Automatic Modelling for Diagnosis

H. Ahriz
LIA, Université de Savoie, Le Bourget du Lac, 73376. FRANCE.
ahriz@lia.univ-savoie.fr

S. Xia
Computing School, De Monfort University, Milton Keynes, MK7 6HP, U.K.
kxia@dmu.ac.uk

Abstract

Much of the past work on fault diagnosis did not pay enough attention to model construction and its important role in aiding problem solving. It was generally accepted that a model was available or was assumed to be present in a certain format before starting the diagnosis process. However in practice a model which can be constructed from an engineering or commercial system is often different from the model on which diagnostic algorithms have been developed. Our paper aims at filling this gap between the model construction and model-based fault diagnosis, providing a framework to integrate them coherently.

1. Introduction

It is generally accepted that three stages of work are involved in the model-based approach to analysing a system. Firstly, a model of the system is built. Secondly, a solution is solicited from the model. Finally, a conclusion about the system is reached based on the interpretation of the solution. The importance of using a good model is obvious because building a model is the starting point in the whole process of problem solving.

Fault diagnosis is a model based study and requires the use of good models. Traditional models are constructed by heuristics and are then used in experiments to ensure results are acceptable. Models produced in this manner tend to include everything, including issues irrelevant to an application, and require large scale computation. These considerations, and many other, indicate the need for new approaches to modelling, based on more rigorously defined modelling processes. These well-defined processes record major intermediate changes on the model, together with their underlying conditions explicitly and make them available for examination when necessary: automated modelling is such an approach. It attempts to generate models which are parsimonious, making need distinctions apparent and aiding problem solving.

In this paper, our focus is on providing a modelling framework for the diagnosis of dynamic systems. Our work is an improvement and a supplement to AIM (Automated Intelligent Modeller) a general-purpose automated modelling system [Xia et al, 93], based on the bond graphs methodology [R. Rosenberg and D. Karnopp, 83] and qualitative simulation. This paper aims at overcoming the needs of the old version of this environment, and includes the task of diagnosis as well. In the rest of the paper, we will refer to the new framework as AIM+.

In section 2 we describe briefly how AIM+ proceeds to build a model which can be useful for diagnosis.
Section 3, is dedicated to the description, in details, of the different tasks of this framework. Along this paper we use the case study of Figure 1: a motor is driven by a voltage source and, in turn, drives a pump, and then pumps fluid from tank 1 to tank 2.

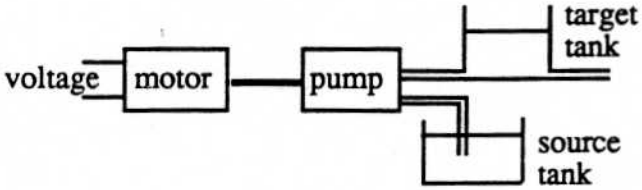


Figure 1: case study system

2. AIM+ Framework

Modelling and diagnosis are the two main functions of AIM+. The first function is executed by the following loop:
- taking as input a structural description of a system to be modelled and a set of conditions that the model must obey. The structural description is given in terms of the components that go to make up the system and how they are connected. The modelling conditions are given in terms of the behaviours the model should exhibit in certain circumstances. Each component has model fragments stored in a library;
- an initial model is created by finding the simplest model for each component (expressed as a bond graph), and then combining them to form an initial model. From this, a causal graph is derived that is used to verify if the model does meet the conditions specified. If it is not the case, a component is chosen to be altered and it is replaced with the next most complex model for that component. The model is then reassessed and the cycle repeats until all the model criteria are met.

The modelling algorithm can process the first best (simplest) model, or all the models satisfying the criteria, and thus, obtain the best one.
The diagnosis task, then, deals with the chosen model, and with given observations on the system, to process candidates.

Figure 2 summarises the modelling method, and the issues adressed in our research, namely: the representation and study of: the device structure, the behaviour constraints (model criteria), the component's functionality, the library
of generic components, model selection and verification, and finally the diagnosis process.

The diagnostic function is intrinsically related to the modelling function and they are integrated in the new framework AIM+. If the model produced from the modelling process is acceptable and is used as a reference against any malfunction, the diagnostic process will deal with any faults of real applications. On the other hand if we have an application and need a good model for it, the diagnostic function can be used to complement the automated modelling process. In either case the modelling and the diagnostic functions are mutually dependent and their close relationship is examined further in this paper.

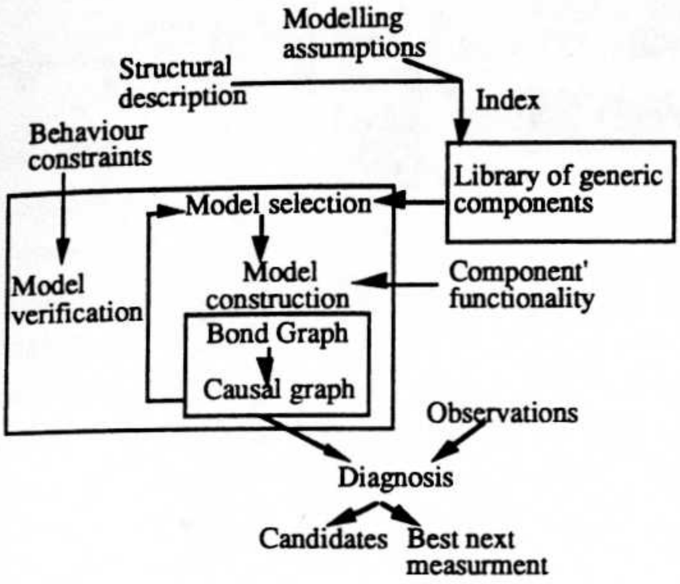


Figure 2: AIM+ Framework

3. Modelling and Diagnosis issues

In this section, we discuss, in detail, modelling and diagnosis issues introduced in the previous section.

3.1 The Device structure representation

The device structure representation is an abstracted view (model) of the physical system. It is Component-Connection based, and, thus, contains the description of system components, connections (in the different physical domains) between component terminals, and the specification of the inputs as well as the outputs of the system. A declarative language (Prolog) is used for the device description.

Let us, consider the structure description of our case study presented in figure 1.

The schematic description is depicted in Figure 3:

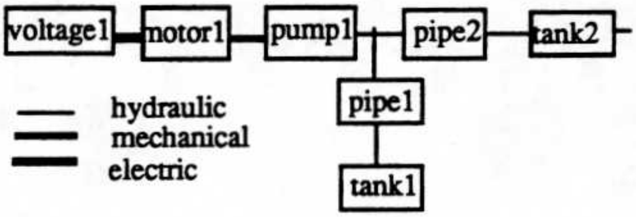


Figure 3: schematic description

The declarative description (equivalent to the previous one), is given below:

```
input ([]).
output (tank2-hydraulic).
set_of_connections([
connection (electric, [voltage1],[motor1]),
connection (mechanical, [motor1], [pump1]),
connection (hydraulic, [pump1, pipe1], [pipe2]),
connection (hydraulic, [tank1], [pipe1]),
connection (hydraulic, [pipe2], [tank2])
                    ]).
```

3.2 Representing Modelling Conditions

3.2.1 Behaviour Constraints

In addition to a description of the system's structure, inputs to AIM+ must include a set of behaviour constraints. A behaviour constraint describes in qualitative terms one possible dynamic behaviour of some device variables.

The specification of these intended behaviours is represented by: "premises ⇒ conclusion" clauses. Premises and conclusion are represented as calculus formula (Object, Attribute, Value).

A particular constraints class is of interest, namely: the constraints which relate a cause (malfunction) to an effect (symptom).

The model verification task must insure that, given a "conclusion", the model should be able to point out the "premises" specified in the constraint as possible causes for it.

For our case study, examples of behaviour constraints can be like this:

```
constraint(motor1,speed,+) :-
        constraint(tank1,volume,0).

constraint(tank2,flow,-) :-
        constraint(pipe2,blockage).
```

The energy-based representation provides the generalised variables: flow, effort, momentum and displacement, which are replaced by: e, f, m and d respectively, using a lookup table. The latter is also used, to allow the user specifying other variables (e.g., acceleration is a flow derivative in the mechanical and hydraulic domains, pressure is proportional to the level ..., etc.).

```
correspond(hydraulic,leak,c,+).
correspond(hydraulic,pressure,e).
correspond(hydraulic,pressure,level).
...
```

3.2.2 Modelling Assumptions

The variety of model fragments of each component are due to the various modelling assumptions one can consider to represent a physical system. The user is allowed to state explicitly such modelling assumptions about the device at hand: an a priori set can be stated, using "Consider" predicates [B.Falkenhainer, K.D.Forbus 91]. Example: "consider the compressibility of the fluid".

These assumptions (approximations) are used to index the model fragments in the library. It means that, they are explicitly represented in each model (in terms of complexity degree).

When such information is available, AIM+ do not explore all the possible combinations of model fragments, but picks out those with the appropriate complexity degree.

3.3 Component's functionality

Ideally, a library of generic components should consist of "context-free" component models that adhere to the "no function in structure" principle. From a practical perspective, it is difficult to build models without any reference to the context of use.

As an example, a pump can be seen either as a source of effort or a source of flow depending on the context of use.

In order to preserve the principle of "no function in structure", and guarantee the reuse of the components library, the user is able to specify the intended functionality of each component.

At this moment, we are, only, considering the case of source components for whome it is difficult to say if there are sources of effort or flow. The user, can specify clearly what kind of source is a certain component, and AIM+ uses this information when assigning the causalities to the bond graph. If no specifications, the system explores all the possibilities.

3.4 The Library of generic Components

For each component we associate one or more models, from the simplest one to a more complex one. Complexity is defined in terms of the number of elements from which a model fragment is composed. The complexity of a whole model, will be the sum of the complexities of all its components.
For example, a motor can be represented by: (1) GY, (2) GY+R, (3) GY+R+C, (4) GY+R+C+I, (GY= gyrator, R= coil resistance, C= coil capacitance, I= coil inductance).

The following clauses are used to represent a motor:

. component(motor,[electric-mechanical], 1, description(input(A), output(A),[A],[A-gy])),
. component(motor,[electric-mechanical], 2, description(input(A),output(B),[A,B,C],[bond (A-gy,B-1), bond(B-1,C-r)])) ...

Each component is represented by:
- a name: the same one must be used in the device description,
- a list of domains: physical domains separated by comas to represent different perspectives, like: [hydraulic,thermal], or joined domains to represent a transformation from one domain to another as in the example of the motor,
- a description (bond graph): input and output of the bond graph, in order to be linked to other component' fragments, and a list of elements (generalised variables: e,f, ..., or junctions), and, finally, a list of bonds.

3.5 Model formulation

3.5.1 Model Selection

For each component, the model selection procedure consists in choosing the simplest model that doesn't contradict the set of the modelling assumptions. Initially, this set may include an a priori list of explicit modelling assumptions; otherwise, the selection procedure takes the simplest model of each component.
The selection task processes, then, further, choosing the next complexity degree of a certain component.
If we consider a device with n components, and that the highest complexity degree of one of them is p, then the search space will cover all the p^n combinations. Fortunately, these combinations are not explored totally, and the user can choose between two possibilities: looking for the best model (the most parsimonious one), or looking for the first best model.
In the first case, only models with complexity degree lower than the actual best model are constructed (Branch-and-bound search).

3.5.2 Model Composition

Given model fragments of the different device components, the model composition task consists of analysing the structure description, to compose the whole bond graph. AIM+ uses the following rules:

- a connection between two components is considered as a serial one: connection(domain, [o1], [i1]);
- a serial connection is represented by a bond relating the model fragments of the two components;
- a connection involving many-to-many components: connection(domain,[o1, o2, ..., on], [i1,i2, ..., in]), is considered as a serial connection between the two lists of components, and as a parallel one between the components of each list;
- a parallel connection is represented by a junction (0 or 1 depending on the domain);
- when a component is declared in the list of inputs (structure description), a source of effort or flow is added to it's model fragment (exogenous variable);
- when a component is declared in the list of outputs, a resistive element is added to it's model fragment;
- using active bonds (information bonds) between the different perspectives of the same component.

The composition task will produce the following bond graph:

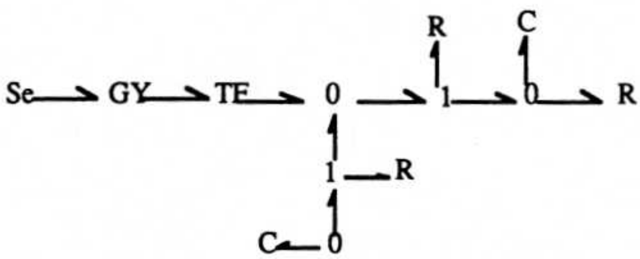


Figure 4: A first (generic) bond graph

At this stage, and before assigning causalities, AIM+ can handle the specifications of intended functionalities given by the user, in order to eliminate some possibilities. For example, the pump can be specified as a source of effort, and thus, the causality bares have to be set consequently. The model of figure 4 can not offer this possibility and a causal conflict is detected. The modelling loop permits to select a mōre complex model fragment for the voltage component, so that the causalities are well assigned. We obtain the model of figure 5.

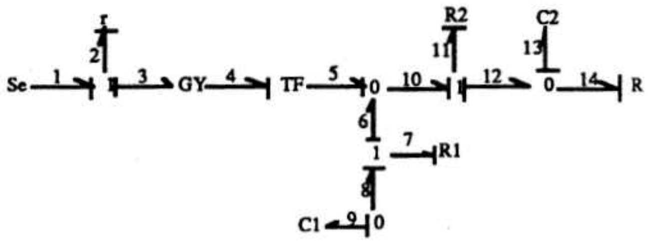


Figure 5: A first bond graph with no causal conflicts

3.5.3 Model verification

In a nutshell, the purpose of verification is to get confident about the device model.
This is crucial to handle the diagnosis task: when a discrepancy between what is observed and what is intended is detected, there is no doubt that something is wrong with the device, so we never incriminate the model in use.
For the purpose of verification as well as diagnosis, a causal graph is derived from the bond graph (see annex). We can now provide the following definition:
A model is said to satisfy the "premises ⇒ conclusion" constraint if :

"premises" are possible causes obtained from the derived tree of "conclusion".
This tree is generated from the causal graph associated to the bond graph.

In our example, the verification task must find the model which satisfies all the constraints.
Let us consider a constraint: "When output flow of pipe2 is blocked, the output flow of tank2 will decrease", represented by:

```
constraint(tank2,flow,-) :-
        constraint(pipe2,blockage).
```

The set of possible causes of this "conclusion" is {R+, C2+, R1+, r+, R2+, C1+}, where R2+ (interpreted as a blockage in the hydraulic domain, thanks to a look up table) corresponds to the premise specified in the constraint. The model, thus, satisfies this constraint.

3.6 Diagnosis

The diagnosis task aims at retrieving the primary causes that explain the deviation observed on the symptom variable. It is composed of, mainly, two tasks: candidates generation and candidates discrimination.

3.6.1 Candidates generation

We derive a tree whose root is the symptom variable and whose leaves are candidates or contradictions. Candidates are represented by component parameters (bond graph elements: R, C, I). AIM+ proceeds to a symptom analysis task at a qualitative level. It deals with deviation signs of variables which are represented by a Deviation Index (DI in the rest of the paper): [DI(x)]={-, 0, +}, means, respectively, that x is: below normal, normal and above normal.
We are, thus, interested in the signs of the causal graph edges: S_{xy}={+, -} depending on whether the cause variable x and the effect variable y change in the same direction or not. Given [DI(y)], we calculate [DI(x)] using the following qualitative constraint:

$$[DI(x)] \otimes S_{xy} = [DI(y)] \qquad (1)$$

$\otimes$ represents the qualitative multiplication.

A formal (or symbolic) execution using (1), and a backward chaining on the causal graph, permit to derive a tree representing all the possible explanations of the observed deviation on the symptom variable.
Let us Consider the following simple example of figure 6:

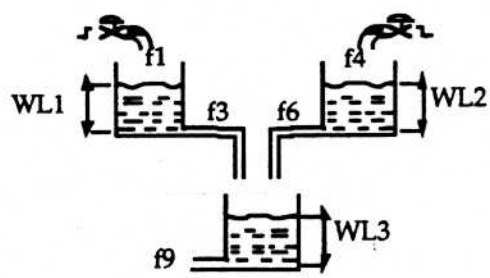


figure 6: a simple three tanks system

A graph corresponding to the deviation: [DI(f3)]=- (or f3=- to be more concise), is represented in Figure 7:

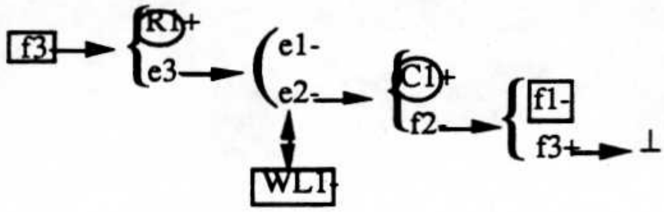


Figure 7: A tree corresponding to f3- symptom

Variables bracketed together are in an exclusive OR (we rule out the double fault hypothesis), whereas the parenthesis represents an AND. Arrows represent the backward chaining process on the causal graph, whereas the double arrow represents the equivalence relation between an effort (or flow) variable with a measurable variable. The latter is squared, and the candidates are circled.

Candidates={R1+, C1+}:

. R1+: the outlet resistance increases (probably due to a blockage);

. C1+: the tank capacity increases (probably due to a leak).

This is a first explanation level: each component is represented by a main parameter (bond graph element), as for example the tank is represented with a C element, and the candidate indicates the deviation of this parameter which can explain the observed deviation on the symptom variable. Moreover, each main parameter is related to other component parameters (e.g., $C=S/g.\rho$), we can, thus, process further when replacing each parameter deviation (e.g., C1+) by the disjunction of deviations of the rest of the component parameters (e.g., S+, ρ-) consistent with its deviation.

3.6.2 Candidates discrimination

This stage proposes to select the best measurement, which can make the best discrimination among the candidates.

We suppose that the measurable variables are well known; for each of them we derive a causal tree as done for the symptom variable. We look for the set of candidates for each measurable variable as it deviates from its nominal value in a given way (- or +) or even when it behaves normally (0). We obtain, thus, three sets (let the measurable variable be: x): Causes(x=+), Causes(x=-) and Causes(x=0). One causal tree (for x=- or x=+) suffices to obtain these results, as the other ones can be derived from it.

In each case, we obtain a new set of candidates when intersecting Causes(x=+, - or 0) with "Candidates", and we use a criteria named PDD (Power of Discrimination) to represent the number of candidates eliminated from the original set. We obtain thus: PDD(x=-), PDD(x=+) and PDD(x=0). A simple way to calculate PDD(x) is, then, to consider the mean value of these three PDDs.

Doing the same with all the measurable variables, we obtain a partial ordering of them, and the discrimination stage consists, then, in proposing the measurable variable whose PDD is the maximum.

If two variables have got equal PDDs, we choose the one with the greater partial PDD (that of x=-, + or 0).

Let us consider the following case: the observed variables are: f9=-, f1=f4=0. The corresponding candidates set is:

Candidates={C1+, R1+, C2+, R2+, C3+, R3+}.
Moreover, WL1, WL2, WL3, f3 and f6 are the measurable variables. We can determine that:
Causes(WL1-)={C1+, R1-}, PDD(WL1-)=5;
Causes(WL1+)={C1-, R1+}, PDD(WL1+)=5;
Causes(WL1=0)={C1 normal, R1 normal}, PDD(WL1=0)=2. Finally: PDD(WL1)=4.
In the same way, we have got: PDD(WL2)=4, PDD(f3)=PDD(f6)=PDD(WL3)=2.
The measurement of WL1 or WL2 may, thus, provide the best discrimination among candidates.

After a new measure is taken, and whatever is its result, AIM+ has already processed the new candidates set (as it always anticipates this task) and will, thus, suggests a new measurement to the operator. AIM+ stops, when a unique candidate remains in the set of candidates, or no more measurements can be taken.

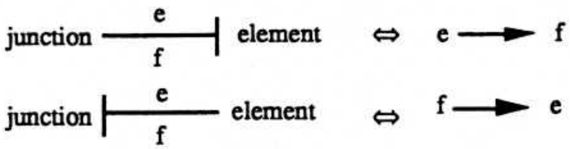
4. Conclusion

Our focus in this paper was on providing a formal framework for automatic modelling and diagnosis. We used the bond graphs modelling methodology to compose a model given a structure description of the device and a library of generic components. Models are checked to be consistent with a set of behaviour constraints provided by a user. For the diagnosis task, we make the steady state hypothesis for the physical system, so a symptom is seen as a deviation from a nominal value. The result of the diagnosis process is a set of candidates and the best following measure which can be made.

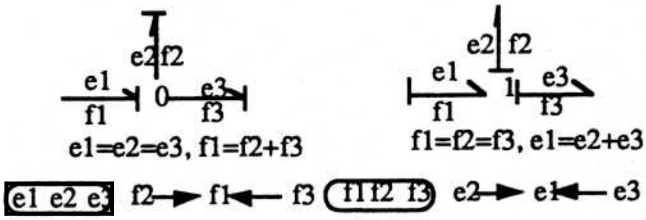
Annex

Causal graph

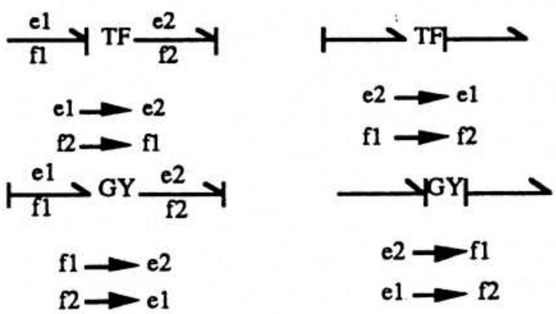
Associated to each bond graph is a causal graph, which is used in both verification and diagnosis task. It is built using the following simple rules:
For the R, C and I elements we apply the following simple rule:



For the junctions (1 and 0): variables with equal values are circled together.



For transformers TF and GY:



Furthermore, each graph edge is labelled with a signed coefficient representing the relation between the two vertices.

References

J. Amsterdam (1993) "Automated Qualitative Modelling of Dynamic Physical Systems" PhD Thesis, MIT AI Lab.

G. Biswas, S. Manganaris, X. Yu (1993) "Extending Component Connection Modeling for Analyzing Complex Physical Systems" IEEE Expert, p. 48-57, February 1993.

L. Chittaro, G. Guida, C. Tasso, E. Toppano (1993) "Functional and Teleological Knowledge in the Multi-modelling Approach for Reasoning about Physical Systems: a case study in diagnosis" IEEE Transactions on Systems, Man and Cybernetics, p. 1718-1751, Nov/Dec 1993.

B. Falkenhainer, K.D Forbus (1991) "Compositional Modeling: finding the right model for the job" Art. Int. 51, p. 95-143, 1991.

D.W. Franke, D.L. Dvorak (1994) "CC Language" Technical Report, University of Texas at Austin.

Y. Iwasaki, A.Y.Levy (1994) "Automated Model Selection for Simulation" KSL-Report, Stanford University.

P.P Nayak (1994) "Causal Approximations" Art. Int.70, p. 277-334, 1994.

Reiter (1987) "A Theory of Diagnosis from first principles" Art. Int. 32, p. 57-95, 1987.

R. Rosenberg, D. Karnopp (1983) "Introduction to Physical System Dynamics" McGraw-Hill, NY, 1983.

D.S Weld & J. de Kleer (1989) "Reading in Qualitative Reasoning About Physical Systems" San Mateo, Morgan Kaufmann.

S. Xia, D.A. Linkens, S. Bennett (1993) "Automatic Modelling and Analysis of Dynamic Physical Systems using Qualitative Reasoning and Bond Graphs". Intelligent systems engineering journal,
p. 201-212, Autumn 1993.