Handling Assumptions in Automated Modelling

Neil Smith

CSIMG, Cranfield University,
Shrivenham, Swindon, SN6 8LA, UK
neil.smith@rmcs.cranfield.ac.uk

Abstract

There are many different classes of decision made during the development of a model. If models are to be shown to be appropriate for the specified task, the information and assumptions upon which these decisions are based must be made explicit. This paper describes AIM, an automated modeller that makes all its decisions with explicit knowledge. AIM separates different types of modelling knowledge in its knowledge base to prevent the embedding of assumptions in the structure of the modelling knowledge. It also uses a novel modelling algorithm that does not rely on either external sources of information or the structure of the knowledge base to generate models. We show that AIM generates parsimonious models and discuss some of the constraints on this approach.

Introduction

The process of modelling is one of creating a suitable abstraction of the real world. This is done by the modeller regarding the referent system and deciding what simplifications and abstractions to make to produce a model. However, it is all to easy for the modeller to make unwarranted assumptions about what abstractions are appropriate during the production of the model. If such unwarranted assumptions are made, the end result is a model that may not be suitable for the task at hand. In order to ensure that such inappropriate models are not used, it is important that all the modelling decisions are explicit during the modelling process. By making all the decisions explicit, the justifications for each decision can be given. This allows the user of the model to have confidence in the final model, as they are certain that only appropriate (*i.e.* justified) decisions were taken during the development of the model. In addition, by examining the justifications for the modelling decisions, the domain of validity of the model (Williams & Raiman, 1994) can be estimated, which can allow the reuse of the model under appropriate circumstances.

In the next section, we examine the different modelling decisions that need to be made in the development of any model and we discuss some of the ways in which unwarranted assumptions can creep into models. The main body of the paper consists of a description of AIM, an automated modeller that addresses many of these problems. We discuss each of the different types of modelling decision in turn, and show how the architecture and algorithms of AIM ensure that all decisions are explicit. In addition, we show that the models generated are parsimonious. Finally, we discuss some limitations of AIM's approach.

Types of Assumption

There are several different types of modelling decision that need to be taken when a system is modelled; all of these must refer to the intended use of the model, to ensure that the generated model is useful. The most basic modelling decision is the selection of a model representation and reasoning formalism (Schut & Bredeweg, 1996) which defines how the referent system is to be regarded by the modeller, *e.g.* whether a component-centred or a process-centred analysis is to be undertaken. Once the most appropriate formalism has been selected, two sets of decisions must be made in parallel: the boundary of the model must be determined, and the referent system must be broken down into its component parts for modelling. Once the referent system has been broken down into the appropriate "lumps" (*e.g.* components or processes), the modeller must decide on the most appropriate ontology to use to represent each of these lumps (the is the equivalent to selecting assumption classes in a model composition system (Nayak & Joskowicz, 1996)). This process provides the information to guide the formation of the model boundary, both in terms of physical extent and the types of phenomena included in the model. As the model boundary is developed, new lumps are identified that require representation in the model. Finally, there are a variety of decisions to be made regarding the various simplifying assumptions that can be applied to the model (Weld, 1992; Williams & Raiman, 1994). Only when all of these decisions have been made is it possible to finally construct the model, which is then used to address the task in hand.

This analysis shows that there are a great many different types of modelling decision that need to be taken during the development of a model. Highly abstract decisions are made at the beginning of the modelling process, with the decisions becoming more concrete and detailed throughout the modelling process. It is necessary that all of these decisions be made explicitly by the modeller to ensure that the model's user has confidence in the final model.

However, not all automated modelling systems are capable of making all of these decisions. Many existing automated modelling systems, particularly model induction systems (*e.g.* Ironi & Stefanelli, 1994; Addanki *et al.*, 1991) have only addressed the problem of selecting the correct simplifications to make in the model. Model composition systems (*e.g.* Nayak & Joskowicz, 1996) have addressed the problem of the selection of the most suitable ontologies to use in the model, but at the cost of increasing the complexity and decreasing the flexibility of their mod-

elling knowledge; this has reduced these modeller's ability to select appropriate simplifications (Smith, 1998).

AIM uses a combination of the model composition and model induction approaches, combining the best aspects of both, while eliminating their drawbacks. The expressive power of model composition techniques is harnessed both to develop the model boundary and to select the ontologies for representing "lumps" within this boundary. Model induction techniques are used to refine the model within this boundary: the ability to select the most appropriate combinations of simplifying assumptions to make in the model gives AIM the flexibility to generate parsimonious models. These three facets of modelling are addressed by a combination of novel knowledge architectures and novel algorithms using these architectures. AIM does not address the problem of selecting an appropriate representation and reasoning formalism: AIM is a component-centred modelling system. Neither does it address the lumping problem, instead relying on the description of the components in the system description to determine the different regions within both the system and the model.

The remainder of this paper describes AIM and how it addresses some of these problems in modelling. We shall also discuss the conditions under which AIM is able to generate models which have behaviours that quantitatively match those of the systems being modelled. We shall then give some results from AIM and draw some conclusions.

To show how AIM operates, we shall illustrate the model generation process by showing how AIM generates a model of a water-filled syringe (figure 1). The model will be used to simulate the response of all variables that are affected by a force applied by the finger.

Selecting Models of Components

The first problem we shall address concerns how AIM decides to represent each component in the model. The danger at this step is that the modeller will be influenced in its choice of representation by knowledge about how components are usually used. Such influences represent hidden modelling decisions and, if such hidden decisions could occur, the user of the model will have a lesser faith in the veracity of the model. In order to prevent such hidden decisions, AIM separates entirely knowledge about *components* from knowledge about *m-components* (or models of components). This separation ensures that the modelling knowledge base cannot contain any implicit links between the components that exist in the referent system and the manner in which these components are modelled.

M-components are similar to model fragments (Falkenhainer & Forbus, 1991) in that they are a partial representation of a component under a particular set of conditions. A wire could be modelled variously as an `electricalConductor` m-component, as a `mechanicalSpring` m-component, or as a simple `linearMass` m-component, depending on the situation in which the wire was found and the interactions to which it was subjected. Basic m-components always represent the ideal model of a component; non-ideal effects are accommodated through the inclusion of corrections in the model (see below).

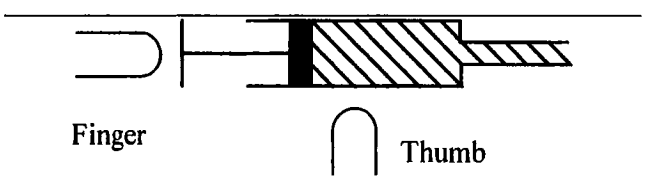


Figure 1: The Syringe

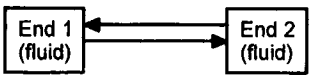
While the total separation between components and m-components prevents implicit decisions, AIM must be able to select the most appropriate m-component to use to represent a given component in a given situation. This facility is provided through a function that maps components onto m-components. This function takes account of the type of component being modelled, the interaction to which it is subjected, as well as the port at which this interaction occurs and the component's state (a switch will have different models depending on whether it is open or closed). The function's domain covers all physically possible combinations of component type, interaction domain, port, and state; this ensures that AIM can always determine the most appropriate m-component to use to represent a component under any given circumstances.

The behaviour of an m-component is described by a fragment of a bond graph (Rosenberg & Karnopp, 1983) contained in the m-component. When the model is generated, these bond graph fragments are merged to produce a bond graph representation of the model; the bond graph is used to produce the state equations that yield the system's behaviour. The various parameters that determine the m-component's behaviour, such as a block's moment of inertia, are calculated from physical parameters defined for the modelled component. M-components have other features, and these will be discussed below when appropriate.

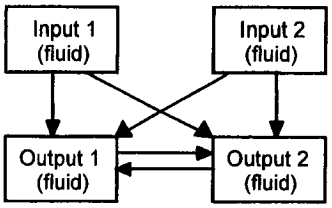
Finding the Model Boundary

With AIM capable of identifying the correct m-component to use to represent any component in the model, the next problem is how to select the components to model. This is the model boundary identification problem, and AIM solves it through an incremental expansion of the boundary from the information specified in the task description.

The key to the development of the model boundary lies in knowledge about how effects at one port of a component affect the other ports of that component. When an m-component is used in a model to represent a component, an effect (an interaction in a specified energy domain) at one port of a component will not necessarily cause similar effects at all other ports of the same component. To reflect this, each m-component has a set of intra-actions, which specify which of the m-component's other ports are affected by an effect at any one port. Intra-actions do not themselves indicate any causal direction within the m-component; they describe the possible causal orientations the m-component can support. Causal directions within an m-component are only defined when the m-component is placed in a model and the casual ordering of the whole



2*a*: Intra-actions in a `fluidPipe`



2*b*: Intra-actions in a sample `bathtub`

Figure 2: Intra-actions

model determined. The explicit nature of intra-actions in the modelling library ensures that the assumptions underlying the model boundary expansion are explicit in the library.

Intra-actions are normally symmetric, to reflect the symmetric nature of the relationships between effects in a component, *i.e.* the symmetric relationship between fluid flow rates at either end of a `fluidPipe` (figure 2*a*). However, the relationships between some effects are non-symmetric and the m-component's intra-actions reflect this. For example, the water flow rate from a tap into a bath affects the flow rate from the plughole, but the converse is not true (figure 2*b*). This information is used by AIM when it determines the model boundary.

The algorithm for the boundary identification process is shown in figure 3. It centres on the *boundary analysis queue*, which stores details of all the ports in the model (with their corresponding effects) that are currently on the model boundary. However, there can be no ports on a correctly-drawn model boundary as any ports on the boundary represent an under-specified effect: each port forms part of a junction, and the effects at a junction depend on all the ports at that junction. For example, the consideration of the current on one lead at an electrical junction requires the consideration of the electrical properties of all the other leads at that junction. This means that the boundary analysis continues until the boundary analysis queue becomes empty.

```
function expand_boundary (sd, initmodel, baq, cd)
% sd: system description
% baq: boundary analysis queue
% cd: causal direction specified in task
% initmodel: the initial model that is expanded
% m: the expanded model
% bap: boundary analysis port
% p: port
% cp, mp: additional ports
% mc : m-component

begin
  m = initmodel
  while baq ≠ {}
    bap = dequeue (baq)
    if bap ∉ m
      cp = system ports connected to bap
      enqueue (baq, cp)
      m = m + cp
    end
    mc = correct_m-component (bap, sd)
    if mc ∉ m ∨ ¬(∃p such that [p ∈ m ∧
        p ∈ intra-actions (mc, bap, ¬cd)]
      mp = intra-actions (mc, bap, cd)
      enqueue (baq, mp)
      m = m + mc
    end
  end
  return m
end
```

Figure 3: Algorithm for finding the model boundary

The boundary identification process starts by identifying the ports and effects referred to in the task definition and placing these in the boundary analysis queue. Using the syringe example (figure 1), the boundary analysis queue would initially contain the single port/effect combination `(finger, end, linearMechanical)`, representing AIM's interest in the mechanical properties of the finger.

Boundary expansion performed port by port, as shown in figure 3. When several ports are connected, an effect at one port requires AIM to consider the same effect at all the connected ports (providing the connection supports the inter-action). All the connected ports are added to the boundary analysis queue and the connection is added to the model. For instance, the finger touches the end of the plunger handle, so this latter port will be added to the boundary analysis queue. However, if a port already exists in the model (as part of a connection), the connection must already have been processed and AIM does not process it again.

AIM checks whether this port requires further processing by examining the m-component's intra-actions. If the current port was placed in the boundary analysis queue through the analysis of other ports of this m-component (*i.e.* if there exists in the model a port of the current m-component whose consideration would have caused this port to be included), AIM can move on to the next port in the boundary analysis queue. If not, details of the m-component, and the component to which it relates, are added to the model.

The m-component's intra-actions are then used to determine which of the component's other ports have effects that are significant in the model. The causal information in the task description indicates whether to use the intra-actions entering the port (to find the ports that affect the port in question) or the intra-actions leaving the port (to find the ports affected by this port). This only becomes significant where an m-component's intra-actions are not

Component	Modelled as
Finger	Source of force
Thumb	Not modelled
Ram	Rigid, massless bar
Plunger	Watertight, rigid, frictionless plunger
Cylinder	Rigid container of inviscid fluid
Nozzle	Rigid container of inviscid fluid
Atmosphere	Source of (zero) pressure

Table 1: M-components of the Syringe

symmetric. All ports mentioned in the relevant intra-actions are added to the boundary analysis queue. In the syringe, the finger is modelled as an exogenous application of force, and so has no intra-actions. The plunger handle, modelled as a rigid bar, has a symmetric intra-action that relates mechanical effects at the ends to each other: the finger pressing at one end of the handle prompts AIM to consider the effects of the other end of the handle on the plunger.

The components found to be inside the model boundary in the syringe example, together with their m-components, are shown in table 1. Note that all the components are represented by ideal models, and note that the atmosphere surrounding the syringe is explicitly included in the model boundary.

Selecting Simplifying Assumptions

Once the model boundary has been determined and the correct representation for all the components within that boundary has been found, attention can be focussed on which simplifying assumptions to include in the model. Basic m-components represent the *ideal* model of a component, *i.e.* a model made under all possible simplifying assumptions. However, not all of these assumptions are valid in all modelling circumstances: the viscosity of fluid in a pipe will be negligible if the fluid flows slowly and the model is only used to represent a small time scale. In other situations, the viscosity might be significant, and the basic `fluidPipe` m-component must be augmented to include the effects of fluid viscosity. This is achieved in AIM through the mechanism of corrections, which can be added to m-components to reflect the retraction of these simplifying assumptions. All corrections in AIM are examples of fitting approximations (Weld, 1992).

Corrections have a very similar structure to m-components. Corrections contain a bond graph fragment that describes how the correction affects the behaviour of the m-component to which they attach. The parameters in this bond graph fragment are defined by the physical parameters of the component to which it relates. Each correction has at least one port, the attachment port, that controls how the correction is added to the m-component (though note that corrections attach to an m-component's body, rather than one of its ports). Some corrections (*e.g.* heating in an electrical resistor) introduce additional ports and intra-actions to the m-component to which they attach. The inclusion of such corrections can expand the model boundary, but a detailed examination of such corrections is outside the scope of this paper.

As each correction represents the retraction of a simplifying assumption, a correction can only be applied once to a particular instantiated m-component. However, the same correction can be applied to an arbitrary number of different instantiations of the same m-component, and can even be applied to different types of m-component. The application of corrections to one m-component is independent of the application of that correction to any other m-component in the model. The *correction candidates* of a model are all those corrections that can validly be added to the model, *i.e.* are not already present in the model.

This architecture for the modelling knowledge used in AIM allows for the combination of the benefits of the model composition and model induction approaches. The component-centred approach to modelling, combined with the m-components' intra-actions, provides the sophistication required to accurately determine the model boundary in response to a specified task definition. The separation of the corrections from the m-components allows AIM the freedom to select only the simplifying assumptions that are appropriate in this model.

The initial model, found during the identification of the model boundary, is likely not to be sufficient to address the task specified. The model is made adequate by the addition of corrections. Corrections can be added for two reasons: to ensure coherence in the model, or to reduce the behavioural difference between the model and the referent system.

A model is said to be *incoherent* if the it does not yield a complete set of state equations; this normally reflects a physically impossible situation in the model. The initial model of the syringe, shown in table 1, would predict an infinite velocity for the ram, due to the lack of friction or other similar phenomena in the model. Such errors in the model are removed by the addition of corrections. The function **coherent**, described by Smith (1998), identifies the corrections that could potentially eliminate the error and selects the one that has the greatest effect on the model's behaviour. In the syringe example, this correction is the friction between the plunger and the cylinder.

The main reason for including corrections in a model is to reduce the behavioural difference between the model and the referent system. We take the view that models are always intended, at some level, to explain the behaviour of the referent system. This allows us to define a model as adequate for a task if the behaviour of the model is not significantly different from the behaviour of interest of the referent system.

If we assume that AIM's libraries contain all possible corrections (thus ignoring any closed world assumption), the behaviour of a model can be brought as close as desired to that of the referent system by the inclusion of corrections in the model. However, many of these corrections will

```
function find_parsimonious_model (sd, voi, cd, ε)
% sd: system description
% voi: variables of interest
% cd: causal direction specified in task
% ε: significance threshold value
% m, m', m_next: models
% ℜ(m): correction candidates of m
% r: correction

begin
  m = coherent (expand_boundary (sd, ∅, voi, cd))
  repeat
    ℜ(m) = all valid correction candidates for m
    J_max = 0
    for each r ∈ ℜ(m)
      m' = coherent (expand_boundary (sd, m + r,
            intra-actions(r, attach_port, cd), cd))
      J = ϑ(m, m')    % difference in behaviour
      if J > J_max,
        J_max = J
        m_next = m'
      end
    end
    if J_max ≪ λ    % J_max is significant
      m = m_next
    end
  until J_max ≪ λ ∨ ℜ(m) = {}
  if ℜ(m) = {}
    return nil    % cannot guarantee an adequate model
  else
    return m
  end
end
```

Figure 4: AIM's Algorithm

have little or no effect on the behaviour of the model. Normally, only a few of the possible corrections will have significant effects on the model's behaviour; the objective of modelling is to identify which corrections fall into this category. A model that contains all the significant corrections will be adequate for the specified task. An adequate model that contains no other corrections is parsimonious.

The problem is how to assess the effect of each correction candidate on the behavioural difference between the model and the referent system. Model induction systems (*e.g.* Addanki *et al.*, 1991) do this by generating the behaviour of the model and comparing it to a trace of the referent system's behaviour, provided as part of the problem specification. This requires that the referent system's behaviour be known.

Alternatively, AIM could produce a "most complex" model to produce a behaviour trace against which other models are compared. Unfortunately, this model would be formed under an arbitrary closed world assumption. Additionally, a model that contains all the corrections available under this closed world assumption might not be coherent. To make it coherent, AIM would have to arbitrarily eliminate corrections to ensure the model's coherence. Finally, increasing the knowledge in the modeller would increase the complexity of the initial model. If the additional complexity is not needed in the final model, including it only serves to increase the cost of developing the parsimonious model.

Instead of the above approaches, AIM takes the novel step of using the behaviour of an existing model as the base against which refinements are compared. The algorithm is shown in figure 4. When a model is generated, its behaviour is found. When each correction candidate is assessed, the model is revised to include this correction and the behaviours of the base model and the revised model are compared. This revision and comparison is repeated for all the model's correction candidates. If no correction causes a significant change in the model's behaviour, the model is deemed adequate and modelling stops. If the largest change in behaviour is significant, then the correction that caused that change is included in the model. This revised model becomes the current model and the process repeats. This approach relies on the effects of corrections on the model's behaviour being nearly independent, and on the insignificant correction assumption (see below).

AIM's search strategy through the space of possible models is similar to a steepest ascent hill climbing search. If the effects of corrections on the model's behaviour were truly independent, the order of inclusion of the significant corrections would be irrelevant. However, while corrections' effects are nearly independent, they are not truly independent. This has the result that if AIM is given a choice between two correction candidates, both of which have a significant effect on the model's behaviour but with one having a larger effect than the other, AIM should first include the correction candidate with the largest effect and reassess the model.

Whichever correction candidate is chosen, there is a chance that the remaining correction candidate will not have a significant effect on the modified model. If the corrections' effects are nearly independent, a correction candidate with a large effect will continue to have a large, significant effect, ensuring that this correction is included in a later refinement of the model. However, the small effect of the other correction candidate might become insignificant in a later model, meaning that this correction might not be included; in this case, this correction is not necessary to form an adequate model, and should not be included in the final, parsimonious model. This consideration leads to AIM's steepest ascent hill climbing search strategy. By including early the corrections with the largest effects on behaviour, AIM defers decisions on correction candidates with smaller effects until their impact becomes clearer.

In AIM's current implementation, behaviours are generated by simple numerical methods and compared using an extension of the integral-absolute error performance index (Palm, 1983). However, any method of generating and comparing behaviours will suffice for AIM's algorithm to work so long as behavioural differences can be found, ordered, and tested for significance.

Rather than compare the behaviours of all variables in the model, AIM selects a subset of variables to track, depending on the structure of the model and the causal direction in the task definition (if the model is to explain how the specified variables are affected, only these variables are tracked; if the effect of the specified variables on the rest of the model is to be found, all state variables and outputs are tracked). If there are n tracked variables in the base model, of which $v_i(t)$ is the ith tracked variable in the base model and $v_i'(t)$ is the corresponding variable in the modified model, the difference in behaviour over the period $[0, \tau]$ is given by:

$$\vartheta(m, m') = \max_{1 \le i \le n} \left(\frac{\int_0^\tau | v_i'(t) - v_i(t) | \, dt}{\int_0^\tau | v_i(t) | \, dt} \right)$$

where the performance index is normalised to give a measure of the relative difference in behaviour.

To show how AIM includes these corrections in the model, consider the simplest coherent model of the syringe (the model shown in table 1, with the addition of the plunger friction correction to ensure model coherence). If the cylinder is made of steel, the remaining correction with the largest effect is the inertia of the water in the nozzle: including this correction gives a performance index of 0.01218. Given a significance threshold value of 0.1, this is an insignificant effect. Therefore, the simplest model, excluding this correction, is deemed adequate. However, if the cylinder is made from soft rubber, the deformation of the cylinder is has the largest effect of all the corrections with a significant performance index of 0.2186. Following the algorithm in figure 4, this means that this correction is added to the model and refinement must continue.

The Insignificant Correction Assumption

AIM uses the insignificant correction assumption to determine when modelling is to stop. Modelling stops when the current model, m, is adequate. m is adequate iff the performance index between m and the referent system $\mathbf{S}$ is insignificant, *i.e.* $\vartheta(m, \mathbf{S}) \ll 1$ ($x \ll 1 \leftrightarrow x < \varepsilon$, where ε is the significance threshold value).

Turning attention to the mythical most complex model m_∞ (for which $\vartheta(m_\infty, \mathbf{S}) \equiv 0$), there are some corrections r in m_∞ for which $\vartheta(m_\infty - r, m_\infty) \ll 1$. Such corrections are termed *insignificant*, as they have no significant effect on the behaviour of the most complex model. $\mathbf{I}$ is the set of all such insignificant corrections:

$$\mathbf{I} = \{r \in \mathbf{R} : \vartheta(m_\infty - r, m_\infty) \ll 1\}$$

where $\mathbf{R}$ is the set of all corrections.

If we assume that effects of all corrections on the behaviour of the model are independent, it is true that:

$$\vartheta(m_\infty - \mathbf{I}, m_\infty) \ll 1$$

which means that $m_\infty - \mathbf{I}$ is an adequate model.

The insignificant correction assumption states that no insignificant correction has a significant effect on the behaviour of any model:

$$\forall r \in \mathbf{I}, \vartheta(m, m + r) \ll 1$$

Therefore, given $\mathbf{R}(m)$ is the set of valid correction candidates for m:

$$\begin{aligned} \forall r \in \mathbf{R}(m), \vartheta(m, m + r) \ll 1 &\rightarrow \mathbf{R}(m) \subseteq \mathbf{I} \\ &\leftrightarrow m \supseteq m_\infty - \mathbf{I} \\ &\rightarrow \vartheta(m, m_\infty) \ll 1 \\ &\leftrightarrow \vartheta(m, \mathbf{S}) \ll 1 \end{aligned}$$

This allows AIM to detect an adequate model by examining the effects of the remaining corrections on the behaviour of that model. If no correction candidate has a significant effect on the behaviour of the model, all the correction candidates must be insignificant; therefore, the model is adequate.

However, the assumption that all corrections have totally independent effects on the model's behaviour is not wholly true. In linear systems, corrections will generally have independent effects, but in order to allow for synergistic effects between corrections, the independence relation is weakened to:

$$\vartheta(m_\infty - \mathbf{I}, m_\infty) \ll 1/\lambda$$

which means that m is adequate when:

$$\forall r \in \mathbf{R}(m), \vartheta(m, m + r) \ll \lambda$$

The value to use for λ seems to be problem dependent, but an investigation of the combined effects of corrections (Smith, 1998) suggests that using $\lambda = \frac{1}{2}$ is reasonable.

Results

AIM's libraries contain approximately 40 components, 40 m-components, and 20 corrections. AIM has been used to generate models for several systems, including a cascaded tank system, a heat exchanger, and a simple tachometer, each containing about a dozen components. Each system was modelled under a variety of different conditions; the results are described by Smith (1998). However, only the syringe system (figure 3) is discussed here.

In order to accurately describe the effect of the force applied by the finger, the basic model of the syringe (table 1) is augmented with various corrections; these are shown in table 2. However, if the physical parameters in the system description are altered, different corrections will be appropriate in different circumstances. Table 2 shows how the necessary corrections change depending on whether the finger exerts a constant or rapidly varying force, and whether the syringe cylinder is made from steel or soft rub-

Cylinder	Force	Corrections
Steel	Constant	Plunger friction*
Steel	Varying	Plunger friction* Nozzle fluid inertia Plunger leakage
Rubber	Constant	Plunger friction* Cylinder deformation Nozzle fluid drag* Nozzle fluid inertia
Rubber	Varying	Plunger friction* Cylinder deformation Nozzle fluid drag* Nozzle fluid inertia

Table 2: Corrections added to the Syringe
* Correction added to ensure model coherence.

ber. The corrections are shown in the order in which they were added to the model (*i.e.* the most significant correction first). All models were built under the same significance threshold value (ε) of 0.1.

To determine whether these models are adequate, a very complex model was built manually and the behaviours of the generated models were compared to this complex model. The results are shown in the third column of table 3. In addition, the precursors of these models were also compared to the complex model and the results of these comparisons can be found in the fourth column of table 3.

As can be seen from these results, the models produced by AIM are adequate ($\vartheta(m, m_\infty) \ll 1$, *i.e.* $\vartheta(m, m_\infty) < \varepsilon$) while the precursor models are not ($\vartheta(m_-, m_\infty) \not< 0.1$). This means that AIM produced the simplest adequate, *i.e.* parsimonious, models.

Related Work

Nayak & Joskowicz (1996) describe a typical model composition system, whose modelling knowledge base shares many features with AIM. Model fragments are organised into hierarchies to allow the reuse of modelling knowledge, and articulation rules (similar to AIM's intra-actions) are used to infer what effects are induced in a component in response to a given effect. However, the modelling algorithm is entirely reliant on the structure of the modelling library to guide the modelling process. This is achieved at the cost of embedding modelling assumptions throughout the library. This is most obvious in the arbitrary nature of the preconditions and coherence constraints, relating to both the behavioural and structural context of a component, that control which model fragments can used in a model. As with all model composition systems, the limited number of model fragments within an assumption class limits the modeller's flexibility in selecting the most appropriate set of simplifying assumptions. Smith (1998a) shows how AIM manages to generate parsimonious models while avoiding most of these embedded assumptions.

Cylinder	Finger Force	$\vartheta(m, m_\infty)$	$\vartheta(m_-, m_\infty)$
Steel	Constant	0.089516	—
Steel	Varying	0.023991	0.451486
Rubber	Constant	0.035248	0.728074
Rubber	Varying	0.031973	1.01271

Table 3: Performance Indices for Syringe Models

DME (Iwasaki & Levy, 1994) uses relevance reasoning to guide modelling. Each model fragment contains a set of modelling assumptions under which it is valid. As the model boundary expands, these assumptions are asserted and retracted, which can prompt the modeller to revise earlier decisions. DME's performance depends critically on the correctness of these assumptions. Iwasaki & Levy do not guarantee this is the case, and instead offer the library coherence assumption. AIM's simpler knowledge base structure obviates the need to make any such coherence assumptions when developing the libraries, and AIM ensures the coherence of models as they are being built.

In contrast, MM (Amsterdam, 1992) is a typical model induction system that depends wholly on behavioural considerations. However, MM compares qualitative behaviours, which greatly reduces MM's ability to resolve qualitatively similar but quantitatively different behaviours. AIM's quantitative method of behavioural comparison allows it to produce models more appropriate to the specified task. In addition, while MM is able to correctly identify the different behaviourally significant regions within the system (the lumping problem), little attention is paid to the determination of the model boundary.

Williams & Raiman (1994) have produced Charicatures, a radically different modelling system based on the simplification of the equations that represent the most complex model. Their major contribution is their exploration of the concept of a model's domain of validity, which describes the situations in which the model can be used with confidence. This is used to indicate when model transitions are required. However, focusing solely on the equations is a very shallow approach as it ignores the physics that underlies the algebraic formulation. This restricts the application of Charicatures to situations that have already been modelled as algebraic systems.

Conclusions

We have described AIM, an automated modelling system that uses a novel architecture to provide both power and flexibility during the modelling process. AIM implements an algorithm that does not rely on external sources of information or fine structure in the modelling knowledge to guide and halt modelling. Instead, AIM compares the behaviours of successive models, including corrections that cause significant changes in the model's behaviour. We have shown that this approach to modelling usually generates parsimonious models.

However, this approach relies on the insignificant correction assumption and assumptions about the independence of corrections. Tests on other systems have shown that, occasionally, these assumptions are not sophisticated enough to always ensure parsimonious models. In addition, AIM's reliance on fitting approximations restricts the types of corrections that can be made.

References

Addanki, S., Cremonini, R., Penberthy, J. S. (1991), Graphs of Models, *Artificial Intelligence* **51**:145–177.

Amsterdam, J. (1992), Automated Modeling of Physical Systems, in Falkenhainer, B & Stein, J. L. (eds.), *Automated Modelling, DCS* **41**, American Society of Mechanical Engineers, pp. 21–30.

Falkenhainer, B., Forbus, K. (1991), Compositional Modeling: Finding the Right Model for the Job, *Artificial Intelligence* **51**:95–143.

Ironi, L., Stefanelli, M. (1994), A Framework for Building and Simulating Qualitative Models of Compartmental Systems, *Computer Methods and Programs in Biomedicine*, **42**:233–254

Iwasaki, Y., Levy, A. Y. (1994), Automated Model Selection for Simulation, *Proceedings AAAI-94*, pp. 1183–1190.

Nayak, P. P., Joskowicz, L. (1996), Efficient Compositional Modeling for Generating Causal Explanations, *Artificial Intelligence* **83**:193–227.

Palm, W. J. (1983), Modeling, Analysis, and Control of Dynamic Systems, New York: J. Wiley & Sons.

Schut, C., Bredeweg, B. (1996), An Overview of Approaches to Qualitative Model Construction, *Knowledge Engineering Review* **11**(1):1–25.

Smith, N. (1998), Reducing the Need for Assumptions in the Automated Modelling of Physical Systems, PhD thesis, School of Computing, De Montfort University.

Smith, N (1998a), A New Architecture for Automated Modelling, *Proceedings AAAI-98*, to appear.

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

Williams, B. C., Raiman, O. (1994), Decompositional Modeling Through Charicatural Reasoning, *Proceedings AAAI-94*, pp. 1199–1204.