a set of equations which describe the goals. One of these goals may be that the system should always be stable and not lose control and this type of goal implies a set of constraints. Thus the constraints can suggest goals which are actually negative, examples of these are financial or technical constraints / goals which state that a certain variable should never exceed a certain level because it requires the utilisation of an expensive resource. With the inclusion of such ‘external’ goals and constraints, the device is then dealing with ‘everything’ that you need to be concerned with. This may take away from the value of the modelled components but gives a much greater view of the process as a whole. From the model building process these two things need to be conceptually separated, although for the end user they may well appear to be the same thing. The overlap comes where the external environment is modelled as another feeder component and, with its own set of constraints, is just another component part of the model.

As has been shown, the increasing complexity in the new diagnosis process is closely related to the previous issues. For instance, a multiple model approach using behavioural, functional and teleological models at different levels of abstraction for the same system could help. At each level proper behaviour / function / goals would be checked.

At the same time, this vast complexity could be tackled by combining model-based reasoning with other techniques. The consideration of situation assessment as a separate task to diagnosis has been done in other areas such as knowledge-based or case-based techniques.

3.1.4 Hybrid Modelling Issues

- Current techniques weak for hybrid systems
- Important to involve the hybrid community
- Multiple levels of abstraction

Hybrid systems are the natural end of many modelling techniques in many domains as they are a better way to represent the complexity which is found there. Although there are of course domains in which hybrid models are not required, they are beneficial in many areas especially if you wish to utilize reconfigurable systems, in which case you need hybrid models. This is one area where the convergence of FDI and DX techniques could prove very powerful and there has been research which has shown how it would be possible to converge the principles in this area.

Tackling continuous and discrete event models provide an entirely new set of issues with regard to many aspects of modelling, tools, level of maturity, etc. Many of these problems can not be solved by one model alone and thus these issues are closely related to hybrid modelling issues. One difficulty in the discrete event world is that of making the abstraction process and in doing so deciding what are the relevant events. However, this task does not exist at all in the continuous modelling world, which is more focused at identifying the relevant variables. The choice between applying continuous or discrete systems is often ‘chosen’ by the nature of the application area, some systems are naturally continuous and some are naturally discrete.

There is actually work coming out of the Defence Advanced Research Projects Agency (DARPA), MoBIES[2] project (Model-based Integration of Embedded Software) and they have investigated a lot of modelling and hybrid model issues. One outcome is to hopefully produce a hybrid systems interchange format (HSIF) in order to have a single format for expressing hybrid models adapted for both industry and academia. This would be interesting for this Task Group to investigate. We now have a fairly clear understanding of the engineering view of dynamic models and of course understand our view of discrete models, we now need to look into how these two could be brought together.

One problem is that the Hybrid community is not very active in the Diagnosis area, thus in the short term we need to concern ourselves with models already in existence, for example, in Matlab / Simulink and in five to seven years both communities will hopefully be in a more advanced position and we should then be discussing the convergence of the fields and the improvement of the results.

3.2 Temporal Issues

- Need to combine continuous and discrete models
- Need to deal better with time and state varying systems
- Deal with changes and fault patterns over long time periods (Diagnosis over Time)

Currently, ‘time’ is only included if we are required to model dynamic behaviour. Nevertheless, there are additional temporal aspects that should be considered in future Diagnosis systems. General theories and techniques are to be developed that handle problem characteristics such as intermittent faults, dependant faults and exploiting information across time, that go beyond current area / system-specific solutions. There are also no general approaches to diagnosing time-varying systems, an important group which includes reconfigurable systems and systems which change parameters over time. Time-varying hybrid systems may even have both aspects. A reconfigurable system is one which can alter its structure (its topology) when required, i.e. if a component fails then the system alters its structure to not require that component. Such

---

[2] http://dtsn.darpa.mil/ixo/programdetail.asp?prodig=38

systems can prove highly complex to model or diagnose because the resulting system will still work even though it contains a fault or faulty component.

We must, however, realise that it is not easy to convert static diagnostic systems to diagnostic systems that include time. This is not so much of a concern in DX, but from within the QR / Symbolic / Control Communities this is a serious issue. The FDI Community is at a further advanced stage with respect to time in models but also tends to eliminate time from the Diagnosis process. These are, however, merely methods of avoiding the problems caused by time, not actually solving them. From the detection and Diagnosis point of view, FDI are not doing as well with time as the DX community, for example, they almost always reason about temporal residuals (even from very complex dynamic systems), as if they were static.

In the FDI Community, much of the work concerns dynamic systems but they do not take into account many of the issues that we have discussed here. Much of their efforts are focused at obtaining highly sophisticated residuals and then applying other techniques (fuzzy, neural nets, etc.) in order to get information out of these residuals. It is, of course, interesting to note that this is an area which is quite complementary between the two communities and an area where the two worlds have differing strengths and weaknesses and where they could both learn a lot from each other. We should probably look into techniques that could take the residuals from FDI techniques and demonstrate how the utilisation of DX techniques could enable them to get considerably more information out of those residuals.

One of the major issues with viewing time varying systems is getting enough data to allow us to ‘step-back’ and see the single big picture over time, so that in an integrated system we have a single view of how that system is operating over time. At the moment we look at faults as ‘islands’ and do not take a holistic view, which leads to a huge amount of small diagnoses over time, when we should be taking this data and producing a clear view of everything that is happening. What is often of use to the customer / user is a single view of what is happening over time. For an insight into the issues and successes of tackling this problem in an industrial context please see the details of the TIGER application, Section 3, page 18 of MBD3 - Industrial Reference Problems Deliverable [2] and Section 3.3, page 10 of MBD5 - Report on the Evaluations to Solutions to MBD3 [3].

We must also question if simply putting derivatives into a system is enough, or whether we actually need to model how we put time into a system. These steps need to be seen as two different issues and dealt with separately. First we need to decide if this is worth doing and then, and only then, think about whether we should do it, or whether it is computationally too complex.

There is also a large class of industrial systems that would benefit from development of temporal techniques.

The explicit inclusion of time points and intervals in Diagnosis theory has yet another set of issues which must be investigated. There is some work in Diagnosis of discrete-event systems which has demonstrated that we can construct a Diagnosis over time which is consistent with the observations across time; however this is all built on a theory that has no explicit time representation in it. The process uses reverse reasoning to gain the results. The question here is whether we are happy to apply these techniques or whether we believe that because the theory has no time element in it, it will not ever be able to reflect real world situations. Theoretically, of course, we should always include time, but there is the question of the vast increase in complexity that this will cause, it may make the models unusable and therefore invalidate the exercise.

This may also suggest that we should adopt models that do not include time explicitly (typically discrete models) to diagnose dynamic systems and to switch to time-explicit models only to refine the results. In other words, time-explicit / continuous models should be invoked only;

(i) when a further discrimination among the candidate diagnoses produced by lighter models is needed, and
(ii) if they actually have a higher discriminating power in the specific situation at hand.

These issues are also closely related to problems with intermittent faults. The system may only show the symptoms some of the time, so a model is needed that links a fault in time to a specific area of the model.

3.2.1 Intermittent Faults

- Revise Diagnosis over time to capture intermittent faults
- Correlate fault occurrences and system state
- Models rich enough to capture system state

This is a significant issue. One of the major issues that makes it so significant, is in determining whether the fault is an instance of a genuine intermittent fault, or whether this is an intermittent manifestation of a permanent fault. To tackle this we would need to be able to revise our Diagnosis across time. An example of intermittent faults versus intermittent symptoms can be shown in a car ESP system which occasionally reports a fault. This was actually due to a steering wheel sensor, which was only registering the fault when the steering wheel was at full lock, this is an intermittent manifestation of a permanent fault. The difference between the two is further clouded by the issue of faulty connections / wires, these may produce an intermittent fault ‘signal’ (sometimes connected, sometimes not) but it is actually a ‘permanent’ fault, in other words the connection is always loose.

Diagnostic systems are designed to record ‘faults.’ They do not record every state of every component in the system at the same time; as would be required to diagnose the above ‘steering wheel’ example. This also brings up issues of automatic recovery through reconfiguration, in which case you still have the faulty component but you do not register the fault because the system is in another state which does not exhibit the fault, although it is obviously still there. Many techniques by-pass many of these issues and complexities by applying detection techniques. This can be done successfully and reasonably simply but the complexity is significantly increased when you attempt to ‘Diagnose’ the intermittent fault.

These issues make the Diagnosis process very important as intermittent faults are a grave weakness of Diagnosis, usually because the models are not rich enough to capture all of these operating conditions. One solution is that the model is not good enough, however we may chose not to develop better models (as this is often a great deal of work) but we could spend our time on developing strategies to manage and understand the intermittent nature over time. This may require management work and not just model-based work.

‘Random’ faults may also play a part here and these are possibly a step towards solving these issues. If the random action of the system (effects on operation of an increase /

decrease in CPU temperature for example) can be built into the system model then it may help us to understand faults that are truly intermittent and genuinely disappear when the system returns to ‘normal’ operating parameters.

One approach to creating more Industrial use of / belief in these systems would be to identify exactly what we need to change in an industrial model and demonstrate exactly how much more an Industry can get out of their system, if they put in, for example, an extra sensor or ECU, etc.

3.3 Diagnosis and Product Life Cycle

- Consider Diagnosis at design time
- Use design models for Diagnosis
- Deterioration as well as crisp faults

If Diagnosis is to have a real and continuously improving effect on the performance of real world systems, it must be integrated with other high-level tasks, such as monitoring, planning, reconfiguration, and so on.

Also, Diagnosis should be considered in the whole product life cycle, from design to maintenance. Diagnosis is not often considered at the design stage. Diagnosability analysis can be highly valuable at this stage and produce key design requirements. For instance, if online Diagnosis is not going to be done, then a structure / system should have a very high level of safety built in from the beginning. However, if we are going to be doing Diagnosis online, then this can be built into the design phase and means that we would not need to build so much safety and control into the system.

Additionally, models themselves need to be considered in the whole product life cycle. The models used would need to be changed, adapted, updated and re-configured.

Finally, there will be a whole spectrum of possibilities that arise from using a set of diverse techniques in combination with MBR.

A factor that needs to be addressed is what elements of a system can be updated automatically as opposed to what requires manual intervention. Ideally we would want to automatically update the model as the system matures, this would, of course, be of great benefit to the longevity of both the system and the model. This is a goal of the development of the field as this is not practicably possible today, outside some basic retuning.

This area is also related to taking deterioration into account during the modelling process. This would require including the results of the Diagnosis in future diagnoses and includes the requirement for a system that can diagnose when the ‘Diagnosis fault’ is in the system or the model itself. This ‘Learning versus Diagnosis’ issue would be a considerable step towards Industrial acceptance of Diagnosis models, however it has the potential problem of the model ‘learning’ the fault and claiming it is normal.

3.3.1 Chronic versus Acute Fault Modelling

- Automatically update model-based system with incremental design changes
- Deterioration sensitivity versus model uncertainty
- Adapt to different instrumentation

In order to model systems over time, we need to deal with two sides of an important issue. Firstly, the system needs updating for incremental design changes. Secondly, as the physical systems get old, sometimes the decision is taken not to fix things. Therefore we also require models that can do the inverse and adapt to this lack of expected information. The former should be possible with current technologies but the latter is a much more difficult challenge for a Diagnosis system. Updating the models is also not just an issue of time, it is also a matter of the conditions which the system operates under (for a car, for example, if it is a city or country based vehicle).

This is partly due to the tolerances inherent in the model; many of these are set only to detect a fault when it is critical and thus will not be good at detecting something that is evolving and worsening over a period of time. However, if we design the model to detect very small changes in the operation of the system then we will receive a vast number of fault alarms, possibly too many to detect the difference between noise and deviations in the operation of a component. We need to have a fault model and a deterioration model and be able to use these in combination, but unfortunately the deterioration models are even harder to generate than system models. This becomes even more difficult still, when we go beyond detection and start trying to model for fault identification. Detection is possible in some circumstances, localisation may also be possible but is harder; fault isolation is almost impossible for this area at the moment.

The problem of long term sensor loss is a major issue for both industry and the industrial acceptance of modelling. When a model is created it is designed to utilise all the available sensor data, but as the system matures, one or more of these sensors may cease to function. This could mean that the model would then not operate optimally, but potentially means that the model may not operate at all. This is slightly different to detecting minute changes in the dynamics of the system. This is a long term temporal issue. In the short term we need Diagnosis of faults and in the long term we need Diagnosis of deterioration, i.e. Diagnosis for maintenance. Statistical methods may prove to be more useful in tackling this kind of deterioration detection than model-based techniques, at least at the present time.

3.3.2 System Overhauls and 'Tune-ups'

- Model-based Diagnosis system to adjust to 'tune ups'
- Understanding when a tune up or system change has taken place
- Preventative and condition based maintenance

This is a very significant area for Diagnosis and is tricky due to the fact that when the service is done the system goes through a major step change and any Diagnosis system would be required to adapt very quickly. Over long periods this will happen several times with the system undergoing these large step changes every time it is serviced. To make our Diagnosis product viable for the whole product lifetime, it needs to very rapidly take these changes into account. This also reflects upon modelling for deterioration, where the system will deteriorate over time (which needs to be reflected in the model) but on occasions take a huge step up in its operating efficiency, which also needs to be reflected in the model. This requires models of how the system deteriorates and how it returns to normal, in other words a family of models is required throughout the life cycle.

Although it is obvious to operators and users when a major overhaul has been done, as this requires manual intervention and usually needs a system shut down, it is common not to tell the ‘system’ or the Diagnosis model that there has been an overhaul. Many current systems do not even include a manual input to tell the system that components have been updated. For car maintenance and servicing this should not be too difficult as much of this work is now computerised, but the model still needs to know that some parameters have changed i.e. the car has new brake disks / clutch fluid / etc.

Models should contain information on how system use affects its life cycle - other research areas (reliability analysis) are dealing with this and bridges to other areas would assist in the future.

It would be interesting to investigate how system changes would affect discrete event models.

Condition based maintenance could be used to tackle this issue, but we must also look at how we can ensure that the system works correctly even if it has a fault (i.e. fault tolerant systems). Fault tolerant systems do not necessarily require Diagnosis; it could just be done with sufficient redundancy. This approach fails, however, when major components cease to function or serious problems occur. In circumstance like these you need Diagnosis in order to identify the area at fault. FDI has dealt with deterioration using the controller which adjusts inputs as required. With these techniques you must differentiate between ‘fault tolerant’ control and ‘robust’ control. With ‘robust’ control (Diagnosis techniques), the model has a set tolerance, however it is not possible to work out where the system is within this tolerance. This is because Diagnosis cannot work alone and needs to be integrated into other forms of monitoring. In other words, Diagnosis should be integrated with other system robustness mechanisms.

If this type of process is to be used in real world settings, then we need to show that Diagnosis makes running these systems cheaper. There are two main ways in which this could be achieved. The first is the ‘insurance’ argument, i.e. you pay a little and this reduces risk and prevents problems, the other is case studies where we can demonstrate how much money these systems have saved people in the past.

3.3.3 Fault Tolerant Control

- Coping with systems that compensate
- Reconfigure the model-based system when the system changes

Fault tolerant control concerns the interaction between a system and its controller (in whatever form) and is the name used by the ‘Control Community’ when they design fault tolerant system. In this area, the process of making a system fault-tolerant, consist of two steps: fault Diagnosis and control re-configuration. The first step concerns the Diagnosis and is used to decide if the fault is critical or not (fault identification). The second step responds to the fault and adapts (if this is possible) the controller to the fault situation with the objective of ensuring that the overall system continues to satisfying its goal(s).

Fault tolerant control, in the Control Community, is more focused on the design of the controller, because the controller must include the required supervisory system(s). In this area a fault tolerant system is obtained by retuning the controller, changing the control specifications or/and control structure. This is done, almost exclusively, by focus on an input / output assessment of the system. For example, if one of the ailerons on a plane stops operating the system merely compensated with the others to ensure normal operation is continued, it makes no attempt to diagnose the cause of the fault. The Control Community, in this case, would attempt to design a controller that allows a system to perform adequately in this situation. This is more control reconfiguration, not system reconfiguration. FDI is used to change the control characteristics but does not concern itself with how the Diagnosis is generated; this is an ideal place for DX to express the value of Diagnosis.

In general, fault tolerant control is an area that needs a variety of techniques, for example:

- An accurate analysis of system redundancies
- Techniques for modelling failure modes and effects analysis failure analysis
- Classical and robust control theory
- FDI
- Supervisor system design

Due to this fact it is an excellent topic with which to integrate both DX and FDI approaches.

The major issue is still the problem of integrating the tools. Timescales are also an issue, control loops operate in real time but the Diagnosis system may not.

It must also be stated that Diagnosis, as a process, is not just aimed at detecting faults; it is also to provide an accurate (and preferably up to date) state estimation even if there are no faults.

3.4 Autonomy versus Human Interface in Applications

- Inform of faults or automatically correct them?
- Appropriate interaction with people
- Autonomous versus active Diagnosis
- Toolkits are needed
- On-board versus off-board Diagnosis

Before this issue is investigated for any particular application, a decision must be taken as to whether we wish the systems to be autonomous or whether we merely want to inform the operator. Much of our work with safety critical industries (nuclear power plants, satellites, etc.) state that they do not want any level of autonomy and the decisions must be made by the operators. Consideration must also be made to distinguishing between operators and engineers, many real time systems are not considered to be so by engineers as they only see them at specific times. One paper [4] states, however, that in the Chernobyl nuclear disaster there were a lot of safety measures that were built into the reactor which were deliberately turned off by the operator in order to get the reactor into a specific state which was one where it should never have been. This is one example of why they may require some level of autonomy in these safety critical systems.

Another important factor to be taken into consideration would be the consequences of the system not operating correctly. In satellites, for example, earth-based operators can always put the satellite into a ‘safe’ position in orbit (i.e. dead position, panels facing the sun) and leave it while technicians try to work out what is wrong and then diagnose it from the ground. This is obviously not the case for a motor vehicle which is being driven. The level of autonomy is increasing in some areas of space applications. The space shuttle for instance, has a high degree of system autonomy due to the length of time that it takes to send the messages that are controlling the systems. They operate autonomously until told to change state; they then comply and operate autonomously within their new operating parameters. However there will always be safety critical ‘times’ (re-entry for example) which may be more critical than others and these are usually run from a pre-determined plan, whereas the rest of the time a certain level of autonomy is acceptable.

From the Diagnosis perspective, we do not really need to focus too tightly on whether there should or should not be autonomy in these systems as it will usually be the end-user industry which will decide this. We should focus on how we would do the Diagnosis differently depending on whether the systems being diagnosed are autonomous or controlled by operators.

We must also distinguish between autonomous systems and autonomous Diagnosis and always keep in mind what the results of the Diagnosis are required for. In other words, do we need a complete diagnostic analysis of a faulty component if we are going to reconfigure the system so as not to include it. In some circumstances you could close the ‘fault loop’ without having a very high level of Diagnosis. Incremental Diagnosis could also be used. This is a technique by which you start running simple diagnostic tasks and increase the complexity slowly until the solution is found.

‘Active Diagnosis’ is also an important area and requires us to investigate what configuration we put the diagnosed systems into, in order to produce a more effective diagnostic result. We should also consider specific (and particularly acute) circumstances, such as Mars exploration. In such areas control is still in-the-loop but where the tasks are not safety critical, for instance, they may be permitted certain levels of autonomy. Another example is water systems for towns in the Andes where the water is collected thousands of feet higher than the town. It takes a long time for a person to physically get to the system to check it, so the Diagnosis is used to ‘suggest’ what might be the problem before the human operator intervenes. This also means that if parts need replacing an operator has a much better chance of selecting the correct one to take with them the first time they go up to the machinery.

This autonomy with feedback work is very close to that of the Control community and therefore makes is an ideal area for bridging the two communities.

Diagnostic applications must interface with human beings. Currently, the industrial application of Diagnosis involves monitoring which merely produces a report for an operator, i.e. the system simply activates an alarm. A considerable amount of human intuition is still required to interpret these alarms. Although it must be pointed out that the Diagnosis process is much more than just ‘firing’ alarms, Diagnosis is about detection and the ‘alarm’ is just the record that this detection has occurred. Modern diagnostic techniques do, however, reduce the processing time of the human operator by providing more information and not just an alarm, of which there can often be a vast number. In the future diagnostics will be used to alert an operator to the fact that the alarm has occurred and produce a Diagnosis of any possible fault. Eventually this would lead to the eradication of ‘alarms’ - instead we would have an autonomous system which identifies and corrects the fault itself and merely informs the operator that this has been done. This is an 'ideal' scenario and in actuality there may be a need to stop the machine to reconfigure, although the diagnostic report could include instructions on how this can be done by the operator.

There are people who believe that there are some domains in which systems are improving and operators are becoming less significant. Many systems still rely heavily on the ability of a human operator, but as systems become more complex it is becoming harder to train experts with all the relevant technical information. Research is being done into the possible removal of operators from a system entirely. It is envisaged that this may even be possible in 'Critical Domains'. There is a great deal of interest in research in unmanned operations in areas such as Satellites; there have even been demonstration models built, but it may still be ten years before any real world applications are available. However, in areas such as Nuclear Power Plants it is considerably harder to say when (or if) an unmanned system would be possible, let alone acceptable.

As a motivating example, we mention an application in the Automotive Domain: ADAS (Advanced Driver Awareness Systems); see for example the European project AWAKE (System for Effective Assessment of Driver Vigilance and Warning According to Traffic Risk Estimation[3]). This is a diagnostic system which is designed to ensure the continued alertness / awareness of a motor vehicle driver. This system uses the wireless support mechanisms needed to deliver the co-ordination of on-board and off-board systems, which has proved necessary due to restrictions of using solely on-board systems. This is an important automation issue as it involves an operator (i.e. the driver) and shows that the operator themselves should be seen as a 'component' of the system and one that can begin to function below optimum levels. It should be noted, however, that this is not a project which utilises a model-based approach.

This application identifies single faults in a system and can inform the driver of these issues. However, it also deals with the possibility that if the driver is tired, they themselves then become the 'fault'. For this Diagnosis to be workable, problems have to be recognised before they can be diagnosed. You then need to define state assessment, which needs to be formalised and the system adjusted. It may be possible to approach the situation from the data and then model it. The one problem with taking a statistical approach to diagnosing systems is that many sensors are required in order to gather sufficient data for statistical analysis.

In the general area of on-board versus off-board Diagnosis there is still a great deal of debate as to which direction to go in the future. For some time most large automotive manufacturers were aiming to put most of their systems on-board, up to 90% in some cases. Recently they have moved to off-board systems and wireless links to the vehicles. Even more recently other issues have emerged to change this direction again. Of primary concern to many automotive manufacturers is that if a garage is licensed to more than one make of car they must have systems to deal with them all and thus there will be detailed models of one car type potentially available to others and manufacturers are concerned about this.

4 Taking the Bridge Community Forward

This section will discuss possible future directions for Bridge and also covers the motivations of the researchers (their 'wish lists') and the hopeful convergence of DX and FDI communities and techniques. This latter point must obviously include an evaluation

---

[3] http://www.awake-eu.org

of our current situation and then look at both how we want the situation to evolve and how we feel it may actually do so. Some of these are very practical issues, such as the co-location of the DX'03 workshop (International Workshop on the Principles of Diagnosis) with SafeProcess'03 (a Control Community Conference) and whether we thought this was useful and whether we should do this again.

It appeared that at SafeProcess'03, many FDI people attended the Bridge Day at the SafeProcess event itself, but very few came to the DX workshop. This would possibly suggest that co-location is not necessarily strategically significant (although it obviously aids with avoiding the requirement to travel to two events) but that the DX stream at SafeProcess was very useful and is something that we should look into repeating in the future.

It is evident that the two Communities have grown in awareness of each other through all of the events at which they came together (many thanks to MONET). Now that this bridge between them is established, it will be more through the scientific issues that the convergence should be pushed, for example, through issues such as hybrid systems. One problem is that researchers (both professional and academic) tend to utilise the techniques that they are familiar with and tend not to change their way of working too often. One way around this is to help to develop a new generation of these researchers (through PhD Studentships, for example) that can grow and utilise a combination of the two techniques. This would create a new generation of researchers who would take these techniques out into real world positions (and latterly real world applications) in the future.

In the near future, there should be a set of Diagnosis Toolkits available for the Technology Supply Companies, containing MBD techniques (with the best from both MBD communities) within a common framework. This fact could also ease the process of integrating Diagnosis with other high-level tasks in real applications.

Another option, is that if MONET continues with an electronic newsletter, we should make an effort to send this to as many interested parties in the FDI community as we can. We should also continue discussions between the people and institutions which MONET has brought together, when we meet at major international events. This could be done to discuss specific issues / projects or merely to keep everyone aware of the research / industrial activities of the people in our field. We are also proposing a workshop on these topics at the European Conference on Artificial Intelligence (ECAI) in 2005. It is one indication of the positive effects that the MONET Network has had on the wider AI Community, that papers by MONET members won both the Awards at this years ECAI Conference: the 'Best Paper' award and the 'Best Technical Paper' award. Marie-Odile Cordier, of Bridge, has also been elected to the ECCAI board.

One possible event that may be of great benefit would be to have a conference based around Diagnosis which is not merely model-based. Usually workshops and conferences are community based, i.e. DX or FDI, but it may be of genuine value to many researchers to have a workshop which deals with all diagnostic techniques from whatever community. Officially DX is entitled the 'Workshop on the Principles of Diagnosis' and does not include just model-based approaches, however in practice it tends to be focused upon them. We could, maybe, work to try and introduce a broader Diagnosis spectrum to future DX workshops. Although it would make the review and acceptance process much harder and the presentations would possibly need to be simplified because not everyone in the audience could be an expert in every approach presented. However, the success of the Bridge Special Issue of IEEE-SMC [5], shows that there is a great deal of interest in diagnostic techniques from other approaches.

There is also the virtual journal Electronic Transactions on Artificial Intelligence (ETAI) which is driven by very different reviewing style and ranges across a spectrum of subject areas. We could try to arrange for a MBR or DX subsection of this ETAI journal. The way the system works is that when someone submits their paper, the submitted paper goes immediately onto the web for everyone to read and if it is ultimately accepted then its publication date is the date it first appears. Everyone can then put their comments up on the web and certain people are appointed to be referees and their comments are also put up on the web. This is a very new style of journal and one which is also very open. There is, therefore, an opportunity for a person from this community to become the technical contact for this domain area.

This approach is not necessarily the way forward but it is one idea. It would, however, assist in tackling the problem of papers (especially student ones) in our domain being published in a wide variety of Journals in different areas. This makes it difficult for the community to keep an eye on the spectrum of work in this area. A Diagnosis specific journal would obviously achieve this but would require a lot of effort and considerable time would be required for it to be widely recognised.

5 Graphical Roadmap

	Now		2 Years		5 Years		10 Years
Drivers	1. On-board / Off-board Debate	1. Industry has too many processes that require Diagnosis techniques for the first time	1. Integration of Diagnosis and other Tasks (i.e. Design, Training, Documentation and Maintenance) (< 10 Companies)	1. Need to obtain a sound explanation of the systems in general use	1. Effective Integration of DX and FDI within a common framework	1. Many systems need to be fault tolerant.	1. Systems so complex that Diagnosis must be considered at Design Time
	2. DX cannot easily Integrate MBR and MBD with engineering environments	2. System variants limit application of Diagnosis techniques	2. Wireless Communications	2. Customers expect customised products (variants) at low cost.	2. Requirement for Preventative Diagnosis and Predictive Maintenance	2. Diagnosis smoothly integrates in a well defined supervisory system.	2. Customers expect customised products (variants) at low cost
	3. Model Building Technology not yet available		3. Managing the different views of several Diagnosis agents	3. Maintenance costs of systems start to become **too** high due to system complexity	3. Desire for Condition based Maintenance	3. Companies want to remotely diagnose and reconfigure (embedded) systems (because of complexity and wireless connections)	
	4. Systems often too complex to train engineers to Diagnose		4. Some systems can be automatically re-configured’	4. Diagnosis of Natural Systems (ecological, biological)	4. Environmental Impact Analysis (EIA) and increased Ecological / Legal requirements		
	5. Internet Support		5. Diagnosis of Software				
	6. Too many 'No Faults Found' or 'Hard to Diagnose' Systems						
	7. Systems are very expensive to support						
	8. Use of multiple level models						

9. New Types of Sensors (e.g. Fibre optic Technology)

10. Requirement to Diagnose Systems that include ‘Feed back’, i.e. dynamics systems

11. Industry has many processes that require Diagnosis for the first time

12. System variants limit the application of Diagnosis techniques

	Now		2 Years		5 Years		10 Years
Technology	1. System complexity restricted	1. Automated Modelling from data	1. Distributed Diagnosis (e.g. in automotive domain due to network introduction, also in space applications - satellites constellations)	1. Industry asks for fault tolerant systems (for productivity and safety reasons). Thus Diagnosis will be more and more required	1. Common technological framework accepted	1. A significant class of temporal issues can be addressed	1. Diagnosis at the design stage - the only way to get models
	2. Re-usable libraries in only a few domains + very limited capability with Industrial Modelling Systems	2. Applications with combined FDI and DX techniques	2. Initial Development of Industrial Autonomous Model updating Techniques	2. FDI and DX tasks will be embedded in components along with other control tasks	2. Ability to relax single faults assumption for complex systems	2. Models able to take account of Deterioration in systems	2. Problem characteristics not a major limitation
	3. Can only model engineering domains of medium complexity		3. Integration of planning and Diagnosis (for autonomy or operator informing)	3. Diagnosis as state estimation within a whole fault tolerant / automated architecture	3. Diagnosis techniques Critical to system demonstration[4]		3. DX as the reasoning core of the Diagnosis tasks, providing a framework to integrate other techniques
	4. Current Diagnosis techniques are limited in their applicability to time-varying system				4. Models automatically development from both data and mathematical techniques		4. Diagnosis integrated in a well defined supervisory system
	5. Industrial Diagnosis as state estimation (Diagnosis over time) in a very few systems[5]				5. Taking advantage of sensor networks		5. Reorganising the cycle Design - On board Diagnosis - Maintenance
	6. Diagnosis not considered at the design stage						6. Most temporal issues are addressed
	7. No temporal Diagnosis in FDI Approaches						7. (12 + years) some systems still too complex
	8. No accounting for deterioration models						8. Components become self diagnostic

---

[4] What we mean is: System that works 'only' because we have DX techniques involved

[5] What we mean is: FDI uses state estimation to make a state estimation and then adjust / revise the Diagnosis when different observations appear over time

	Now		2 Years		5 Years		10 Years
Products	1. Not Distributed - Only one core part		1. DX and FDI Techniques used together in 1 application	1. DX toolkit software product	1. FDI and DX integrated software toolkit product	1. Model-based techniques become commercial valuable in 1 application area. Hence more people learn about it (Killer Application)	1. FDI techniques used by almost all companies in 1 sector for a specific task
	2. System Alarms, i.e. fault detection - No Diagnosis		2. DX techniques used as standard in 1 aspect of at least 1 application domain	2. Diagnose systems with high complexity	2. Diagnosis distributed in the system	2. (MBR) Diagnosis systems included as standard part of product for several products	2. Automated DX in some Critical Systems (Satellites, autonomous systems)
	3. FDI Tool-kit Products		3. FDI techniques used as standard in 1 aspect of >10 application domains	3. Key technology companies understand DX and FDI Techniques	3. Systems can go beyond only Alarms	3. FDI and DX Techniques supplied successfully in >10 domains	3. Diagnosis systems can perform fault identification for complex systems
	4. FDI techniques used in some commercial applications			4. Key Technology Supply companies can offer FDI and DX techniques	4. Automated Modelling tools in 5 Domains	4. Integratable Model Based Diagnosis. Modules in Hybrid Solutions (5 to 7 Years)	4. Systems are designed to make the use of DX techniques easy
	5. DX techniques used in a few commercial application				5. Automotive Domain - Intercollaboration between different modules in a process. (Advanced Driving Assistance Systems - ADAS)	5. Tele-Diagnosis systems in use	5. Automated Modelling tools in >20 domains
	6. Diagnosis systems are limited to detection and localisation in complex systems				6. Automotive Domain - Optimisation of the entire on-board system (Common resources, communication, fault tolerance and recovery)	6. Smart sensors (image, taste, smell, etc.) exploited for commercial applications	6. Ubiquitous Diagnosis

7. No products include FDI and DX as standard products	7. Commercial application exploiting Diagnosis with sensor networks
8. Correction of faults (re-configuration) is done manually	
9. Industry Applications are limited to techniques that suit FDI techniques (MATLAB, Simulink modelling)	
10. No Generic systems available	

6 Technology Gap Migration Roadmap

	Today	Gap - Migration Path	Vision
	>>TIME >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>		
1 Assumptions	1. Have to make restrictive assumptions due to restrictions in techniques	1. More widespread use of DX techniques	1. Elimination of restricting assumptions
	2. Limits potential application domains	2. DX techniques need to be applied to more systems	2. Dynamic & Temporal issues included in Theories
	3. Problems with fault propagation and fault localisation in time. (State based)	3. Restricting assumptions to be reduced	3. These technologies applied to hybrid systems
	4. No dynamic reasoning	4. Temporal issues to be resolved	
	5. Non-time varying Models	5. More complex and combined models	
	6. No wear models	6. Consider fault identification and early detection	
	7. Not able to perform Preventive Diagnosis	7. More research required. Techniques need to be further improved	
	8. Set of faults are often anticipated	8. Consider masked and compensating faults in practical applications	
		9. Model and technique combination	
		10. Robust solutions for 'noisy' and disturbed systems	

2 Diagnostic Community Understanding	1. Currently FDI, DX, with Bridge in between	1. Core techniques need to be combined	1. Technical Support Companies would offer both techniques
	2. Potential for DX to be absorbed by FDI as it is so small	2. People need to understand the other set of techniques – possibly need to teach both techniques at an earlier stage	2. People would understand both techniques and use them (as well as combined techniques)
	3. Tech companies know FDI, but don't know DX	3. Better characterise the best techniques for more classes of problems	3. Maintain 2 communities with a good communication system and superb understanding of one another's techniques
	4. No agreed terminology	4. SafeProcess co-located with DX again	4. One common conference for the whole Diagnostic community
	5. People in Industry understand one technique but not the other – limited technological cross over	5. Key research groups need a common understanding	
	6. FDI taught at Under-grad level. DX taught only at Post-grad Level	6. Identify common areas	
	7. FDI people do not use DX, but DX Community beginning to look at FDI	7. Key researchers to use both sets of techniques	
	8. No 'Diagnosis Community' in AI (unlike FDI for engineering)	8. Key Conferences have papers using both techniques	
		9. Leading edge technology supply companies understand FDI and DX techniques	
		10. Leading Professors (5 of) in the field tackle difficult problems that require both FDI and DX techniques	
		11. Hybrid systems development drives the union of FDI, DX and related	
		12. FDI and DX Summer School for major players in fields	
		13. Requirement for funding in Europe	

3 Integrated technology approaches	1. Bridge!	1. Combined approaches will be required	1. To clearly identify and integrate the technologies so that more complexity and system autonomy and completeness with respect to types of fault can be achieved
	2. Very few other attempts being made and these only in research world	2. Consider lessons learned from other AI techniques paths into FDI for integration of DX technologies	2. To be able to design fault tolerant systems and produce a better system state assessment
	3. Domains that require both technologies are limited with today's formulated requirements	3. Links and common concepts need to be explored further. Leading Profs. to address problems in both areas	3. Smooth integration of Diagnosis within higher-level tasks such as supervision
	4. Research world is far in advance of industrial world	4. Example problems and results of the common ground need to be found	
	5. People are beginning to diversify at present	5. Compare different techniques on relevant reference problems	
	6. Hybrid systems track offers potentially good opportunity	6. Extend DX/FDI techniques into other application domains and communities. Work with people from other communities with some areas of research	
	7. Very few Diagnosis combined approaches (model based only)	7. Need to demonstrate what DX can achieve for the other research communities	
		8. Need to be ambitious without moving too fast and losing researchers along the way	
		9. Should look at high level tasks such as ‘Supervision’, ‘Monitoring’ and 'fault tolerant control'	
		10. Provide DX examples in areas where FDI falls down	
		11. Common Terminology	
		12. Integration of techniques across application Domains	

7 References

[1] IDD Project - FP5 project 'Integrated Design Process for On-Board Diagnosis' 1st February 2000 to 31st of January 2003. Project Reference: GR3-CT-1999-00058.

[2] MONET Deliverable MBD3 - Collection of Industrial Reference Problems. http://monet.aber.ac.uk:8080/monet/docs/tg_minutes_and_reports/bridge/mbd3.pdf

[3] MONET Deliverable MBD5 - Report on Evaluation of Solutions to Selected Problems of Industrial Relevance.
http://monet.aber.ac.uk:8080/monet/docs/tg_minutes_and_reports/bridge/mbd5.pdf

[4] Stein, Gunter. Respect the Unstable. The practical, physical (and sometimes dangerous) consequences of control must be respected, and the underlying principles must be clearly and well taught. IEEE Control Systems Magazine. pg. 12 - 25. August 2003

[5] Special issue of IEEE SMC Transactions - Part B, Diagnosis in Complex Systems: Bridging the methodologies of the FDI and DX Communities

8 Document History

Version	*Date*	*Changes made to document*	*Changed by*
2.1		Final Roadmap Version One	
3.0	28th September 2004	Reformatted and V2 Drafted	RIR
	15th October 2004	Sent to Rob Milne and Louise Travé-Massuyès for pre-approval	RIR
3.1	20th October 2004	RM and LTM changes at RM meeting	RM, LTM
3.2	29th October 2004	Updated with Initial Reviews from Task Group	TG
4.0	1st November 2004	Revisions Received from entire Task Group and incorporated into document	RIR
4.1	4th November 2004	Final Proofread	JNT / RIR
4.2	18th November 2004	Updated with Teresa's Comments	RIR
	18th November 2004	Document Status Set to Release	RIR