Modeling as a Fragment Assembling Process

H. Ahriz
School of Computer and Mathematical Sciences,
Robert Gordon University,
Aberdeen AB25 1HG, U.K.
ha@scms.rgu.ac.uk

M. Tomasena
SysCom, Université de Savoie,
Le Bourget du Lac, 73376, France.
tomasena@univ-savoie.fr

Abstract

Model based reasoning about physical systems deals with diagnosis, supervision, interpretation, explanation, etc. Most of the contributions to this domain do not pay much attention to model construction, and it was generally accepted that a model was available or could be easily obtained. This assumption is no more valid when we tackle real industrial problems rather than toy examples. The situation is even worst since there is no standard methodology or approach to making models. In this paper we provide a framework to elaborate models which are suitable for model based reasoning in general, and for fault diagnosis in particular. The framework relies on the bond graphs notation, which allows a uniform approach for the different physical domains and offers compositional view of the system.

Introduction

It is generally accepted that three stages of work are involved in the model-based approach to analysing a system [Weld & de Kleer 1989]. 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 ([Dague & al. 1987], [Falkenhainer & Forbus 1991], [Nayak 1994]).
Traditionally, models are constructed by hand and are then used in experiments to ensure acceptable results. Models produced in this manner tend to include everything, including issues irrelevant to an application, and require solid competencies in mechanical, hydraulics, electricity and thermodynamics. These considerations indicate the need for new approaches to modeling, based on more rigorously defined modeling processes. Automated modeling is such an approach [Xia & al. 1993]. It attempts to generate models, which are parsimonious and accurate for model-based reasoning.

Few research works have already addressed this issue, or part of it. We can quote GoM (Graph of Models) [Addanki & al. 1991], which represents a collection of models built by an expert in a particular domain. The collection is represented in terms of a graph in which each node represents a model, whereas an edge is labeled with an assumption (simplification or refinement). The Prompt

system [Weld 1992] allows for navigation through this kind of graphs. A most significant work is CM (Compositional Modeling) [Falkenhainer & Forbus 1991], which is devoted to generate qualitative and quantitative answers to queries about physical systems. Other works have been derived from the latter, namely "Automated Model Selection for Simulation" [Iwasaki & Levy 1993] and "Causal Approximations" [Nayak 1994]. MM [Amsterdam 1993], is also a very related work to ours since he was introducing bond graphs as a modeling language. Biswas and Yu [Biswas & Yu 1993] propose a formal modeling scheme that also use bond graphs as modeling language. The main distinction of our approach is its non-deterministic nature. Actually we consider that modeling process requires the exploration of a search space. This search space could have several solutions (i.e. models), could accept several cost criteria (e.g. parsimonious notion) and could be explored with various search strategies. The explicit use of modeling hypotheses and behaviour constraints is a means to limit the exploration of the search space.

Our work intends to introduce more automation in the different modeling tasks, and led to the system: AIMD (Automated Intelligent Modeller for Diagnosis). Modeling and diagnosis are the two main functions of AIMD. In this article we focus on modeling, the diagnosis process is outside the scope of the present article. In section 2 we introduce a case study that is used along this paper. In section 3 we present an overview of the modeling approach, process and language. Section 4 describes what kind of generic knowledge AIMD is going to use, while section 5 describes the knowledge specific to a particular scenario. The different tasks constituting the modeling process are presented in detail in section 6. Section 7 presents the results obtained on our case study.

Case study

Before going through the details of our work, let's give an overview of what kind of results AIMD is expected to provide given some specific entries. Let us consider along this paper the following pump system[1] as a case study (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.

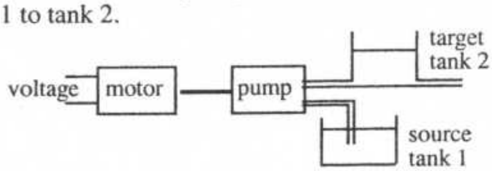


Figure 1: Case study system

The specific inputs (scenario dependent) for the pump system are:

1. A structure description of the pump system. Figure 2 illustrates a schematic description of the structure. The structure describes a decomposition of the system in terms of primary components, which are related by physical (hydraulic, mechanical, electric) connections.
2. A set of hypotheses (eventually empty) about the components of this system. An example of a hypothesis is "consider the friction in the motor".
3. Optionally, an expected behaviour of the system, like "when the source tank becomes empty, the motor speed increases". The behaviour is given in terms of qualitative constraints. The user has the choice of the variables and the interval of time to be considered.

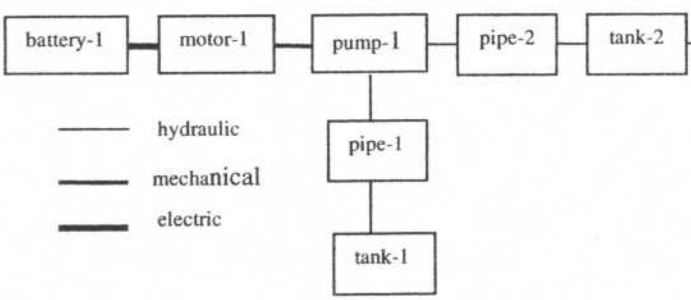


Figure 2: Schematic description of the case study

system

Given such entries, AIMD is able to elaborate a parsimonious model representing the system. This model is given in terms of a bond graph, a set of qualitative and quantitative equations and a causal graph. We describe later the details of the obtained model. This result is in turn exploited by AIMD to perform diagnosis, proving in this way the adequacy of the model. Diagnosis is achieved trough a dialog between AIMD and the user. Previously, the set of measurable variables of the system must be specified. An example of a diagnosis session for the case study is the following:

```
USER > observation: flow of (tank-2) is
below_normal.
AIMD > candidates: (r-3, +), (c-6, +),
(r-7, +), (c-4,-)[2]
what is the value of - pressure - tank-1?
(above, below, normal)
USER > bellow.
AIMD > a single candidate[3] remains:
(c-6, +)
Interpretation of diagnosis:
(tank-1, leak)[4]
```

Modeling Framework

Overview of the Modeling Process

The modeling process is based upon the consideration of two groups of inputs, which feed the modeling process: the scenario dependent ones, and the scenario independent ones (by scenario we mean the modeling session tackled by the designer). As stated before, this group is constituted by the description of the physical system, a set of modeling hypotheses and a set of behaviour constraints, whereas the second group is constituted by a library of generic model fragments, as long as other generic knowledge concerning physical systems. Observations and measurements from the physical system concern the diagnosis part.

The modeling process consists of the following tasks (see figure 3 below):

1) Fragment selection. Each component has a set of model fragments stored in a library. Successive selections are made increasing the degree of complexity[5] of fragments starting with the least complex fragments.

2) Fragment assembling. Fragment assembling is made according to the structural description and some compositional rules.

[1] This test-case has been introduced for the first time in [Xia & al. 1993]

[2] r-3, c-6, r-7, c-4 represent internal variables of the model. We describe later the correspondence between internal variables and the parameters and measurements of the system. - and + stand for below and above normal respectively.

[3] The discrimination capacity of the diagnosis depends on the number of measurable variables and their position on the system.

[4] (c-6, +) stands for: the capacity of tank-1is above normal. One possible interpretation of this fact is a leak in the tank.

[5] The complexity degree will be defined latter.

3) Model verification. The purpose of this task is to verify that the obtained model could really exhibit the expected behaviour. This is done by comparing the simulated behaviour (using the qualitative equations obtained in the precedent task) to the expected one.

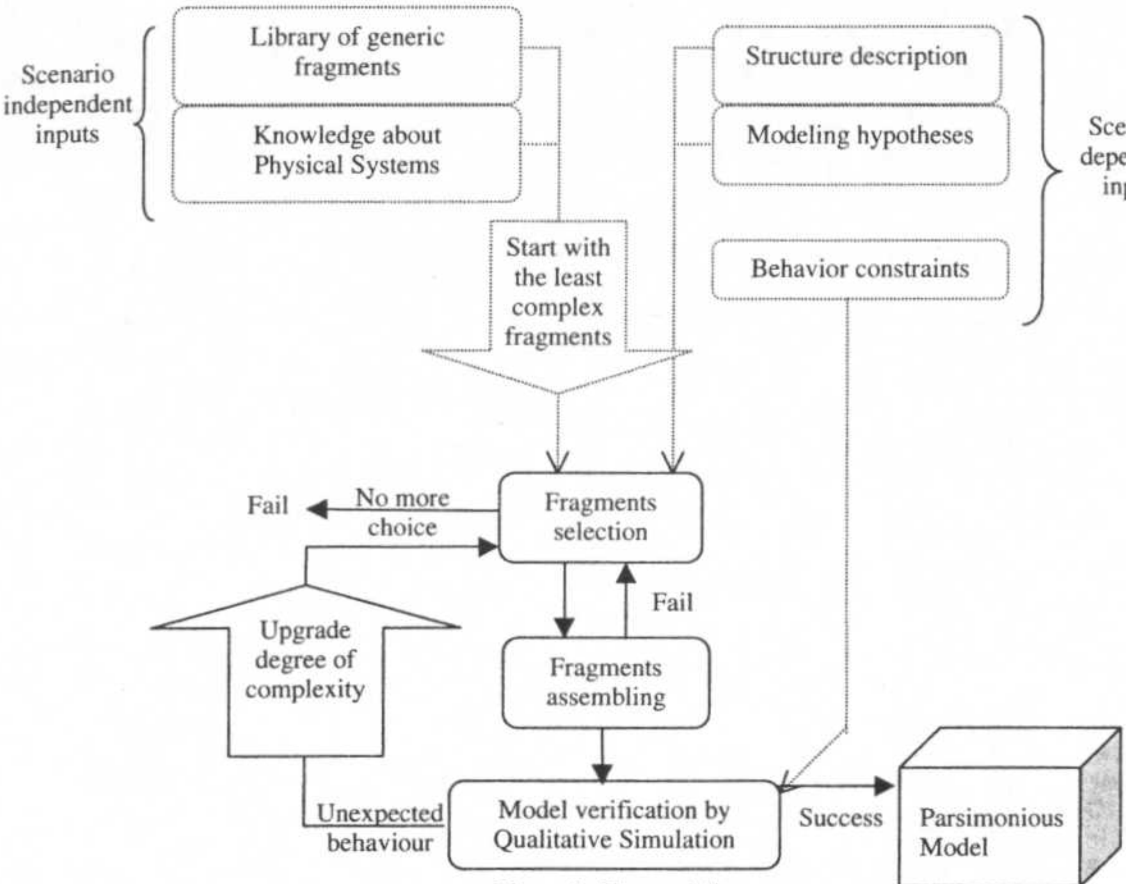


Figure 3: The modeling process

The diagnostic function is intrinsically related to the modeling function and they are integrated in AIMD. The model produced from the modeling process will be used as a reference against any malfunction on the physical system.

Modeling language

Bond graphs [Rosenberg & Karnopp 1983] are based on modeling energy flow, power, between system components and inherently enforce continuity of power and conservation of energy. This provides a systematic framework for building consistent and well-constrained models of dynamic physical systems across multiple domains (e.g., electrical, mechanical, hydraulic). Bond graphs rely on effort variables (e) to represent generalised voltage, pressure, temperature, etc., and on flow variables (f) to represent generalised current, volume flow, entropy flow, etc. The topological character of bond graphs allows for compositional modeling and makes them directly applicable to qualitative processing. This renders them useful in situations where precise numerical information may not be available. However, analytic system models derived from bond graphs are also amenable to quantitative simulation and analysis. Furthermore, bond graphs embody a direct relation between state variables and physical component parameters, and their causality constraints provide the mechanisms for effective and efficient diagnosis. More detailed presentation of bond graphs are given in [Rosenberg & Karnopp 1983].

Implementation

As we pointed out, the modeling approach aims to be modular and declarative. On the other hand the nature of modeling is intrinsically non-deterministic. In fact the three tasks of the modeling process are highly non-deterministic. For these reasons Prolog was used for implementation, this choice allows for a declarative representation of the different kinds of knowledge in terms of logical relations and is naturally adapted for the exploration of a search space. A counterpart of this choice is the performance of execution, but this point becomes secondary since the model construction part of AIMD is made off line.

Scenario-independent inputs

Library of fragments

Ideally, a library of generic components should consist of "context-free" component models that adhere to the "no function in structure" principle [de Kleer & Brown 1984]. The definition of the library of components respects this principle. This is possible since the modeling process takes into account, explicitly, other sources of knowledge. That means that for a given component the fragment selection task could pick up one specific fragment of model in the library even if this selection is aberrant from a global point of view. That doesn't matter, the fragment will be rule out in the successive tasks and other fragment of the same component will be considered.

Each component, in a given domain, has one or more associated model fragments, from the simplest one to a most complex one. Complexity is defined as the number of bond graph elements from which a model fragment is composed. It forms a partial order relation.

For example, a motor can be represented by 5 model fragments, one of complexity 1 (GY), one of complexity 2 (GY+R), two of complexity 3 (GY+R+C and GY+R+I) and finally one of complexity 4 (GY+R+C+I). The previous elements stand for GY= gyrator, R= coil resistance, C= coil capacitance, I= coil inductance.

The complexity of a whole model, will be the sum of the complexities of all its fragments.

The following "`component`" predicates are used to represent these model fragments of a motor in the electric-mechanical domain:

```
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)])).

component(motor, electric-mechanical, 3,
   description(input(A),output(B),
   [A,B,C,D],[bond(A-gy,B-1),bond(B-1,C-r),
   bond(B-1,D-c)])).

component(motor, electric-mechanical, 3,
   description(input(A),output(D),[A,B,C,D,E],
   [bond(A-1,B-i),bond(A-1,C-gy),bond(C-gy,D-
   1), bond(D-1,E-r)])).

component(motor, electric-mechanical, 4,
   description(input(A),output(B),
   [A,B,C,D,E],[bond(A-gy,B-1),
   bond(B-1,C-r),bond(B-1,D-i),bond(B-1,E-
c)])).
```

Each fragment is represented by:

- a name: the same one must be used for the component in the device description;
- a domain: simple physical domain, or joined domains to represent a transformation from one domain to another as in the example of the motor. A component could be presented in different domains. For example a tank could be present in the hydraulic domain and in the thermal domain;
- an integer representing the fragment complexity;
- a description (i.e. bond graph) composed by: (a) input and output of the bond graph, in order to be linked to other component' fragments, (b) the list of Prolog variables each of which refers to a node in the bond graph, and, finally, (c) a list of bonds between nodes. A node is represented by a couple name-type, where name is a Prolog variable and type is the type of the element (i, c, r, gy) or the junction (1, 0). For example the fourth fragment of the motor corresponds to the bond graph of figure 4.

```
component(motor, electric-mechanical, 3,

description(input(A),output(D),[A,B,C,D,E]
        [bond(A-1,B-i), bond(A-1,C-gy),
         bond(C-gy,D-1),bond(D-1,E-r)])).
```

Figure 4: A fragment model of the component: motor

Knowledge about modeling hypotheses of physical systems

As we mentioned before, a component may have several model fragments corresponding to various situations of use, and depending on the presence or not of particular physical phenomena.

In each model fragment there is an indication on the subjacent modeling hypothesis implicitly used. This indication is obtained from the elements presented in the bond graph. All we need to do, thus, is to provide a correspondence about these elements and the physical phenomena. This is achieved by defining a set of « `corresponds` » relation. Some of these relations are given below:

```
corresponds(friction,r).
corresponds(dissipation,r).
corresponds(compressibility,c).
```

Furthermore, we don't need the library to contain, for a particular component, all the possible model fragments. A tube, for example, can be represented either by one of the following elements: C, I, or R, which correspond to three modeling hypotheses, or by one of their

combinations (see figure 5 below). Actually, these hypotheses do mean respectively:
H1: compressible fluid (flexible walls); H2: inviscid liquid (long and narrow tube); H3: viscous liquid (rough walls).

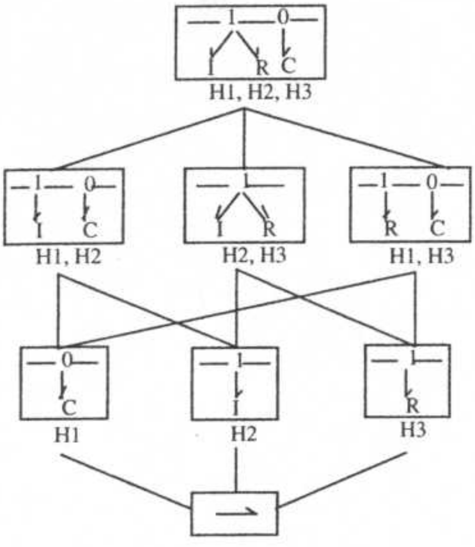


Figure 5: Graph of the possible model fragments of a tube

It appears that beyond the second level of the above graph (starting from the bottom), it is unnecessary to encumber the library with the rest of the fragments, which can be obtained by combining the basic ones. This choice constitutes an improvement compared to the graph of models in [Addanki & al. 1991], which enumerates and represents all the combinations. The mutually exclusive hypotheses are grouped in a same class [Falkenhainer & Forbus 1991], in such a way that all the combinations involving both hypotheses are discarded. For example:

```
class(tube,[corresponds(viscosity,r),
      corresponds(compressibility, c)]).
```

Scenario dependent inputs

The device structure

The device structure representation is an abstracted view (model) of the physical system. It is Component-Connection oriented, and, thus, contains the description of system components, relations (including connections) between component terminals, and the specification of the inputs as well as the outputs of the system.
Let us, consider the structure description of our case study presented in figure 1. The declarative description (equivalent to the schematic description of figure 2) is given below:

```
input ([]).
output (tank2-hydraulic).
set_of_relations([
connection(electric,[battery-1],[motor-1]),
connection(mechanical,[motor-1],    [pump-1]),
connection(hydraulic,  [pump-1],    [pipe-2]),
connection(hydraulic,  [pipe-1],    [pump-1]),
connection(hydraulic,  [tank-1],    [pipe-1]),
connection(hydraulic,  [pipe-2],    [tank-2])
            ]).
```

In addition to the "`connection`" relation, other kinds of relations can be used in AIMD, to allow someone to represent relations such as a heat transfer.

Modeling Hypotheses

The variety of model fragments of each component are due to the various modeling hypotheses one can consider when representing a physical system. The user is allowed to state explicitly such modeling hypotheses about the device at hand: an a-priori set can be stated, using "`consider`" predicates like in [Falkenhainer & Forbus 1991]. For example: "consider the friction in the motor", which is represented as following:

```
consider(mechanical,friction, motor-1).
```

These hypotheses are used to index the model fragments in the library. It means that, they are explicitly represented in each model (in terms of bond graph elements). When such information is available, AIMD do not explore all the possible combinations of model fragments, but picks out those with the appropriate elements to meet hypotheses. In our case study, since there exist a knowledge that states a correspondence between friction and "r", AIMD will consider only the model fragments associated to the motor component in the mechanical domain that contain the element "r" in their description.

Behaviour Constraints

In addition to the description of the system's structure and the modeling hypotheses, inputs to AIMD could include a set of behaviour constraints. A behaviour constraint describes in qualitative terms one possible dynamic behaviour of some device variables.
The representation of these expected behaviours is done through a "`constraint`" predicate:

`constraint(<component>,<variable>,`
`<segment>).`

It specifies the physical component and the concerned variable within it, as well as an ordered list of couples (value, derivative) for this variable, describing its expected dynamic behaviour in qualitative terms (segment). The qualitative space considered in our study is {-, 0, +}.
For example, the following constraint: "when the source tank becomes empty, the motor speed increases", is represented by the couple of constraints:

```
constraint(motor-1,speed,
           [(0,+),(+,+),(+,+)]).
constraint(tank-1,volume,
           [(0,-),(-,-),(-,0)]).
```

Figure 6 shows graphically the evolution of these variables.

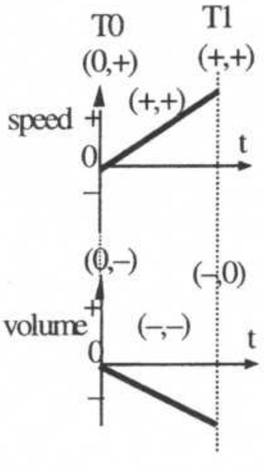


Figure 6: Graphical representation of the pump constraints

Each segment is a succession of time points and time intervals, such as in QSIM representation [Kuipers 1986]. All the couples of values for each variable are represented for the same times.
Each time point has the 0 value for at least one variable or for his derivative. This point time corresponds to a change for the sign or evolution of a variable.

Modeling Process

The modeling process consists of the following tasks:

Selection of model fragments

The inputs to this task are the structural description of a system to be modelled and a set of modeling hypotheses. For each component in a given domain, the model selection procedure consists in choosing the simplest model that doesn't contradict the set of the modeling hypotheses. Initially, this set may include an a-priori list of explicit modeling hypotheses; otherwise, the selection procedure takes the simplest model of each component. Successive selections are made increasing the degree of complexity of fragments starting with the least complex ones.
If we consider a device with n components, and an average complexity degree of p, then the search space will cover all the p^n combinations. Fortunately, these combinations are not explored totally, and AIMD allows for the application of two research strategies. The modeling process could produce either:

- The first parsimonious model (least complex) satisfying the criteria, by the application of a branch and bound search, or
- The first model (not necessarily parsimonious) satisfying the criteria by the application of a depth-first search. The depth-first search means that one component, at a time, has to be made more complex.

Fragment Assembling

Fragment assembling is made according to the structural description and some compositional rules. Once the assembling is performed, AIMD analyses the whole model to detect if it could be acceptable from the point of view of causality (i.e. the obtained model must have a plausible causality). If it is not the case, a component is chosen to be altered and its actual fragment is replaced with the next more complex one. The obtained model is given in terms of a bond graph (with an affected causality) from which a set of qualitative equations is derived.
Furthermore, AIMD uses the following compositional rules:

1. 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;
2. 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;
3. a parallel connection is represented by a junction (0 or 1 depending on the domain);

4. 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);
5. when a component is declared in the list of outputs, a resistive element is added to it's model fragment;
6. when different perspectives (domains) are possible for the same component, an information bond is introduce between the different points of view.

At this stage, AIMD tries to assign the causality bars to the bond graph. This procedure, described in [Rosenberg & Karnopp 1983] and in [Top & Akermans 1991], may lead to two cases:

1. a conflict of causalities: we must, thus, loop (backtrack) to the selection task to pick up other fragments (we choose a more complex fragment for one component);
2. the procedure is successful: we continue with the next task (if there are many solutions, we have to cope with all these possible models (i.e. this represents a non deterministic point during the modeling process).

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, a set of qualitative differential equations is derived from the bond graph. We can now provide the following definition: *A model is said to satisfy a (or a set of) behaviour constraint(s), if we find a matching between one of the possible simulated behaviours and the expected one.*

In order to be able to compare simulated behaviours with the expected one, AIMD uses a table of correspondences between external variables used to state the behaviour constraint (like speed) and the internal variables of the bond graph (like "f"). Some or this correspondences are described in table1.

```
correspondence(hydraulic, pressure,    e).
correspondence(hydraulic, flow,        f).
correspondence(mechanical, force,      e).
correspondence(mechanical, speed,      f).
correspondence(electric, tension,      e).
correspondence(electric, courant,      f).
correspondence(thermal, temperature,   e).
correspondence(thermal, energy_flow,   f).
```

TABLE 1: Correspondence relation

We use a QSIM [Kuipers 1986] like simulation in order to simulate the behaviour(s) of a model. Adapting to the bond graph formalism, we elaborated the following qualitative differential equations (QDEs):

- add(Y, [(X_1, s_1), ..., (X_n, s_n)]): represents the sum of efforts or flows in a junction,
  $Y = \sum (s_i) X_i$, where s_i are the signs (+, –) of each X_i variable;
- equal([X_1, ..., X_n]): represents the equality of efforts or flows in a junction, $X_1 = \ldots = X_n$;
- int(Y, X): integration relation used in the case of a C or I element (Y and X are either efforts or flows), $Y = \int X\,dt$;
- mon(Y, X): a monotonic function used for a R, TF or GY element.

We adopt the time alternation between time points and time intervals [Kuipers 1986], and adapt it to the qualitative variables domain {–, 0, +}[6]. As a result, we obtain 15 P-transitions and 15 I-transitions which are valid between each ([x], [x'])$_1$ state and a next ([x], [x'])$_2$ state.

The verification algorithm can be described as following:

1. Input:
   - the set of qualitative differential equations (obtained from the bond graph);
   - the expected dynamic behaviour (segment);
   - a partial initial state (represented by the first state of the segment).
2. Simulation/Comparison:
   - If the current (or initial) state is incomplete[7]
     Then complete this state by propagating its values through the QDEs;
   - A next state is determined following these stages:
     a) We apply the simulation algorithm inspired from QSIM [Kuipers 1986] to produce a set of new current qualitative states;
     b) Compare the current state in the segment with the current qualitative states. Eliminate those states from the current qualitative states, which do not match the segment.
3. Output/Loop:
   - If the set of current qualitative states is empty (i.e. there is no state that can match

[6] We also allow the "?" value.

[7] This is the case in most of the situations, either because we've got "?" values, or because the segment does not concern all the system variables.

the current state in the segment) then verification fails and the process backtracks to the last non-deterministic point,

- Else, if the simulation reaches a state corresponding to the last state in the segment, then stop, the model satisfies the constraints,
- Else go to 2 (Simulation/Comparison).

Results

Let us consider several modeling scenarios of our case study. Each modeling scenario corresponds to a combination of modeling hypotheses and behaviour constraints, which are as following:

Modeling hypotheses:

Hy1: `consider(dissipation,hydraulic, pipe-1).`

Hy2: `consider(dissipation,hydraulic, pipe-2).`

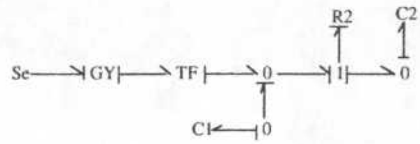
Behaviour constraints:

Cst1: `constraint(pressure,tank-1, [(0, -), (-, -), (-, 0)]).`

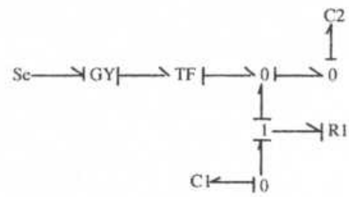
Cst2: `constraint(flow,tank-2, [(0, +), (+, +), (+, 0)]).`

Table 2 summarises the results obtained for each scenario (we present here 8 scenarios). For each scenario AIMD produce a parsimonious model. The crosses mean that a hypothesis or a constraint has been considered:

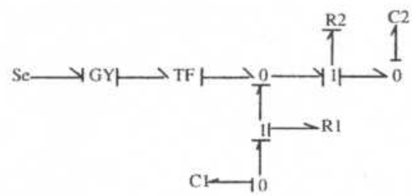
Models obtained for the scenarios are shown below. We remark that different scenarios may give rise to identical models: a) for scenarios 1, 4 and 7, b) for scenarios 2, 5 and 6, and finally c) for scenarios 3 and 8. These models are presented in figure 7.



— Model a —



— Model b —



— Model c —

Figure 7: Obtained models

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8
Hy1		×	×			×		×
Hy2			×				×	×
Ct1					×			×
Ct2				×	×	×	×	×
Time (sec)	1,18	0,62	1	4,22	400,87	14,68	3,80	28,93
Inferences	23493	12715	20998	73141	7368638	268714	64053	534983
Obtained Model	a	b	c	a	b	b	a	c

TABLE 2: Modeling scenario

To ease the understanding of the graphs, note that C1 and C2 represent tank 1 and tank 2, R1 and R2 represent pipe 1 and pipe 2, TF represents the pump and GY the motor, whereas Se is used for the tension.

These modeling scenarios were very instructive, and as we can see from the table 2, we can point out the following conclusions:

- As much as we consider modeling hypotheses, as fast as is the modeling process. Indeed, the model fragments are, in that case, picked up more accurately;
- A modeling hypothesis may be equivalent to the specification of a behaviour constraint (the same model can be obtained). As a future work, we are going to look closely to the relation between behaviour constraints and hypotheses in order to avoid simulation when an equivalence can be found;
- Even in the absence of modeling hypotheses and behaviour constraints, the modeling process doesn't lead necessarily to the simplest model, as the causality assignment can fail when applied on the latter. It appears, then, that the causality assignment is, implicitly, considering modeling hypotheses, which may be forgotten by the user of AIMD. This characteristic is quite interesting, as we are able to present the missing hypotheses to the user.
- The model verification, and particularly the qualitative simulation, is the most time consuming task;

Diagnosis session presented in section 2 is based on the use of model c and its derived causal graph (figure 8). For more details about the diagnosis task as handled by AIMD the reader can refers to [Ahriz & Xia 1997]). Others causal-model based diagnosis systems are described in [Mosterman & Biswas 1996], [Tomasena & al.1992] and [Console & al. 1989].

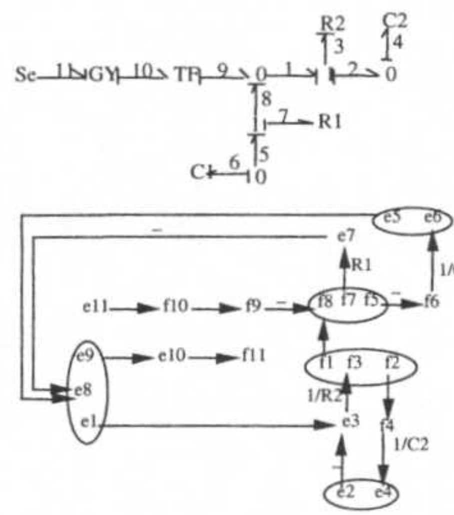


Figure 8: Bond graph and causal graph corresponding to model c

Conclusion

The Compositional point of view of the modeling task is the basis of our modeling framework. This approach requires first to break a physical system into smaller parts (components) and then to assemble the system from the parts. Bond graphs modeling greatly facilitates this requirement since it reposes on the structure of the system and offers an uniform formalism for the definition of generic component models which is an important step through a library of reusable models. The nature of our modeling approach is intrinsically non-deterministic and requires the exploration of a search space. Different models are checked to be consistent with a set of behaviour constraints and modeling hypotheses provided by the user. Results show that there is a close relation between behaviour constraints and modeling hypotheses, further studies are necessary to understand this relation in order to avoid simulation during the model verification.

References

Addanki, S.; Cremonini, R.; and Penberthy, J.S. 1991. Graphs of Models. *Artificial Intelligence* 51:145-177.

Ahriz, H.; and Xia, S. 1997. Automatic Modeling for Diagnosis. In *Proceedings of the Eleventh International Workshop of Qualitative Reasoning*, 3-13. Cortona, Italy.

Amsterdam, J. 1993. Automated Qualitative Modeling of Dynamic Physical Systems. Ph.D. diss., AI Lab, MIT.

Biswas, G.; and Yu, X. 1993. A Formal Modeling Scheme for Continuous Systems: Focus on Diagnosis. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1474-1479. Chambery, France. IJCAI, Inc.

Console, L.; Dupre, D.; and Torasso, P. 1989. A theory of diagnosis for incomplete causal models. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1311-1317. Detroit, USA. IJCAI, Inc.

Dague, P.; Raiman, O.; and Deves, P. 1987. Troubleshooting: when modeling is the difficulty. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 600-605. AAAI Press.

de Kleer, J.; and Brown, J.S. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence* 24: 7-83.

Falkenhainer, B.; and Forbus, K.D. 1991. Compositional Modeling: finding the right model for the job. *Artificial Intelligence* 51: 95-143.

Iwasaki, Y.; and Levy, A.Y. 1993. Automated Model Selection for Simulation, Technical Report, KSL-93-11, Dept. of Computer Science, Stanford Univ.

Kuipers, B.J. 1986. Qualitative Simulation. *Artificial Intelligence* 29 (3):289-338.

Mosterman, P. J.; and Biswas, G. 1996. An integrated architecture for model-based diagnosis. In *Proceedings of the Seventh International Workshop on Principles of Diagnosis*, 167-174. Val Morin, Canada.

Nayak, P.P. 1994. Causal Approximations. *Artificial Intelligence* 70: 277-334.

Rosenberg, R.; and Karnopp, D. 1983. *Introduction to Physical System Dynamics*. McGraw-Hill.

Tomasena, M.; Belmeguenai, Y.; and Robles, J.P. 1992. Coping with complexity in continuous process diagnosis. In *Proceedings of the Fifth International Symposium on Artificial Intelligence*, 213-219. Spain.

Top, J.; and Akermans, H. 1991. Computational and Physical Causality. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1171-1176. IJCAI, Inc.

Weld, D.S.; and de Kleer, J. 1989. *Reading in Qualitative Reasoning About Physical Systems*. San Mateo, Morgan Kaufmann.

Weld, D.S. 1992. Reasoning about Model Accuracy. *Artificial Intelligence* 56: 225-300.

Xia, S.; Linkens, D.A.; and Bennett, S. 1993. Automatic Modeling and Analysis of Dynamic Physical Systems using Qualitative Reasoning and Bond Graphs. *Intelligent Systems Engineering Journal*, 201-212.