



Composition in Modeling Macromolecular Regulatory Networks

Ranjit Randhawa

September 9th 2007

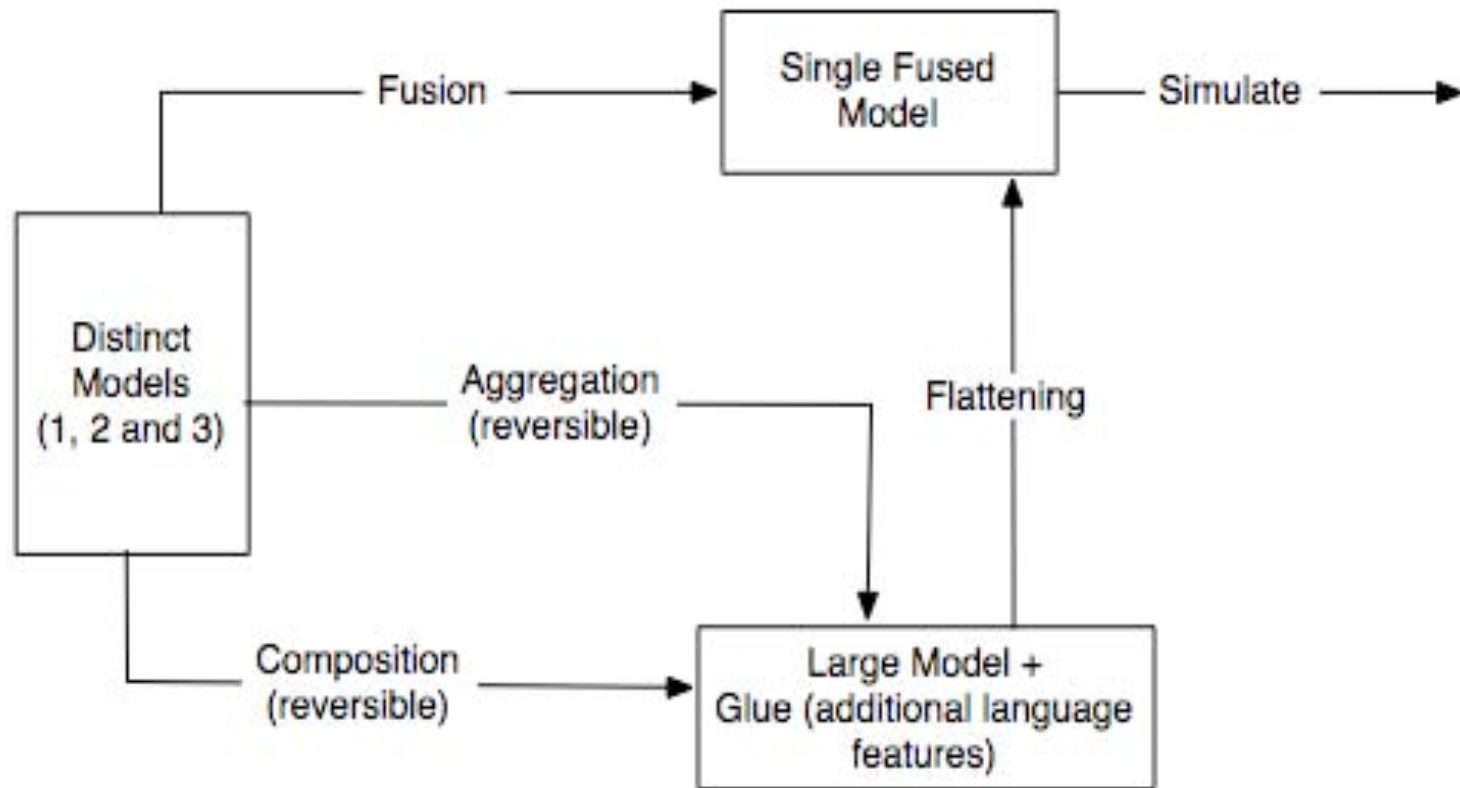
Motivation/Problem

- # Models are reaching the limits of human comprehension
- # Making the model suitable for stochastic simulation increases the number of reactions by a factor of 3-5
- # Models of the Mammalian cell cycle will require 100-1000 (more for stochastic simulation)

Our Approach

- # We recognize four distinct activities related to model decomposition
 - Fusion: Take existing models and merge them
 - Composition: Build up from existing models, no information hiding
 - Aggregation: Build up from building blocks, controlled interfaces
 - Flattening: Merge the building blocks back into a “flat” (non-composed) model

Relationships



Fusion

- # Given two or more existing models, we wish to create a new model that combines the information
- # Remains standard SBML
- # We provide a tool to support users combining models
 - Implemented in “wizard” style

Composition

- # Connects submodels together to form a hierarchy of models
- # Submodels are each valid SBML models
- # Add language features to SBML to support composition
 - Describe hierarchy
 - Describe interactions, links, replacements
- # No information hiding within models

Composition and the Fusion Wizard

- # There are significant similarities between fusion and composition
 - # Fusion defines a series of steps taken to merge models
 - Series of steps captured by the fusion tool can be viewed as an “audit trail” used in generating the mapping tables
 - # Precisely this same information can be used to describe the set of instructions needed to connect/link the submodels for composition
-

Is Composition the Right Model?

- # Composition allows us to take existing models and use them as components to build larger models
- # No information hiding
- # Submodels might fit together more or less well
 - Links let us replace things in one model with things in another
- # Good for legacy models (?)
- # We might do better to build models from components designed to work as components, with proper information hiding

Aggregation

- # In aggregation, models are built up from components designed for aggregation.
- # Each component could be, for example, a collection of reactions
- # This collection exposes certain variables for input/output via ports/interfaces
- # Hopefully this is a natural concept for modelers
- # Not intended as a solution for reusing legacy models

Flattening

- # The purpose of flattening is to generate a standard SBML file from our modified file, for the purpose of running simulations, etc.
- # An automated form of fusion
- # The composition/aggregation language features provide exactly what the user would provide in fusion, so automation is possible

SBML Language Features

<listOfSubmodels>

- Contains 1 or more <models> structures (which can be flat or composed models)

<listOfInstances>

- Allows for multiple instances of the same model

<listOfLinks>

- Types (merge/replacement)
 - Enables grouping (e.g. N to N links)
 - Ignore/Exclude ability
 - A link with an empty <from> field
-

SBML Language Features

<listOfPorts>

- Input/output ports
 - Currently species are outputs and parameters are inputs
 - Boolean *input* attribute
 - Ports connect to:
 - Parameters
 - Species
 - Other ports (used for multiple level aggregation)
 - Modifiers (?)
 - Modifiers will need to be “special” input ports (hence the use of the input field/attribute)
 - Or can one just replace modifiers with parameters?
-

Instance and Links Syntax

```
<listOfInstances>
  <instance id="instanceOfSubmodel1_1"
    xlink:type="simple"
    xlink:href="#xpointer(/sbml/model/listOfSubmodels/model
      [@id=%22Submodel1_1%22])" />
</listOfInstances>
```

```
<listOfLinks merge=false>
  <link>
    <from object="cell_4" />
    <to object="instanceOfSubmodel1_1">
      <subobject object="cell_1" />
    </to>
  </link>
</listOfLinks>
```

Ports Syntax

Syntax:

- `<listOfPorts>`
 - `<port id="p_sA_1" name="sA" input=true>`
 - `<target object="sA_1" />`
 - `</port>`
- `</listOfPorts>`

ObjectRef.SubobjectRef for port Target field (as well as Link To/From fields)

Status of Tools

- # Fusion Wizard
 - Implemented
 - # Composition Wizard
 - Implemented and combined with Fusion tool to produce the Fusion/Composition Wizard (needs a new name)...
 - # Aggregation Connector
 - Implemented prototype
 - Currently working to be able to traverse up/down aggregation tree.
 - # Flattening Algorithm
 - Implemented for Composition
 - Need to extend it for Aggregation as well
-

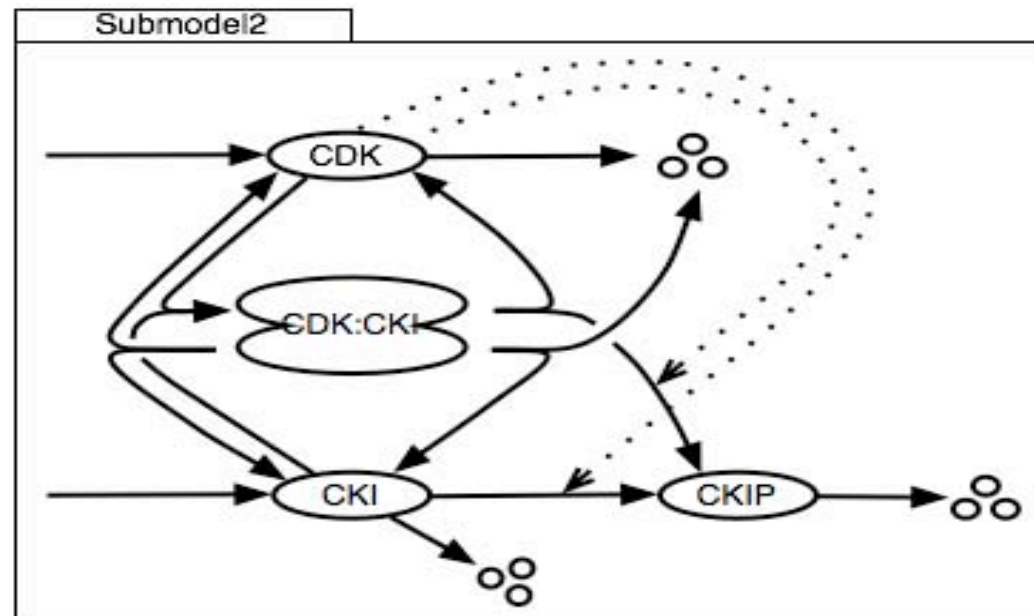
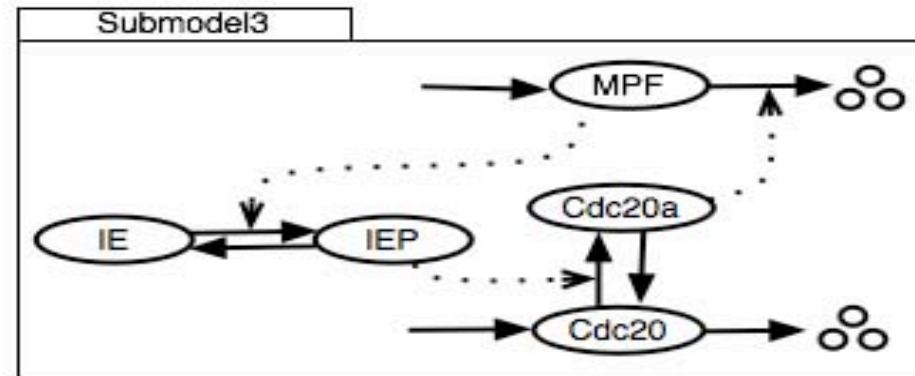
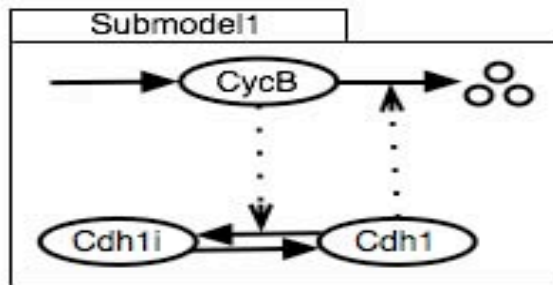
Evaluation

- # How does one evaluate the approaches?
 - # What are the metrics?
 - # How will we know if we are successful?
 - # How to recognize a solution?
-

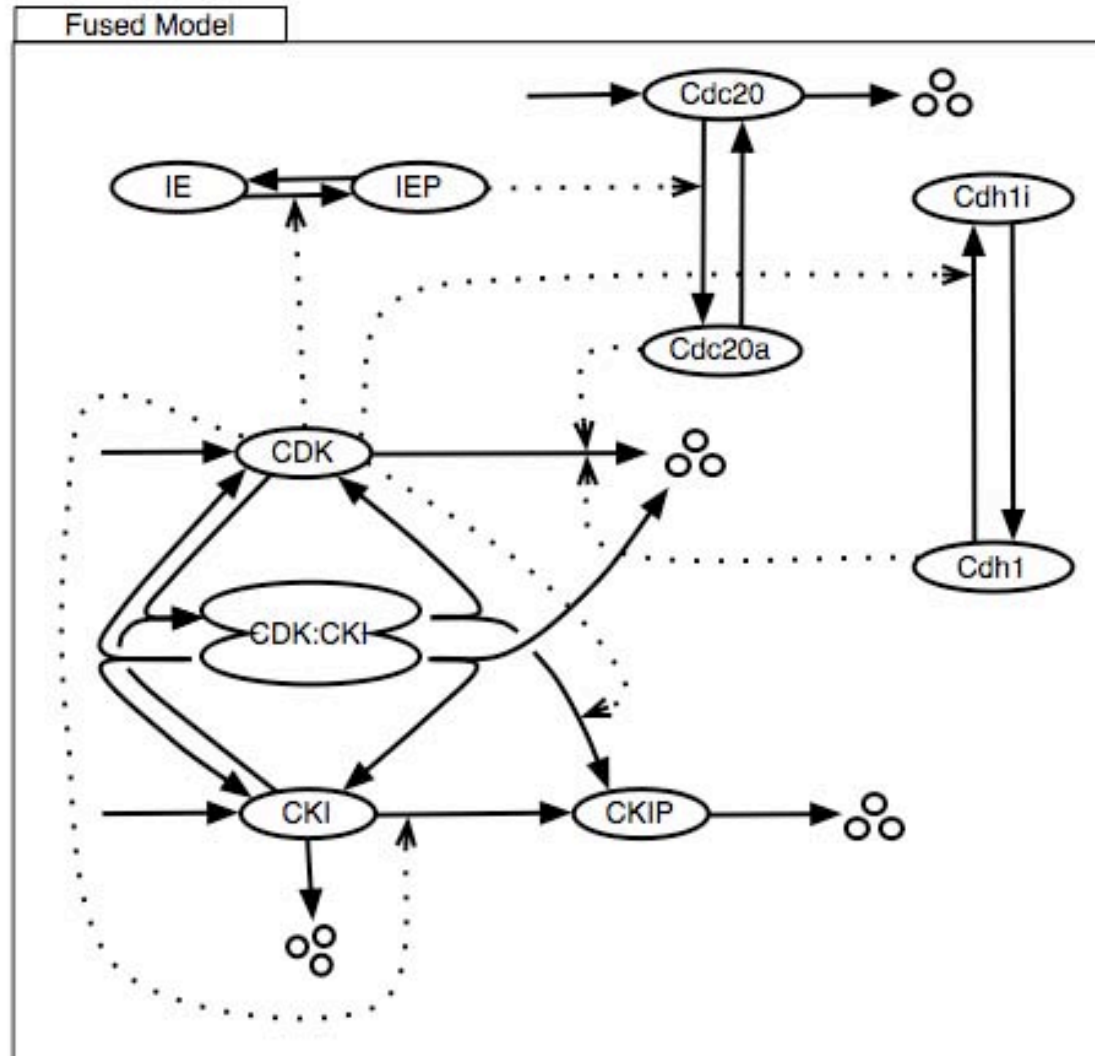
Creating Larger Models

- # 3 sample models from Tyson & Novak 2001
 - # Approaches:
 - Submodels fused together
 - Additional components added to fused model (in red)
 - Submodels composed together
 - Additional components added to composed model (in red)
 - Modified submodels aggregated together
-

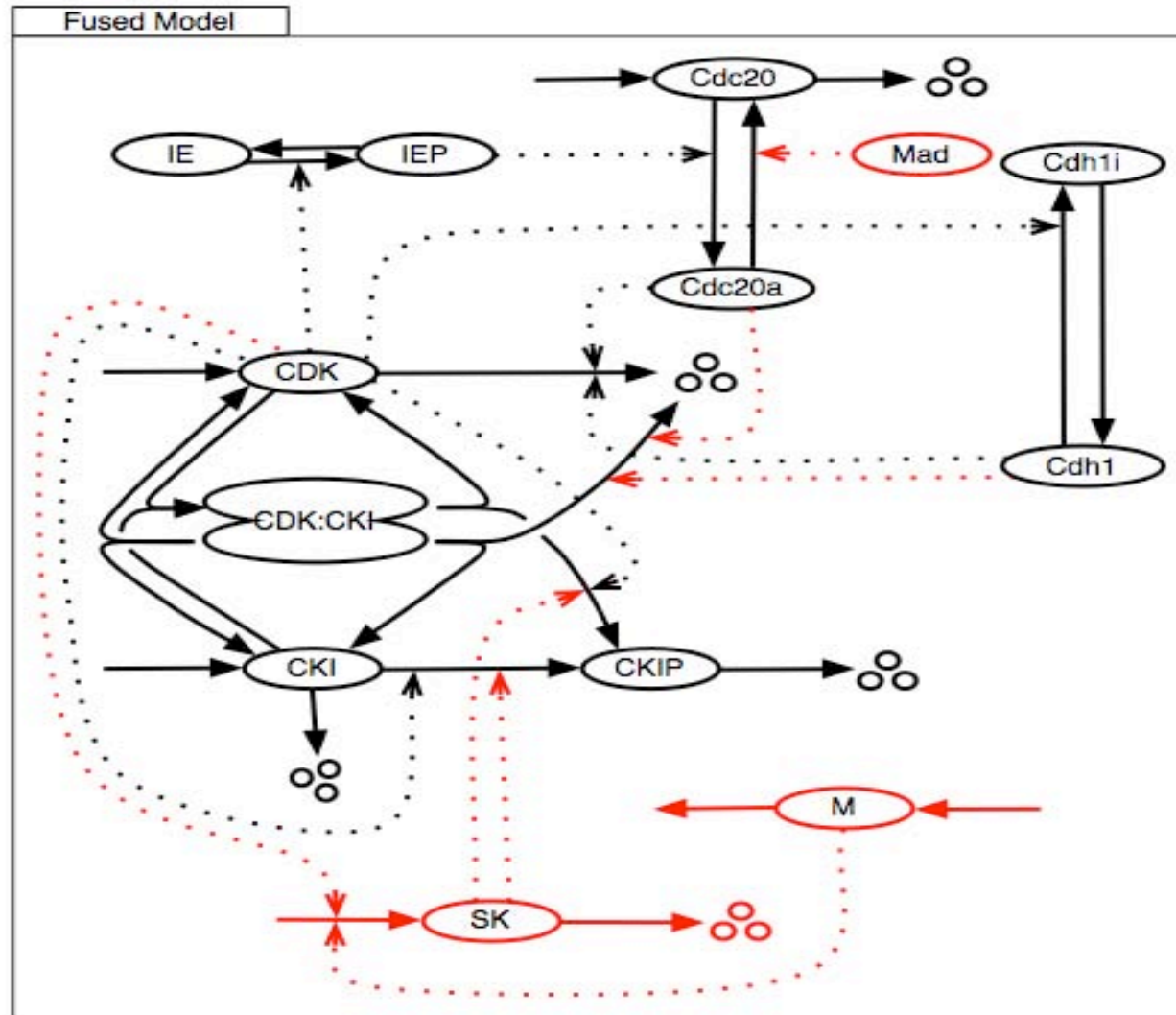
Sample models



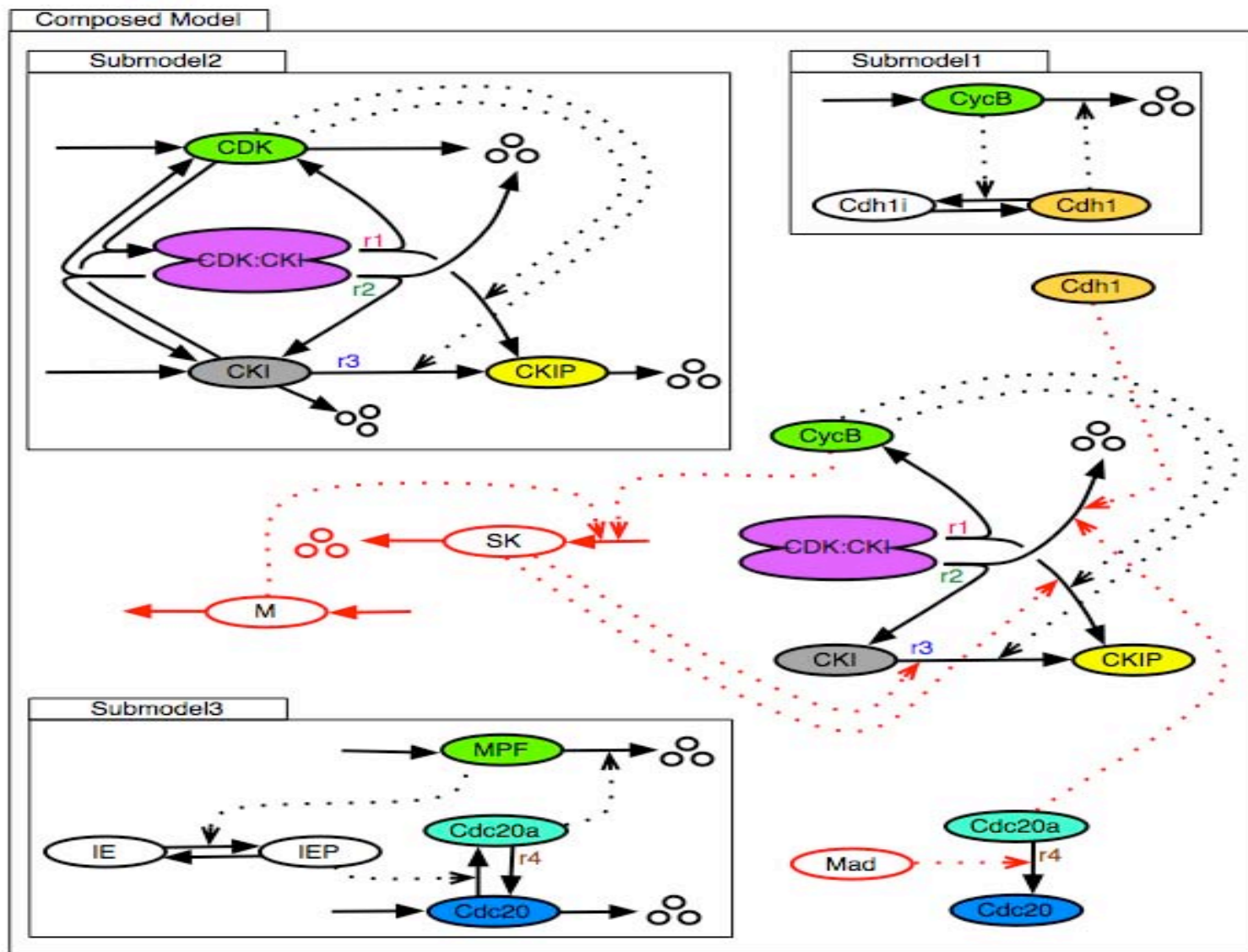
Initial Fused Model



Fused Model with New Components



Implemented Composed Model



Screen shots Composition Wizard

Final Species Mapping Table

Species Mapping Table

	ComposedModel	ComposedModel.Submodel1	ComposedModel.Submodel2	ComposedModel.Submodel3	ComposedModel Notes
1	CycB	CycB	CDK	MPP	
2	CKI		CKI		
3	CKIP		CKIP		
4	CDK:CKI		CDK:CKI		
5	Cdc20			Cdc20	
6	Cdc20a			Cdc20a	
7	SK				
8	M				
9	Mad				
10	Cdh1	Cdh1			
11		Cdh1i			
12				IE	
13				IEP	

Description of Attributes:

1. Click on a row number above to view the attributes
2. Click on a cell below to scroll through the attribute list

Species in grey are NOT INCLUDED in the fused or composed model.

Back

Next

Cancel

Screen shots Composition Wizard

Final Reaction Mapping Table

Reaction Mapping Table

	ComposedModel	ComposedModel.Submodel1	ComposedModel.Submodel2	ComposedModel.Submodel3	ComposedModel Notes
1	CKI -> CKIP		CKI -> CKIP		
2	CDK:CKI -> CKI		CDK:CKI -> CKI		
3	CDK:CKI -> CycB + CKIP		CDK:CKI -> CDK + CKIP		
4	Cdc20a -> Cdc20			Cdc20a -> Cdc20	
5	-> SK				
6	SK ->				
7	-> M				
8	M ->				
9	-> CycB	-> CycB	-> CDK	-> MPF	
10		CycB ->	CDK ->	MPF ->	
11		Cdh1i -> Cdh1			
12		Cdh1 -> Cdh1i			
13			-> CKI		
14			CKI ->		
15			CKIP ->		
16			CDK + CKI -> CDK:CKI		
17			CDK:CKI -> CDK + CKI		
18				-> Cdc20	
19				Cdc20 ->	
20				IE -> IEP	
21				IEP -> IE	
22				Cdc20 -> Cdc20a	

Description of Attributes:

1. Click on a row number above to view the attributes
2. Click on a cell below to scroll through the attribute list

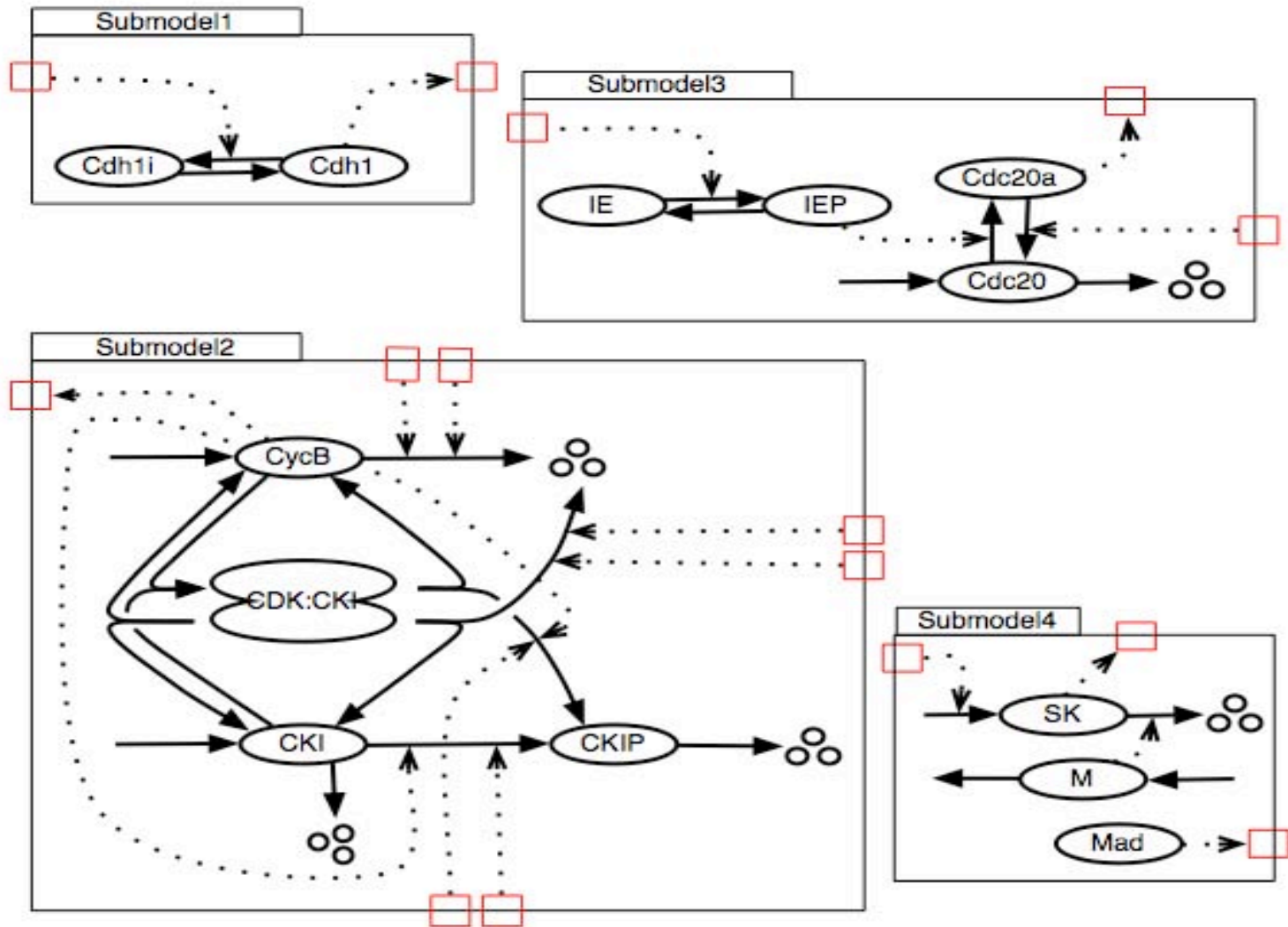
Reactions in grey are NOT INCLUDED in the fused or composed model.

Back

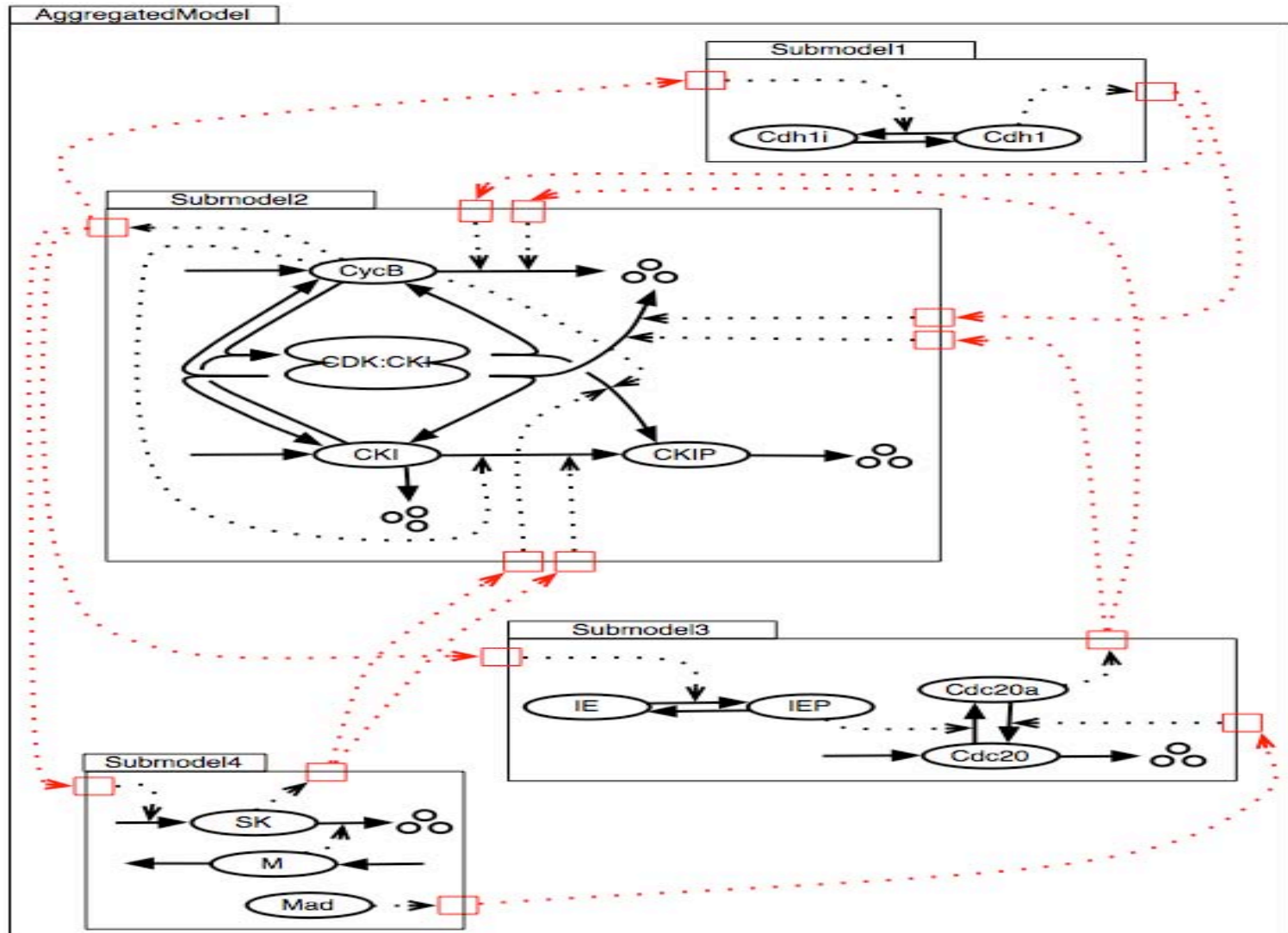
Next

Cancel

Aggregated Submodels



Aggregated Model



Screen shots Aggregation Connector

JigCell Aggregation Connector

Open Module(s) Create New Module Save Module Save Layout Flatten Module

Module: **Submodel1** File Edit

Reactions Functions Rules Compartments Species Parameters Events Units Conservation Relations Equations

Name	Reaction	Type	Equation	Fast	Notes
	$Cdh1_i \rightarrow Cdh1$	Michaelis-Menten	$\frac{((k3 * Cdh1_i) * (1.0))}{(J3 + Cdh1_i)}$	<input type="checkbox"/>	
	$Cdh1 \rightarrow Cdh1_i$	Michaelis-Menten	$\frac{(((k4 * CycB) * Cdh1) * (1.0))}{(J4 + Cdh1)}$	<input type="checkbox"/>	

Observations

Fusion

- 1st round of submodel construction
- 1st round of fusion
- 2nd round of model construction (adding additional components)

Composition

- 1st round of submodel construction
- 1st round of composition (resolving components and first round of linking)
- 2nd round of model construction (updating reactions and adding new components)
- 2nd round of composition (resolving newly added components and second round of linking)

Aggregation

- 1st round of submodel construction
 - 1st round of aggregation (linking ports)
-