

TOWARDS A COMPUTER AIDED DESIGN OF REACTIVE SYSTEMS

Marc Frappier and Richard St-Denis
Département de mathématiques et d'informatique
Université de Sherbrooke, CANADA

Plan

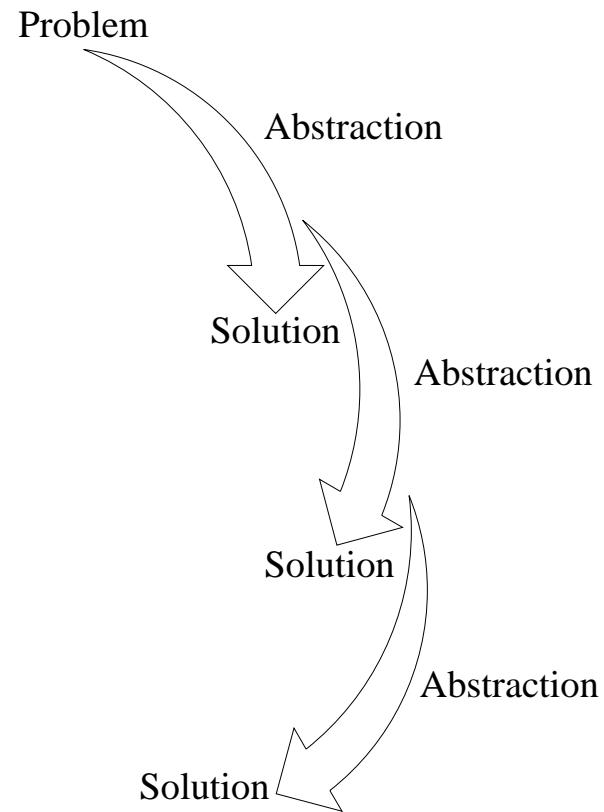
1. Motivation
2. A control problem
3. Overview of the method
4. Attributed controller
5. Illustration by means of an example
6. Correctness and soundness
7. Computer aided design tool

Analogy with the Domain of Compiler Construction

	Parser Construction	Controller Construction
Theory	<i>Theory of Parsing</i>	<i>Supervisory Control Theory</i>
Formal Notations	context-free grammar (G)	timed transition graph (G) temporal logic formula (f)
Synthesis Procedure	$G \rightarrow$ characteristic machine	$G, f \rightarrow (S, \varphi)$
Tools	<i>yacc</i>	<i>MELODIES</i>

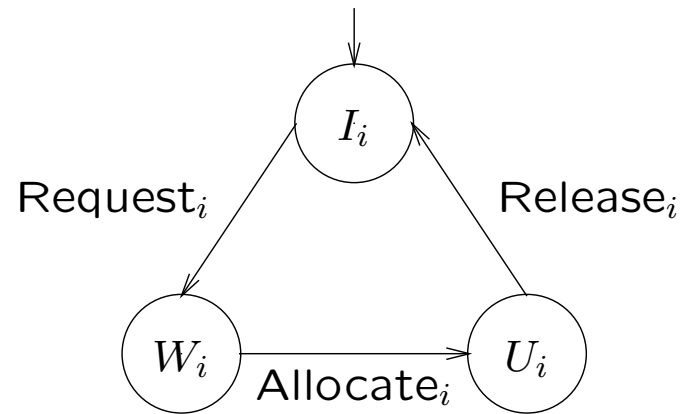
MELODIES : Modeling Environment for LOfical DIcrete Event Systems

Design Exploration or Rapid Software Prototyping (by means of a synthesis approach)



State Explosion Problem

l users (processes) sharing a single resource (CPU)



Constraints: mutual exclusion and fairness (FIFO)

$$3^l - \sum_{k=0}^{l-2} \binom{l}{k} 2^k + \sum_{k=2}^l \binom{l}{k} (l - k + 1)(k! - 1)$$

l	3^l	# of states
2	9	9
3	27	31
4	81	129
5	243	651
6	729	3 913
7	2 187	27 399
8	6 561	219 201
9	19 683	1 972 819
10	59 049	19 728 201
11	177 147	217 010 223

Techniques to Circumvent the State Explosion Problem

- Symmetries or quotient structures (Eyzell *et al*)
Reduction by a factor of n when the system consists of n similar components
- *On-line* calculation of a control policy (Lafortune *et al*)
Linear computational complexity, but a weaker level of reliability

Formal Description of a Control Problem

Given the behavior of a process (\mathcal{M}) and properties (f), construct an *attributed* controller (C^A) such that the behavior of the closed-loop system ($\mathcal{M} \parallel C^A$) satisfies the properties:

$$(\mathcal{M} \parallel C^A) \models f$$

Generally, \mathcal{M} can be specified in a parametric form :

- $\mathbf{M} := \langle M_1, \dots, M_m \rangle$

Objects of the attributed controller are also characterized by internal dimensions :

- $\mathbf{N} := \langle N_1, \dots, N_n \rangle$

Candidate Approaches to Solve this Problem

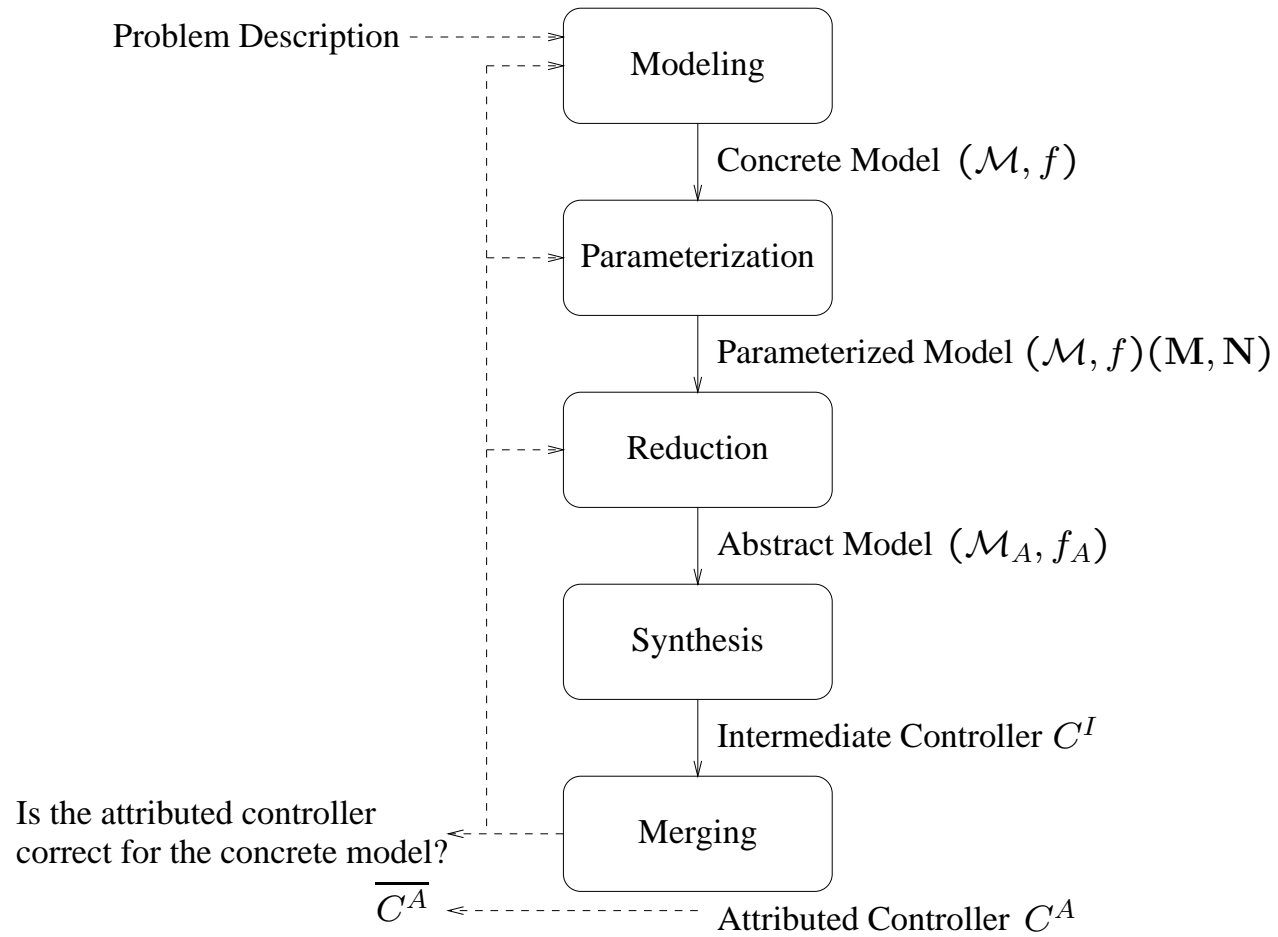
- Synthesize a controller with respect to specific vectors \mathbf{M} and \mathbf{N}
- Construct a correct controller for arbitrary values of \mathbf{M} and \mathbf{N}

The Selected Approach

model for the concrete problem \longrightarrow model for an abstract problem

solution for the abstract problem \longrightarrow solution for the concrete problem

Overview of the Method



Attributed Controller

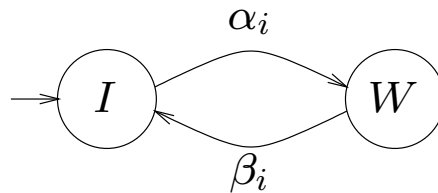
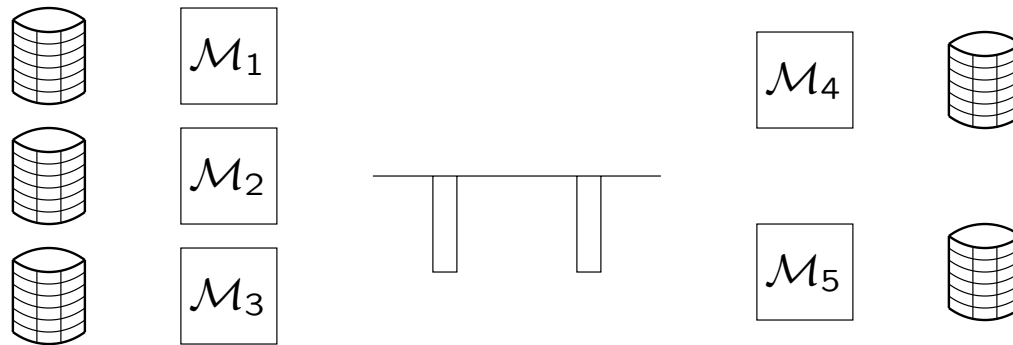
- Typed objects
Base its decisions not only on the current state of its attributed transition structure but also on current internal states of its objects
- Attributed transition structure
Synchronize the behavior of the attributed controller to process behavior
- Conditional control policy
Restrain the behavior of the process by disabling some controllable actions

Attributed Controller

$$(Q_a, Q_o, A, \delta_a, \delta_o, q0_a, q0_o, \varphi)$$

- Active components
- Passive components (objects)
- $\varphi : Q_a \times Q_o \rightarrow \mathbb{P}(A)$

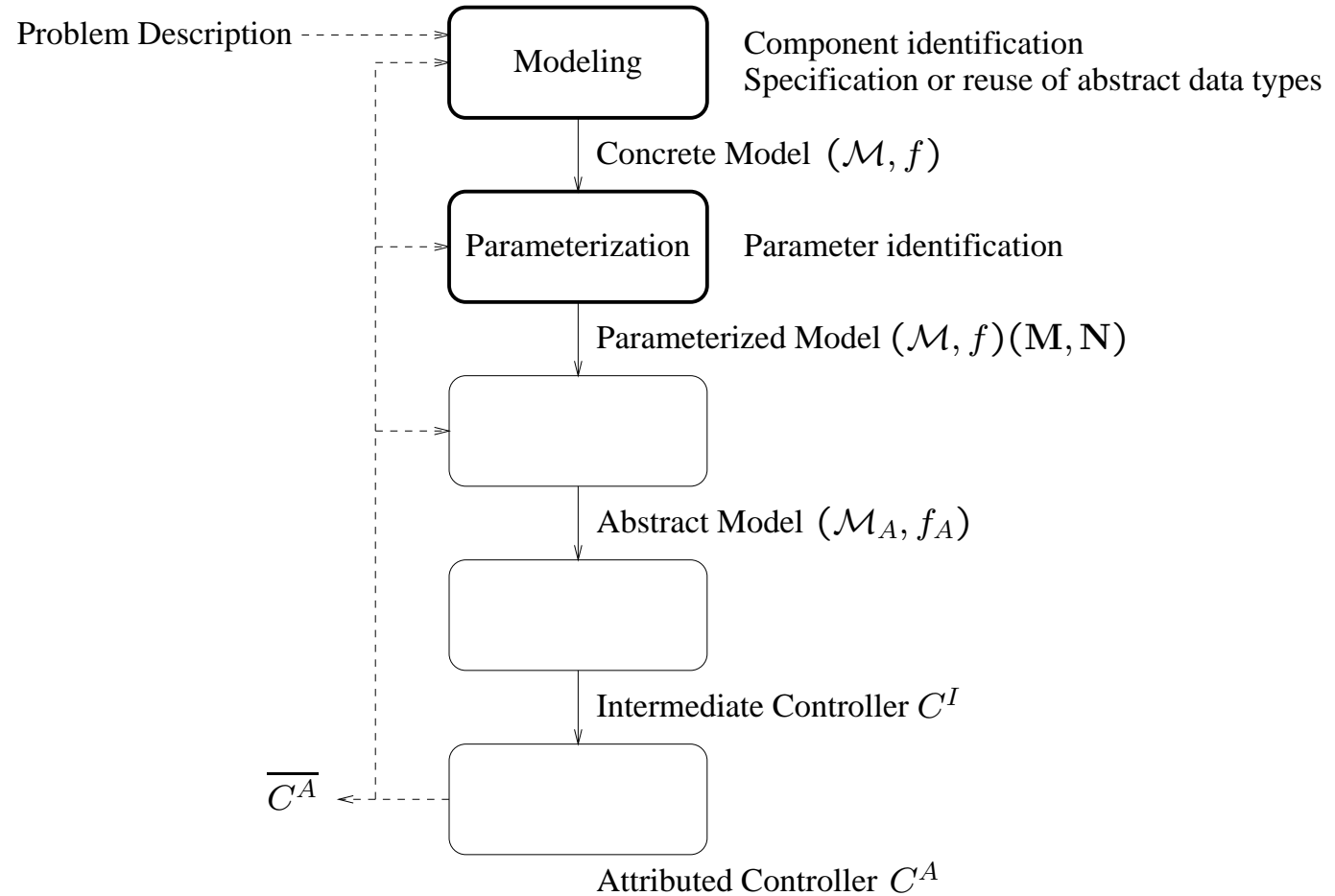
Running Example



Actions α_i are controllable and actions β_i are uncontrollable.

The table capacity is 10.

The First Two Steps of the Method



The First Two Steps of the Method

- Component identification
 - Active components: $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5$
 - Passive components: the table is modeled as a buffer (b)
- Specification or reuse of abstract data types and association of operations to actions

$$\rho(\beta_i) = \{\langle b, Add(b) \rangle\}$$

if machine i is at the left of the table

$$\rho(\alpha_i) = \{\langle b, Remove(b) \rangle\}$$

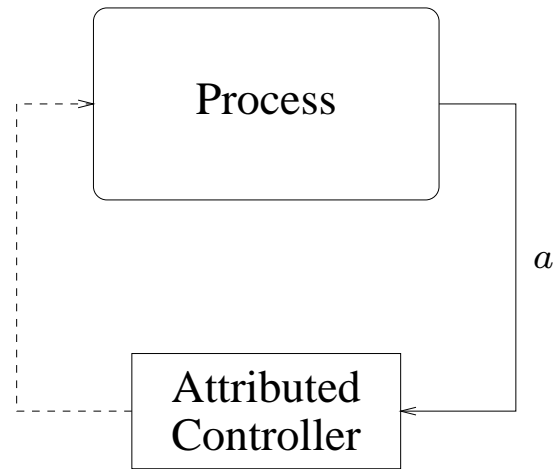
if machine i is at the right of the table

- Parameters identification

$$\mathbf{M} = \langle 3, 2 \rangle \text{ and } \mathbf{N} = \langle 10 \rangle$$

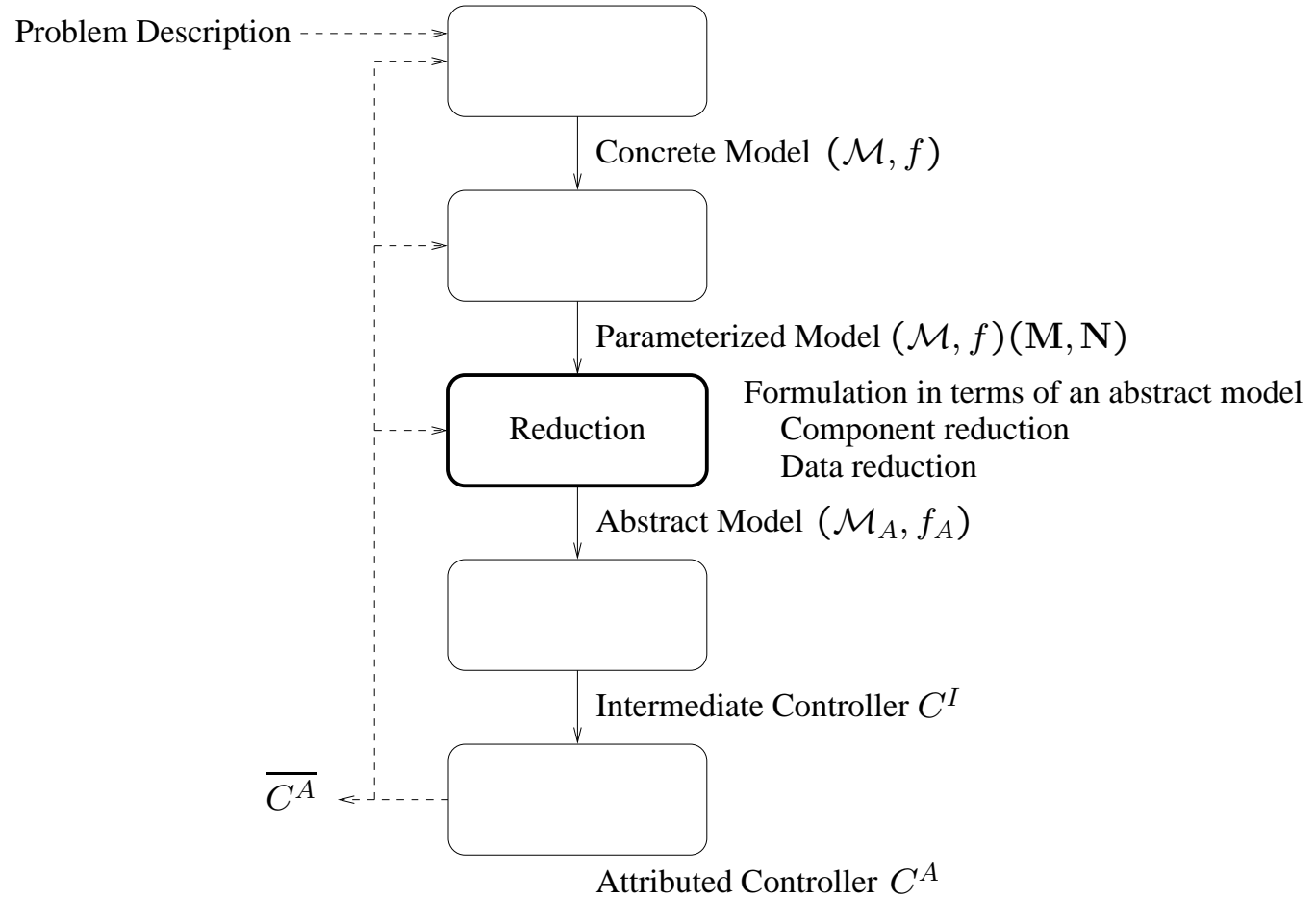
An Algebraic Specification of an Abstract Data Type (a buffer)

```
Buffer(capacity ∈ nat) :=
import : bool, nat
hidden sorts : buffer
operations :
  New : → buffer
  Add : buffer → buffer
  Remove : buffer → buffer
  Size : buffer → nat
  Is_Empty : buffer → bool
  Is_Full : buffer → bool
equations : b ∈ buffer, n ∈ nat
  Is_Empty(New) = TRUE
  Is_Empty(Add(b)) = FALSE
  Is_Full(New) = FALSE
  Is_Full(Addn(New)) = FALSE (n ≠ capacity)
  Is_Full(Addcapacity(New)) = TRUE
  Size(New) = 0
  Size(Addn(New)) = n
  Remove(New) = ERROR
  Remove(Add(b)) = b
  Add(Addcapacity(New)) = ERROR
```



If $\rho(a) = \{\langle b, New \rangle\}$, then the new state of b is $Add(New)$

The Third Step of the Method



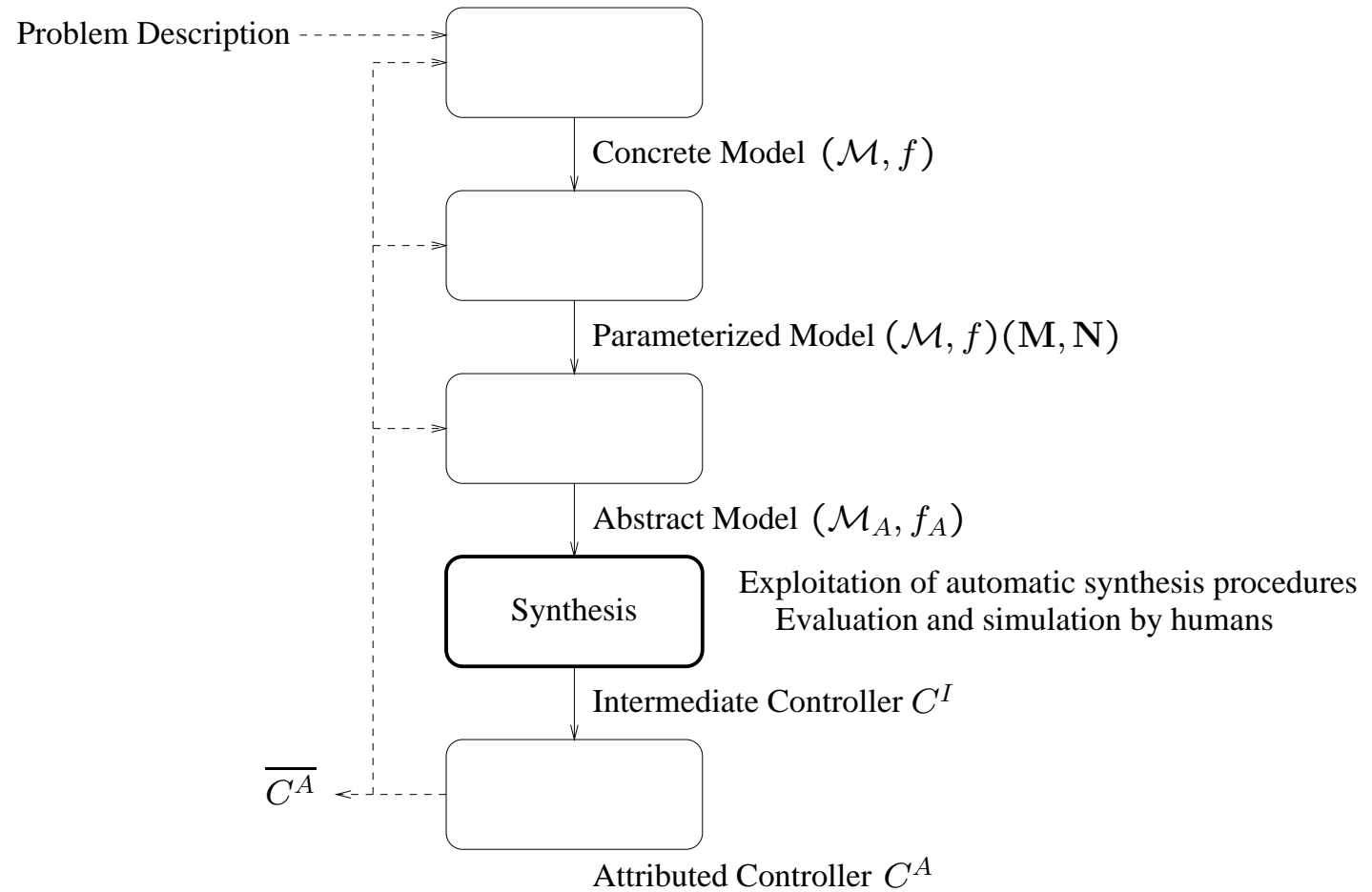
Formulation in Terms of an Abstract Model

$$\mathbf{M} = \langle p', p'' \rangle \text{ and } \mathbf{N} = \langle c \rangle$$

We must solve the problem for small values of p', p'' , and c , say $\mathbf{M} = \langle 2, 1 \rangle$ and $\mathbf{N} = \langle 3 \rangle$, and for the constraints modeled by the following temporal formula :

$$\begin{aligned} & \square(\\ & \quad (W_1 \wedge Is_Full(b) \rightarrow \bigcirc \neg I_1) \wedge \\ & \quad (W_2 \wedge Is_Full(b) \rightarrow \bigcirc \neg I_2) \wedge \\ & \quad (I_3 \wedge Is_Empty(b) \rightarrow \bigcirc \neg W_3) \\ &) \end{aligned}$$

The Fourth Step of the Method



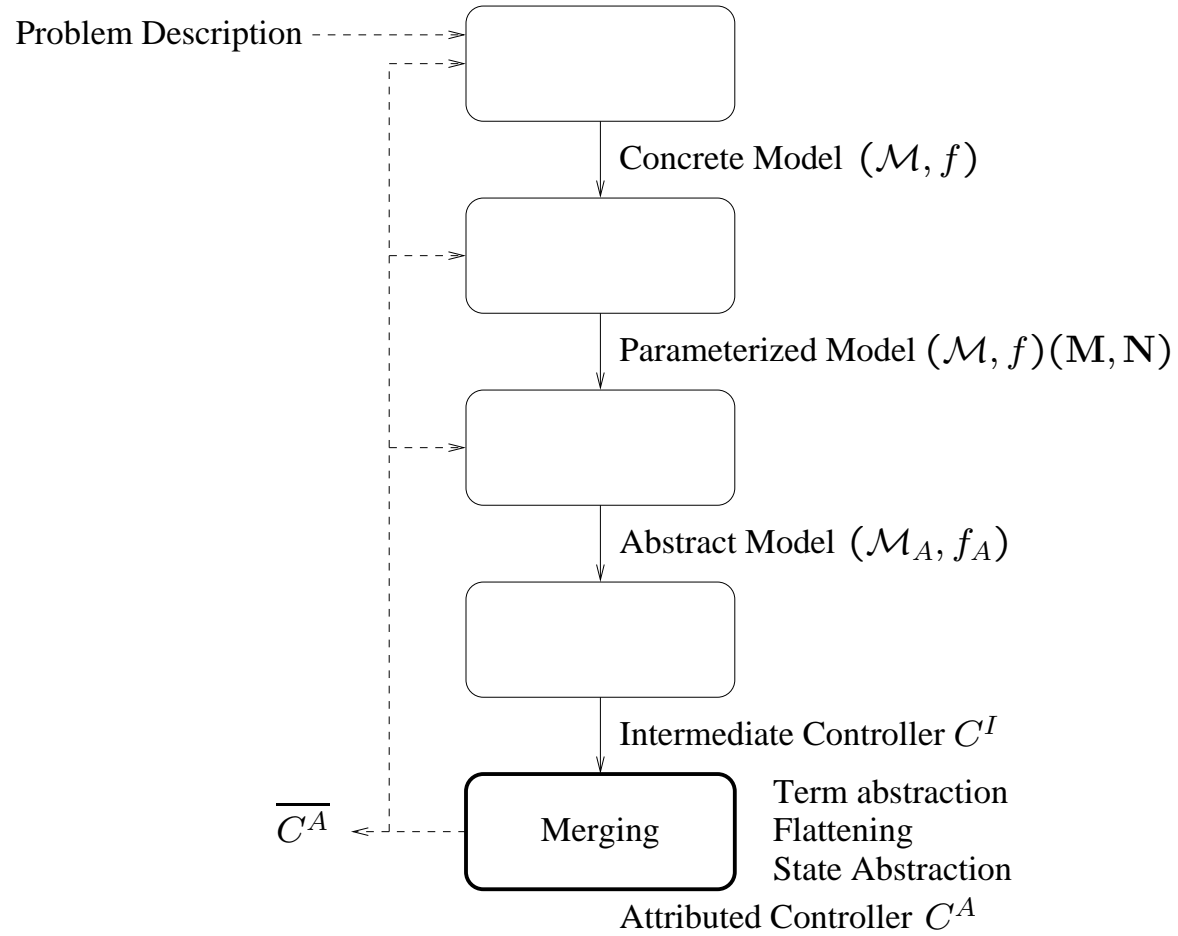
Synthesis of an Intermediate Controller

(Adaption of the Barbeau, Kabanza, and St-Denis's synthesis method)

Intermediate controller (control policy) :

$$\begin{array}{ll} (I I I New) : \{\alpha_3\} & (W I I New) : \{\alpha_3\} \\ (I W I New) : \{\alpha_3\} & (W W I New) : \{\alpha_3\} \\ (W I I Add(Add(New))) & : \{\alpha_2\} \\ (I W I Add(Add(New))) & : \{\alpha_1\} \\ (I I I Add(Add(Add(New)))) & : \{\alpha_1 \alpha_2\} \\ (W I W Add(Add(New))) & : \{\alpha_2\} \\ (I W W Add(Add(New))) & : \{\alpha_1\} \\ (I I W Add(Add(Add(New)))) & : \{\alpha_1 \alpha_2\} \end{array}$$

The Fifth Step of the Method



Term Abstraction

- Rewriting non-Boolean terms into Boolean terms
 - Semi-automatic operation
 - Richness of the set of equations
- Transfer of Boolean terms in control actions

$$(I, W, I, New) : \{\alpha_3\} \rightsquigarrow (I W I Is_Empty(b)) : \{\alpha_3\}$$

$$(I W I Is_Empty(b)) : \{\alpha_3\} \rightsquigarrow (I W I) : \{(\alpha_3, Is_Empty(b))\}$$

$$(I I I Add(Add(Add(New)))) : \{\alpha_1 \alpha_2\} \rightsquigarrow (I I I Is_Full(b)) : \{\alpha_1 \alpha_2\}$$

$$(I W W Add(Add(New))) : \{\alpha_1\} \rightsquigarrow (I W W (Size(b) = capacity - 1)) : \{\alpha_1\}$$

Flattening

- Merging strategies
- A particular case (if action a is physically impossible in the process, then add a pair of the form $(a, TRUE)$)
- General conditions versus specific conditions

$(I I I New)$: $\{(\alpha_3, Is_Empty(b))\}$
 $(I I I Add(New))$: $\{\}$
 $(I I I Add(Add(New)))$: $\{\}$
 $(I I I Add(Add(Add(New))))$: $\{(\alpha_1, Is_Full(b)), (\alpha_2, Is_Full(b))\}$

Conditional Control Policy after Merging

- (*I I I*) : $\{(\alpha_1, Is_Full(b)), (\alpha_2, Is_Full(b)), (\alpha_3, Is_Empty(b))\}$
- (*W I I*) : $\{(\alpha_2, Size(b) = capacity - 1), (\alpha_3, Is_Empty(b))\}$
- (*I W I*) : $\{(\alpha_1, Size(b) = capacity - 1), (\alpha_3, Is_Empty(b))\}$
- (*W W I*) : $\{(\alpha_3, Is_Empty(b))\}$
- (*W W W*) : $\{\}$
- (*W I W*) : $\{(\alpha_2, Size(b) = capacity - 1)\}$
- (*I W W*) : $\{(\alpha_1, Size(b) = capacity - 1)\}$
- (*I I W*) : $\{(\alpha_1, Is_Full(b)), (\alpha_2, Is_Full(b))\}$

Second Cycle

Add a new passive component *cnt* (a counter) memorizing the number of working producer machines

$$q : \{(\alpha_1, \text{Size}(b) = \text{capacity} - \text{cnt}), \\ (\alpha_2, \text{Size}(b) = \text{capacity} - \text{cnt}), \\ (\alpha_3, \text{Is_Empty}(b))\}$$

Correctness

A transformation for merging maintains *correctness* if C^A has the same behavior as C^I for the selected values of the parameters.

- Transformations preserving correctness
 - Term abstraction
 - Flattening
 - Strengthen local policy
 - State abstraction

Soundness

The abstract model (\mathcal{M}_A, f_A) is a *sound* abstraction for the concrete model (\mathcal{M}, f) if $(\mathcal{M}_A \parallel C^A) \models f_A$ implies $(\mathcal{M} \parallel \overline{C^A}) \models f$.

Summary

Transformation	Automated Support	Proving Soundness	Proving Correctness
Data Reduction	N	?	
Component Reduction	N	?	
Term Abstraction	Y		Y
Flattening	Y		Y
Strengthen Local Policy	Y		Y
State Abstraction	Y		Y

Computer Aided Design Tool

- Based on our current environment *MELODIES*
- Based on the proposed strategy for problem solving
- Supported by human assistance
- Guided by a cognitive model
- Enriched by a composite proof theory
- Reinforced by locally testable soundness criteria