

Software for Formalized Molecular Interaction Maps (MIM)

SBGN 5.5

4/21/2010

Augustin Luna

augustin@mail.nih.gov

Joint work with:

Itir Karac, Margot Sunshine, Mirit Aladjem,
Kurt Kohn



Outline

- Objective: Present work on MIM software which may be useful for the development of libSBGN and can serve in the translation of MIM to SBGN
- Molecular Interaction Maps (MIM) Introduction
- Software Development for Formalized MIMs
 - Specification
 - Data model
 - Validation
 - API
 - Diagram editor

Objectives for MIM Software

- Create tools to facilitate the use of MIM
 - Tools for biologists to create/edit MIM diagrams
 - Tools for programmers
 - Tools used in conversions to/from other formats (e.g. SBGN, BioPAX, etc)
 - Create programming interface and validation software

Molecular Interaction Maps (MIM)

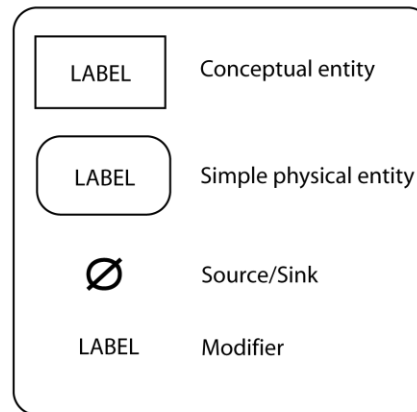
- A graphical notation for bioregulatory networks created by Kurt Kohn in 1999
- Small set of glyphs to provide information about interaction mechanisms
- Traceable diagrams; basis for SBGN ER diagrams
- MIM diagrams often have associated metadata

Formal MIM Specification

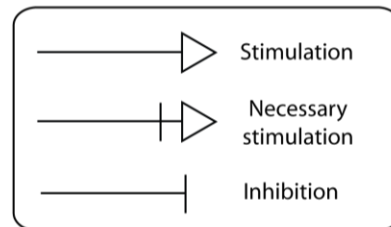
- Formalized specification based on the Kohn 2006 MIM specification
- Precludes the possibility for ad hoc creation of MIM glyphs and eliminates variations
- Create strict connection rules and provide definitions
- Some changes MIM glyphs necessitated by the software implementation

Basic MIM Glyphs

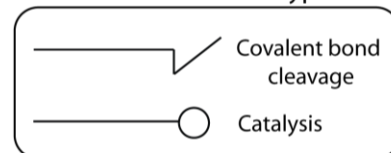
Entity Glyphs



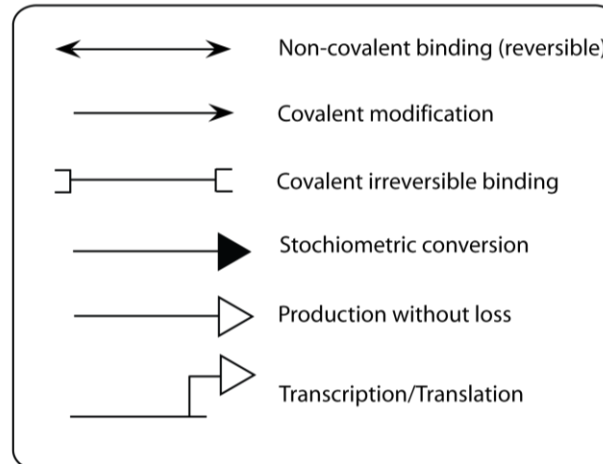
Contingency Glyphs



Reaction Path Glyphs



Reaction Glyphs



Data Modeling MIM

- Two possibilities for the MIM data model
 - BioPAX Format (OWL Ontology)
 - Emphasis on biological data; no graphics
 - Pathvisio/GPML Format (XML Schema)
 - Emphasis on visual characteristics
 - Embeds BioPAX data
 - Used by the Pathvisio diagram editor used by MIM
 - Similar concept to sXBL (SVG's XML Binding Language)
- XML schema chosen to simplify validation
- Two schemas: MIM visual and biological; biological data contained by the visual one

Example of MIM Dataset

```
<mimVis:Diagram mimVis:width="606.1" mimVis:height="308.0"
xmlns:mimVis="http://imp.nci.nih.gov/mim/mimVisLevel1">
  <mimVis:EntityGlyph entityType="SimplePhysicalEntity" mimVis:visId="ca53d" mimVis:displayName="CAMK"
centerX="166.0" centerY="167.0" mimVis:width="80.0" mimVis:height="20.0">
    <mimVis:mimBioRef>c1c63</mimVis:mimBioRef>
  </mimVis:EntityGlyph>
  ...

```

```
<mimVis:MimBio>
  <mimBio:Dataset xmlns:mimBio="http://imp.nci.nih.gov/mim/mimBioLevel1">
    <mimBio:title>CAMK Regulation</mimBio:title>
    <mimBio:SimplePhysicalEntity mimBio:bioid="c1c63">
      <mimBio:standardName>CAMK</mimBio:standardName>
    </mimBio:SimplePhysicalEntity>
    <mimBio:ExplicitComplexEntity mimBio:bioid="b2b4c">
      <mimBio:component ref="f3135"/>
      <mimBio:component ref="be41c"/>
    </mimBio:ExplicitComplexEntity>
    <mimBio:ControlledVocabulary mimBio:bioid="c75d6">
      <mimBio:term>GeneProduct</mimBio:term>
    </mimBio:ControlledVocabulary>
  </mimBio:Dataset>
</mimVis:MimBio>

```

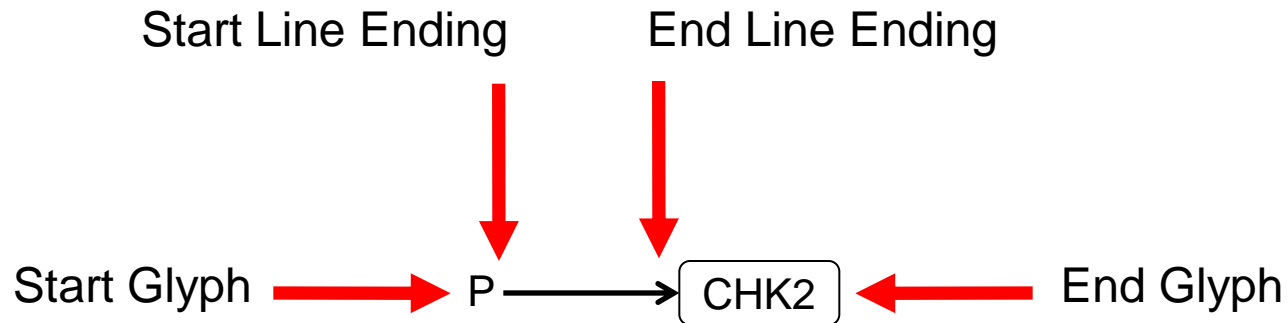
```
</mimVis:Diagram>
```

 MIM-Vis

 MIM-Bio

Validation of MIM Diagrams

- Validate for document well-formedness
 - Example: All entities must have a unique ID
- Validate in context using connection rules



Simple Schematron Validation Example

```
<iso:pattern
  name="num_ec_components"
  id="num_ec_components">
<iso:rule
  context="mimBio:ExplicitComplexEntity">
<iso:assert
  test="count(mimBio:component)
= 2"
  diagnostics="id.diag">Explicit
ComplexEntity must possess two
components.</iso:assert>
</iso:rule>
</iso:pattern>
```

Validation Stylesheet

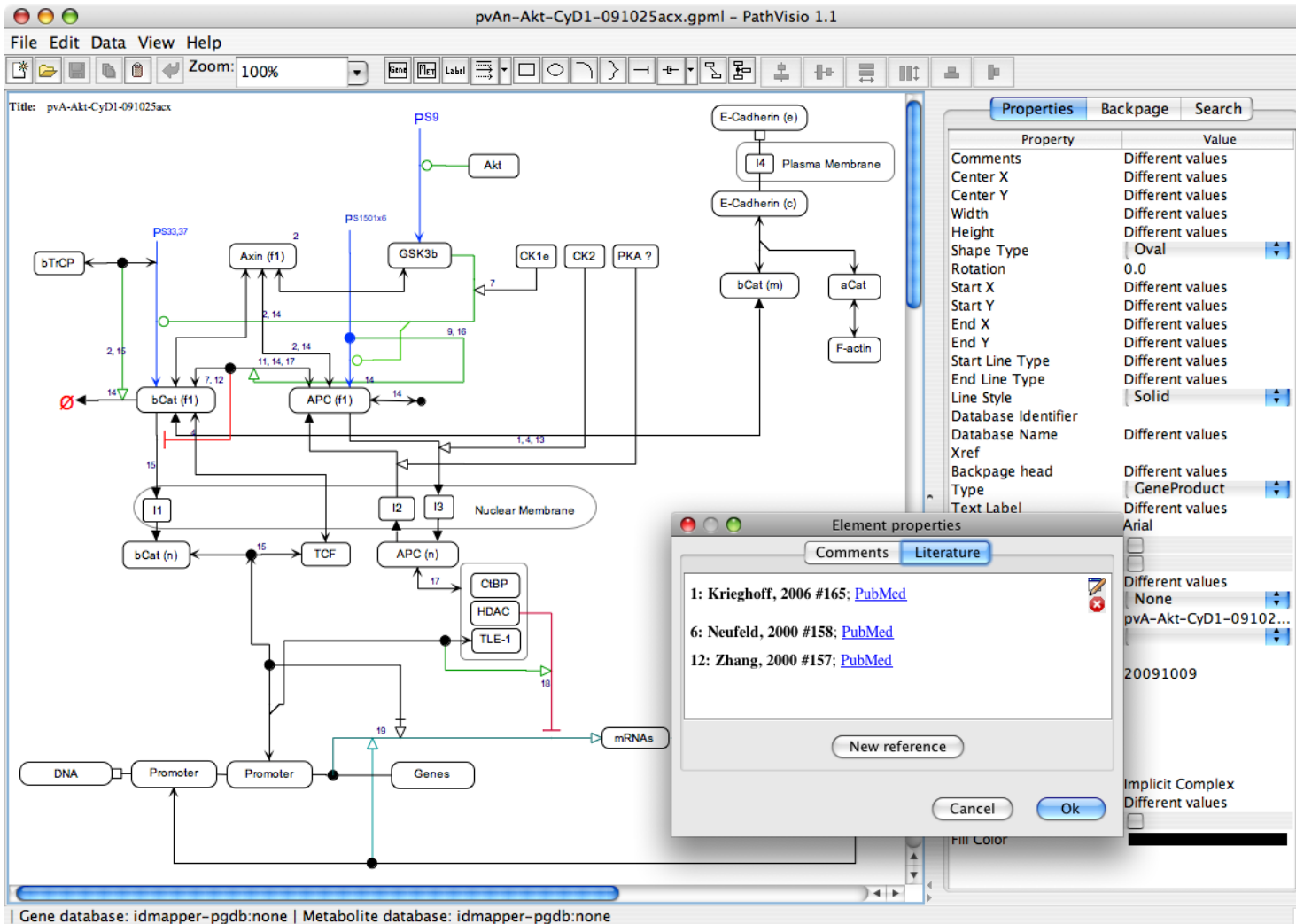
```
<svrl:active-pattern
  name="num_ec_components"/>
<svrl:fired-rule
  context="mimBio:ExplicitComplexEntity"/>
<svrl:failed-assert
  test="count(mimBio:component)
= 2" location="...">
<svrl:text>ExplicitComplexEntity
must possess two
components.</svrl:text>
<svrl:diagnostic-reference
  diagnostic="id.diag">...</svrl:d
iagnostic-reference>
</svrl:failed-assert>
```

Validation Report

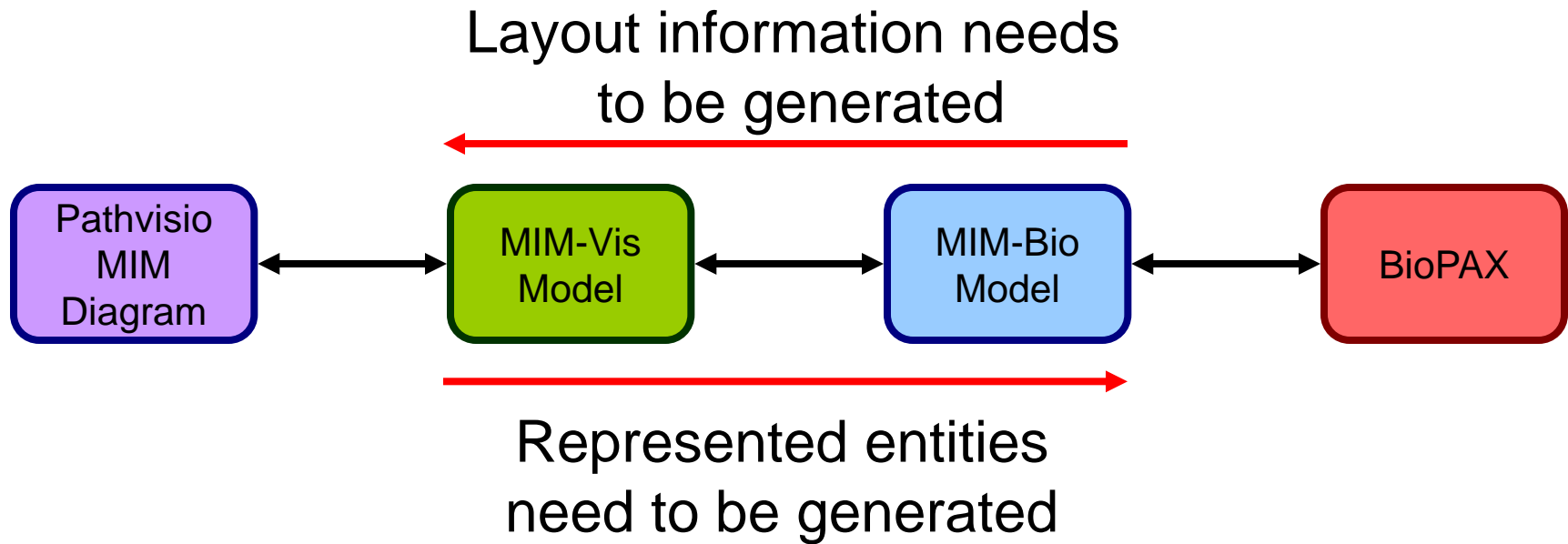
MIM Application Programming Interface

- Use off-the-shelf components and converters
 - Accommodate various programming languages
 - Attempt to minimize future maintenance tasks
- Usage of XML binding frameworks
 - XMLBeans: Java-to-XML binding framework
 - getFoo/setFoo-style accessors
 - Also provides cursor model to traverse XML datasets

AKT-CyD1 Interactions in Pathvisio-MIM



Issues in conversion of Pathvisio-MIM Diagram to/from BioPAX



Acknowledgements

- NCI LMP
 - Sohyoung Kim
 - David Kane
- MSKCC (BioPAX)
 - Emek Demir
 - BioPAX Community
- SBGN Community
- U of Maastricht (Pathvisio)
 - Martjin van Iersel
 - Pathvisio Developers
- Network Analysis and Schemas
 - Bogazici University
 - Turkan Haliloglu
 - Can Ozturan
 - Mine Edes
 - NCI-Frederick
 - Ruth Nussinov