

# libSBML

Sarah Keating



# SBML

provides the syntax

```
<listOfSpecies>  
  <species id="B" compartment="comp1" initialAmount="1.66057788110262e-21"\/>  
  <species id="BL" compartment="comp1" initialAmount="0"\/>  
</listOfSpecies>
```

Low info content

```
<kineticLaw>  
  <math xmlns="http://www.w3.org/1998/Math/MathML">  
    <apply>  
      <times\/>  
      <ci> comp1 </ci>  
      <ci> kf_0 </ci>  
      <ci> B </ci>  
    </apply>  
  </math>  
</kineticLaw>
```

Unregulated



# Need to annotate

Standard scheme for machine-readable annotations

Guidelines for model quality

- Authorship, publication info
- Links to other data resources

Defined  
by SBML

Semantics of the mathematics

# Need to annotate

Standard scheme for machine-readable annotations

**Guidelines for model quality**

- Authorship, publication info

- Links to other data resources

Defined  
by  
MIRIAM

Semantics of the mathematics

MIRIAM

Minimum

Information

Requested

In the

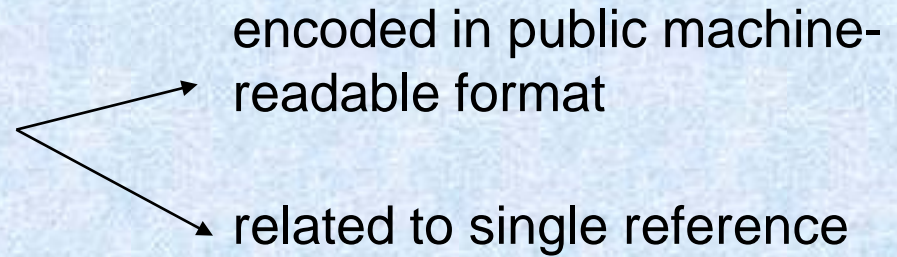
Annotation of

biochemical Models

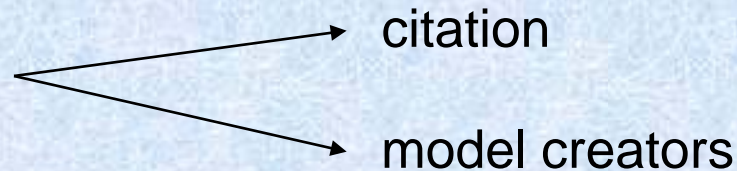


# MIRIAM

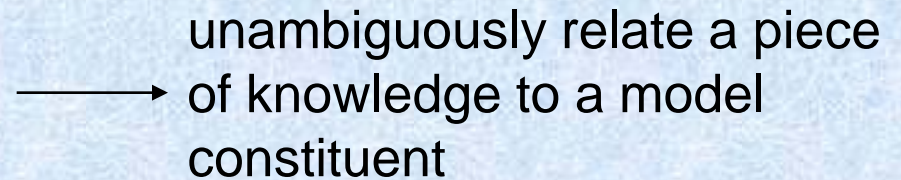
Reference correspondence



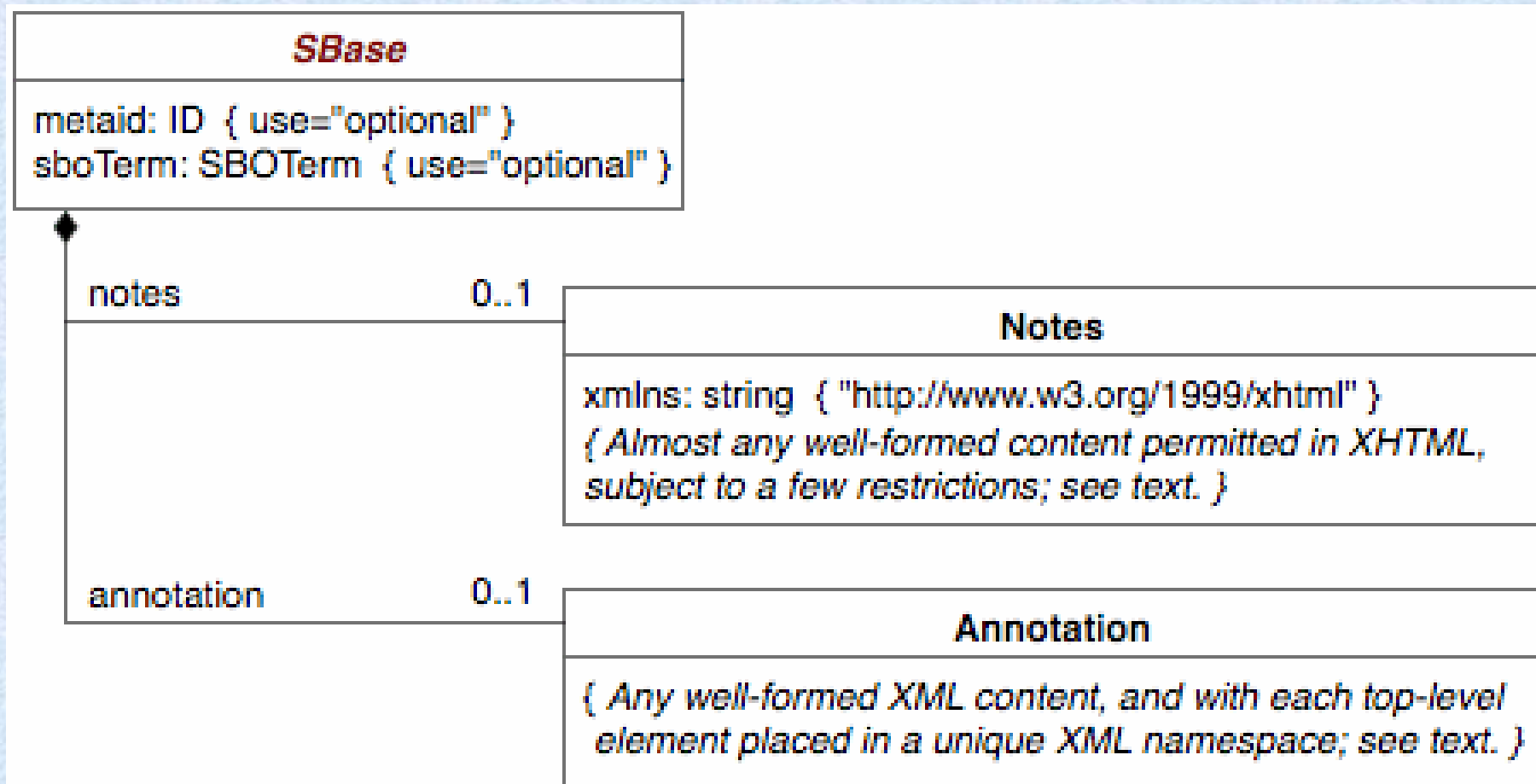
Attribution annotation



External resource annotation



# SBase – the SBML base class

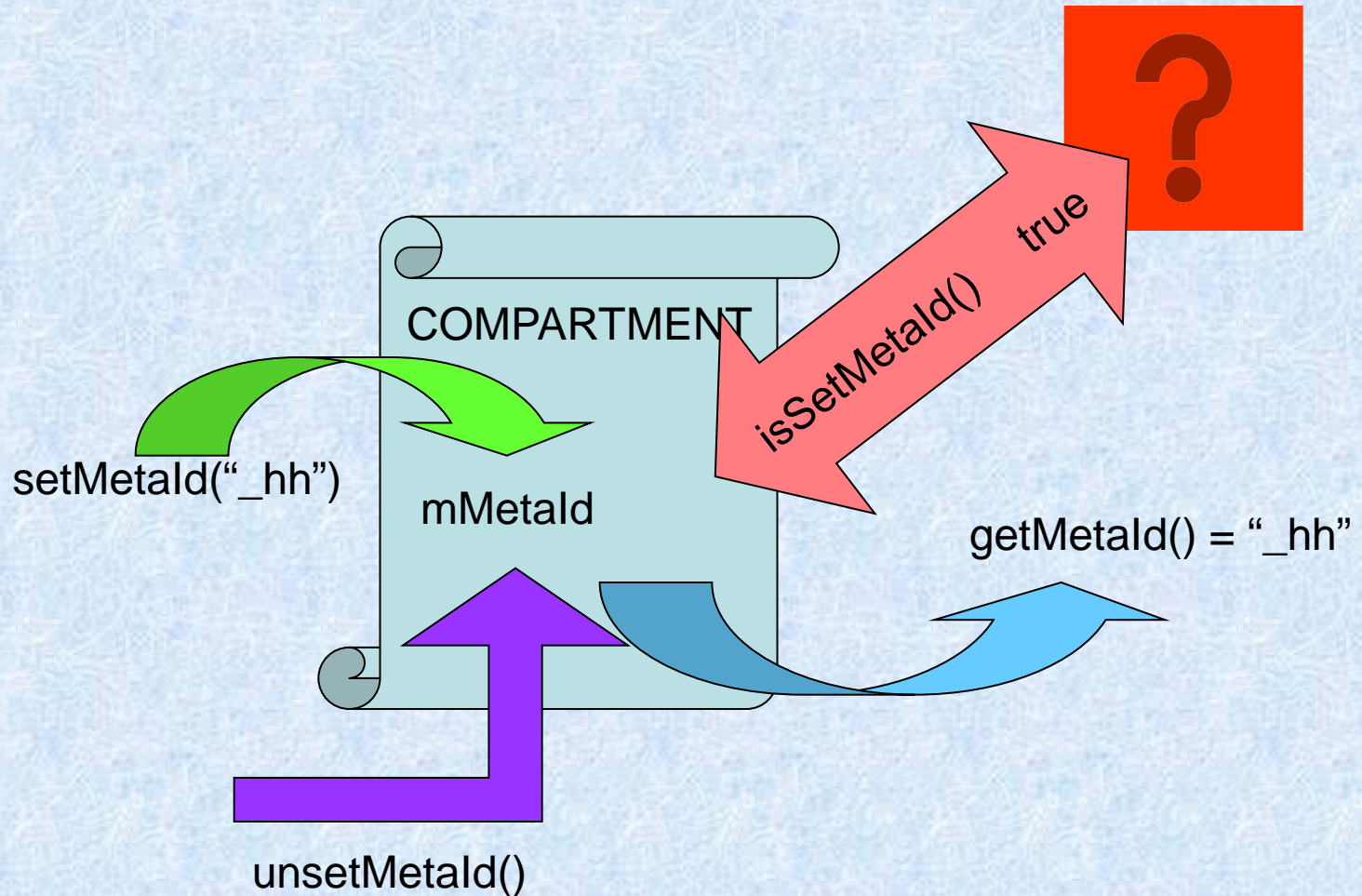


# SBase – the libSBML base class

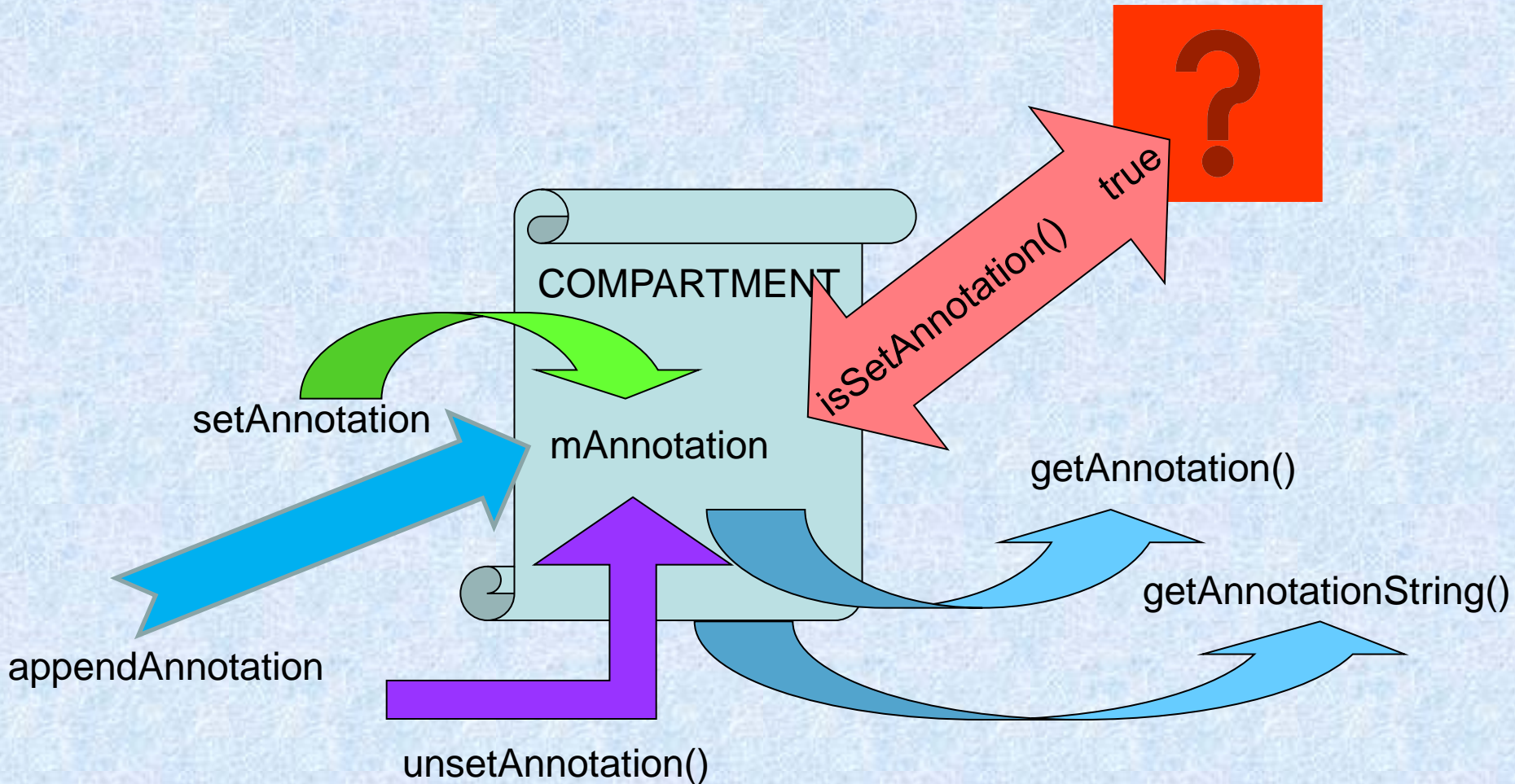
```
SBase.h* Start Page
(Unknown Scope)
2868
2869
2870  std::string      mMetaId;
2871  XMLNode*        mNotes;
2872  XMLNode*        mAnnotation;
2873  SBMLDocument*   mSBML;
2874  SBMLNamespaces* mSBMLNamespaces;
2875
2876  int mSBOTerm;
2877
2878  unsigned int mLine;
2879  unsigned int mColumn;
2880
2881  /* store the parent SBML object */
2882  SBase* mParentSBMLObject;
2883
2884  /* storing annotations */
2885  List * mCVTerms;
2886  ModelHistory* mHistory;
2887
2888  /* flag that allows object to know it has been deleted
2889   * for OS where the memory is still readable after a delete
2890   */
2891  bool mHasBeenDeleted;
```



# MetalD API



# Annotation API

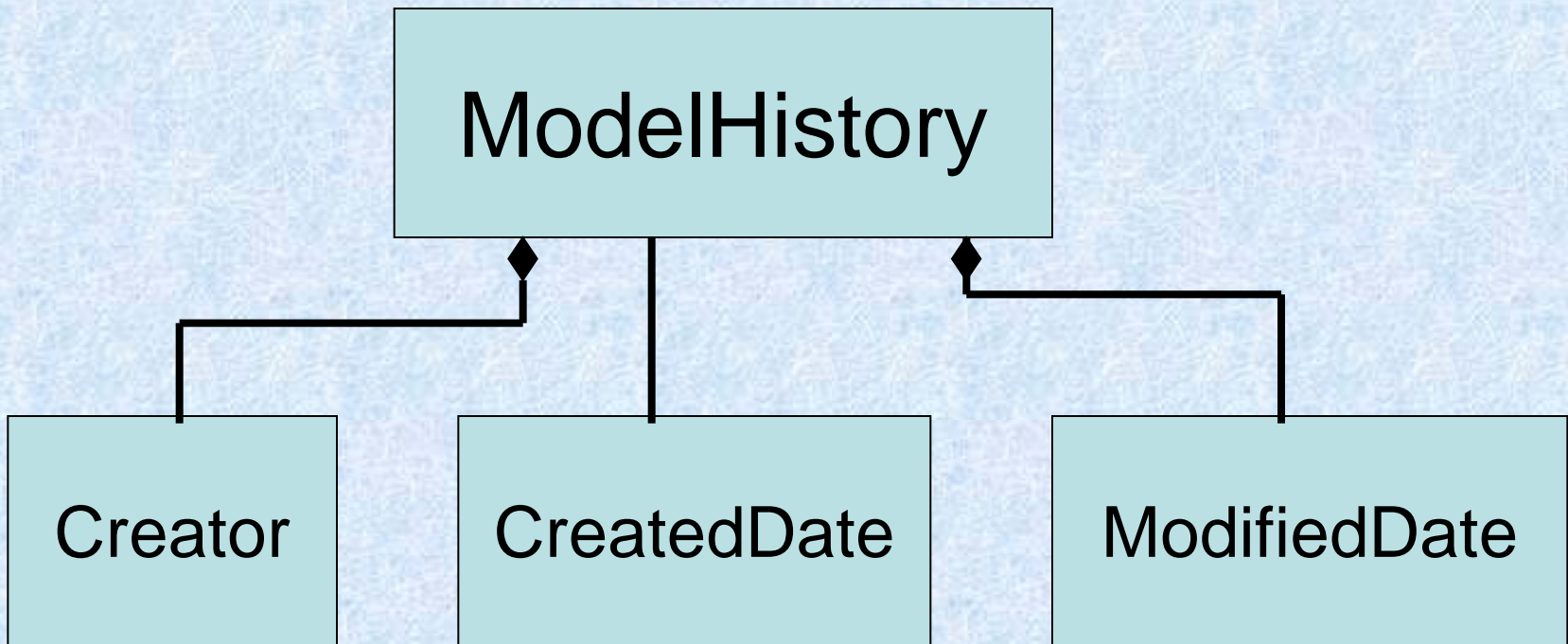


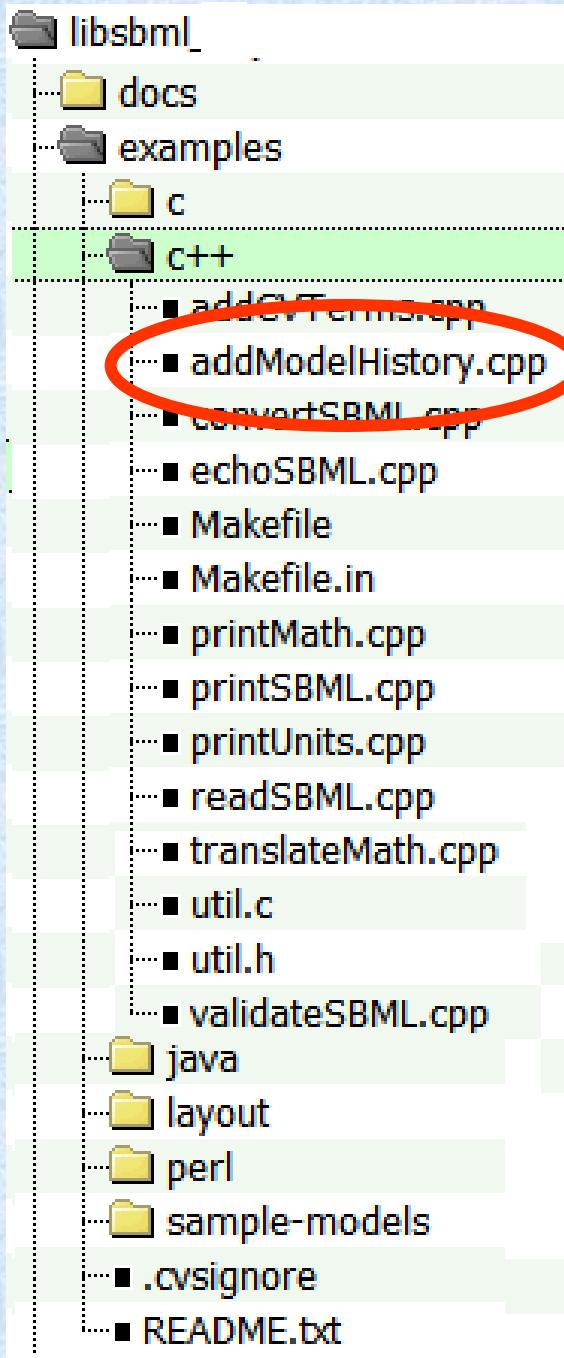
# Annotation API

```
SBase.h* Start Page
(Unknown Scope)
1172 * Sets the value of the "annotation" subelement of this SBML object.
1173 */
1174 virtual int setAnnotation (const XMLNode* annotation);
1175
1176
1177 /**
1178 * Sets the value of the "annotation" subelement of this SBML object.
1179 */
1180 virtual int setAnnotation (const std::string& annotation);
1181
1182
1183 /**
1184 * Appends the given @p annotation to the "annotation" subelement of this
1185 * object.
1186 */
1187 virtual int appendAnnotation (const XMLNode* annotation);
1188
1189
1190 /**
1191 * Appends the given @p annotation to the "annotation" subelement of this
1192 * object.
1193 */
1194 virtual int appendAnnotation (const std::string& annotation);
1195
1196
```

# MIRIAM compliant annotations

```
SBase.h* Start Page
(Unknown Scope)
2868
2869
2870     std::string      mMetaId;
2871     XMLNode*        mNotes;
2872     XMLNode*        mAnnotation;
2873     SBMLDocument*   mSBML;
2874     SBMLNamespaces* mSBMLNamespaces;
2875
2876     int mSBOTerm;
2877
2878     unsigned int mLine;
2879     unsigned int mColumn;
2880
2881     /* store the parent SBML object */
2882     SBase* mParentSBMLObject;
2883
2884     /* storing annotations */
2885     List * mCVTerms;
2886     ModelHistory* mHistory;
2887
2888     /* flag that allows object to know it has been deleted
2889      * for OS where the memory is still readable after a delete
2890      */
2891     bool mHasBeenDeleted;
2892
```





## addModelHistory.cpp

```
}  
  
d      = readSBML(argv[1]);  
errors = d->getNumErrors();  
  
if (errors > 0)  
{  
    cout << "Read Error(s):" << endl;  
    d->printErrors(cout);  
  
    cout << "Correct the above and re-run." << endl;  
}  
else  
{  
    ModelHistory * h = new ModelHistory();  
  
    ModelCreator *c = new ModelCreator();  
    c->setFamilyName("Keating");  
    c->setGivenName("Sarah");  
    c->setEmail("sbml-team@caltech.edu");  
    c->setOrganisation("University of Hertfordshire");  
  
    h->addCreator(c);  
  
    Date * date = new Date("1999-11-13T06:54:32");  
    Date * date2 = new Date("2007-11-31T06:54:00-02:00");  
  
    h->setCreatedDate(date);  
    h->setModifiedDate(date2);  
  
    d->getModel()->setModelHistory(h);  
  
    writeSBML(d, argv[2]);  
}
```



# <annotation>

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
  xmlns:bqmodel="http://biomodels.net/model-qualifiers/">
```

```
<rdf:Description rdf:about="#_000001">
```

metaid



```
<dc:creator rdf:parseType="Resource">
  <rdf:Bag>
    <rdf:li rdf:parseType="Resource">
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Keating</vCard:Family>
        <vCard:Given>Sarah</vCard:Given>
      </vCard:N>
      <vCard:EMAIL>sbml-team@caltech.edu</vCard:EMAIL>
      <vCard:ORG>
        <vCard:Orgname>University of Hertfordshire
        </vCard:Orgname>
      </vCard:ORG>
    </rdf:li>
  </rdf:Bag>
</dc:creator>
```





```
<vCard:N rdf:parseType="Resource">  
  <vCard:Family>Keating</vCard:Family>  
  <vCard:Given>Sarah</vCard:Given>  
</vCard:N>  
<vCard:EMAIL>sbml-team@caltech.edu</vCard:EMAIL>  
<vCard:ORG>  
  <vCard:Orgname>University of Hertfordshire</vCard:Orgname>  
</vCard:ORG>  
</rdf:li>  
</rdf:Bag>  
</dc:creator>
```

```
<dcterms:created rdf:parseType="Resource">  
  <dcterms:W3CDTF>  
    2005-02-02T14:56:11</dcterms:W3CDTF>  
</dcterms:created>
```

```
<dcterms:modified rdf:parseType="Resource">  
  <dcterms:W3CDTF>  
    2006-05-30T10:46:02</dcterms:W3CDTF>  
</dcterms:modified>
```

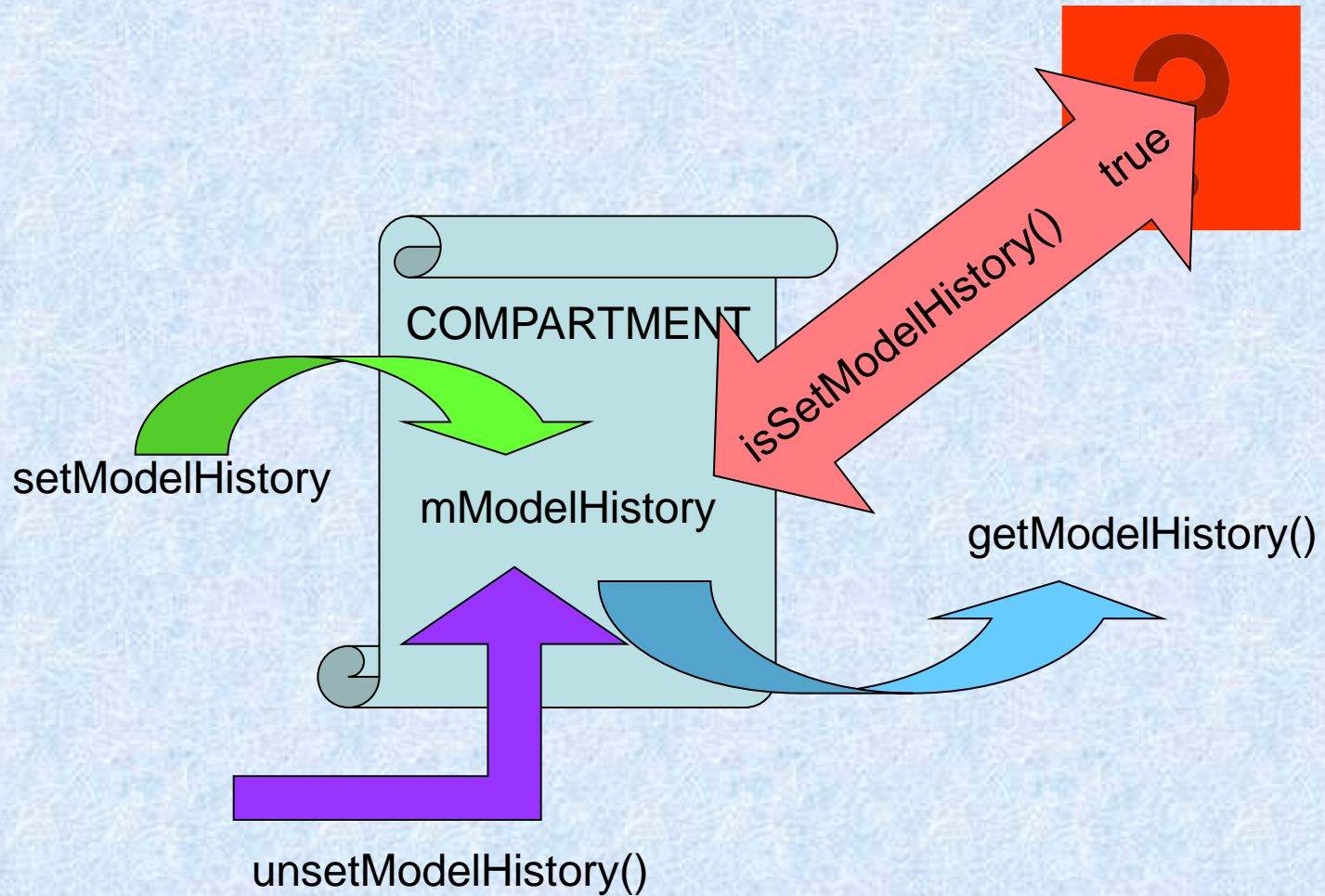


<annotation>

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
  xmlns:bqmodel="http://biomodels.net/model-qualifiers/">
```

```
<rdf:Description rdf:about="#_000001">
  <dc:creator rdf:parseType="Resource">
    <rdf:Bag>
      <rdf:li rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>Le Novere</vCard:Family>
          <vCard:Given>Nicolas</vCard:Given>
        </vCard:N>
        <vCard:EMAIL>lenov@ebi.ac.uk</vCard:EMAIL>
        <vCard:ORG>
          <vCard:Orgname>EMBL-EBI</vCard:Orgname>
        </vCard:ORG>
      </rdf:li>
    </rdf:Bag>
  </dc:creator>
  <dcterms:created rdf:parseType="Resource">
    <dcterms:W3CDTF>2005-02-02T14:56:11</dcterms:W3CDTF>
  </dcterms:created>
  <dcterms:modified rdf:parseType="Resource">
    <dcterms:W3CDTF>2006-05-30T10:46:02</dcterms:W3CDTF>
  </dcterms:modified>
</rdf:Description>
</rdf:RDF>
</annotation>
```

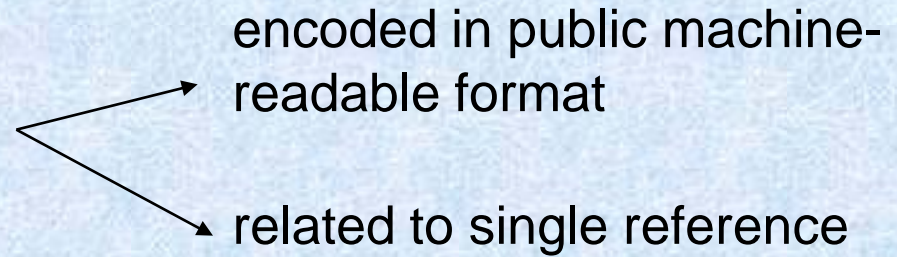
# ModelHistory API



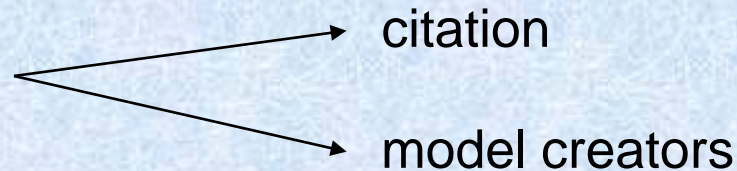


# MIRIAM

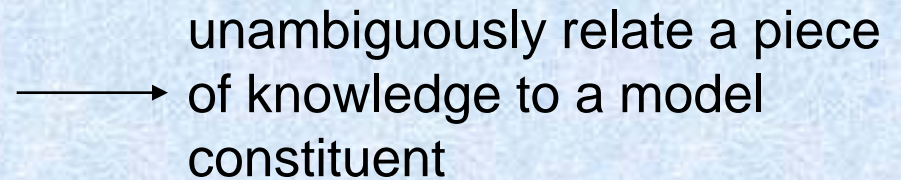
Reference correspondence



Attribution annotation



External resource annotation





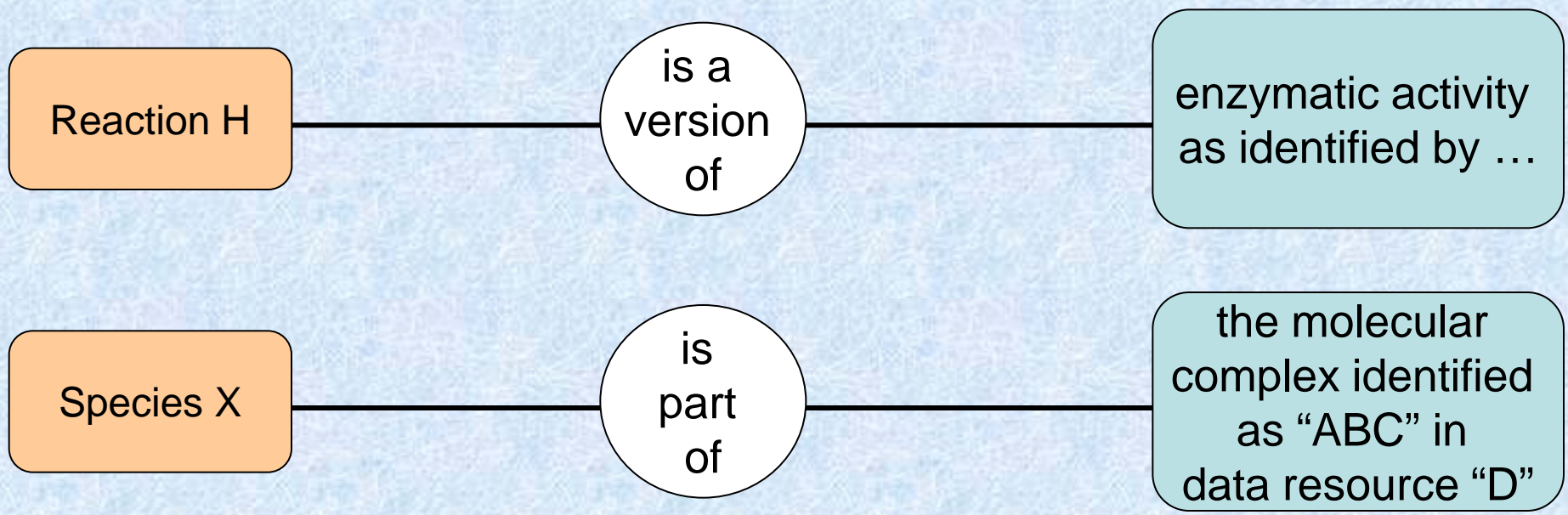
# MIRIAM

Annotation format

{data item}

{optional qualifier}

{entity identifier}



# CVTerm

## Public Member Functions

QualifierType\_t [getQualifierType](#) ()  
*Returns the Qualifier Type code for this CVTerm.*

ModelQualifierType\_t [getModelQualifierType](#) ()  
*Returns the [Model QualifierType](#) code for this CVTerm.*

BiolQualifierType\_t [getBiologicalQualifierType](#) ()  
*Returns the Biological QualifierType code for this CVTerm.*

XMLAttributes \* [getResources](#) ()  
*Returns the resources for this CVTerm.*

const XMLAttributes \* [getResources](#) () const  
*Returns the resources for this CVTerm.*

void [setQualifierType](#) (QualifierType\_t type)  
*Sets the "QualifierType\_t" of this CVTerm.*

void [setModelQualifierType](#) (ModelQualifierType\_t type)  
*Sets the "ModelQualifierType\_t" of this CVTerm.*

void [setBiologicalQualifierType](#) (BiolQualifierType\_t type)  
*Sets the "BiolQualifierType\_t" of this CVTerm.*

void [addResource](#) (std::string resource)  
*Adds a resource to the CVTerm.*

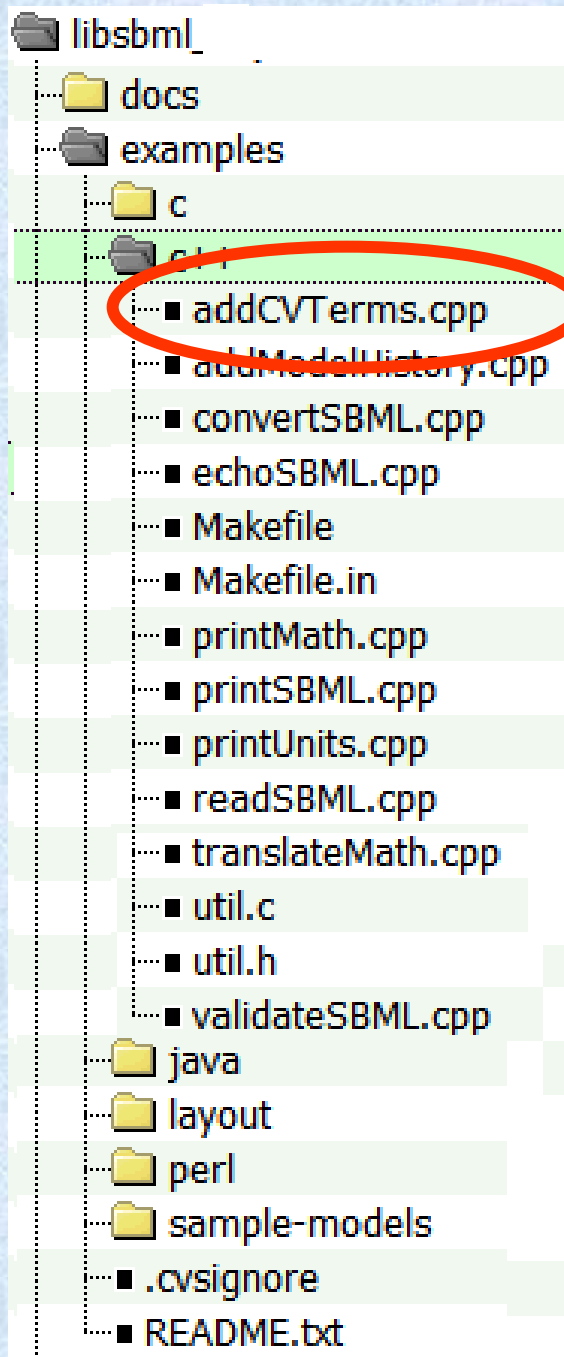
## Protected Attributes

XMLAttributes \* mResources

QualifierType\_t mQualifier

ModelQualifierType\_t mModelQualifier

BiolQualifierType\_t mBiolQualifier



```
{
    s = d->getModel()->getSpecies(0);

    CVTerm *cv = new CVTerm();
    cv->setQualifierType(BIOLOGICAL_QUALIFIER);
    cv->setBiologicalQualifierType(BQB_IS_VERSION_OF);
    cv->addResource("http://www.geneontology.org/#GO:0005892");

    CVTerm *cv2 = new CVTerm();
    cv2->setQualifierType(BIOLOGICAL_QUALIFIER);
    cv2->setBiologicalQualifierType(BQB_IS);
    cv2->addResource("http://www.geneontology.org/#GO:0005895");

    CVTerm *cv1 = new CVTerm();
    cv1->setQualifierType(BIOLOGICAL_QUALIFIER);
    cv1->setBiologicalQualifierType(BQB_IS_VERSION_OF);
    cv1->addResource("http://www.ebi.ac.uk/interpro/#IPR002394");

    s->addCVTerm(cv);
    s->addCVTerm(cv2);
    s->addCVTerm(cv1);

    writeSBML(d, argv[2]);
}

delete d;
return errors;
```

# <annotation>

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
  xmlns:bqmodel="http://biomodels.net/model-qualifiers/">
```

```
<rdf:Description rdf:about="#_species1">
```





<rdf:Description rdf:about="#\_species1">

<bqbiol:isVersionOf>

<rdf:Bag>

<rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0005892"/>

<rdf:li rdf:resource="urn:miriam:interpro:IPR002394"/>

</rdf:Bag>

</bqbiol:isVersionOf>

<bqbiol:is>

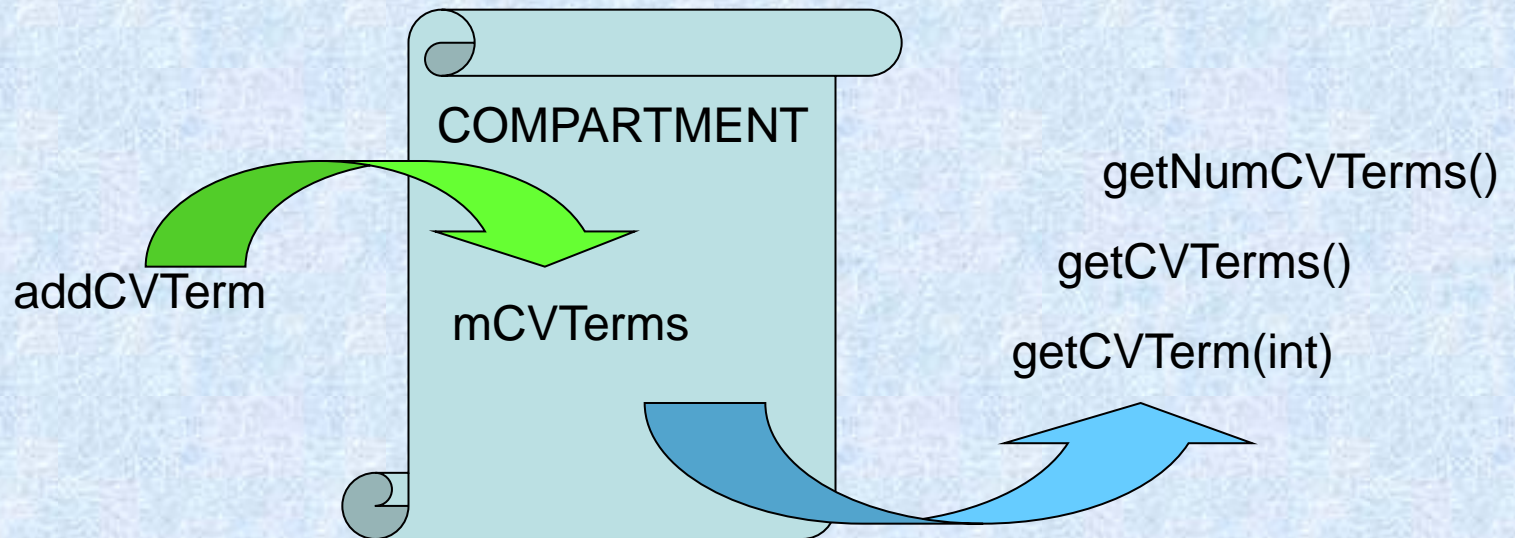
<rdf:Bag>

<rdf:li rdf:resource="urn:miriam:obo.go:GO%3A0005895"/>

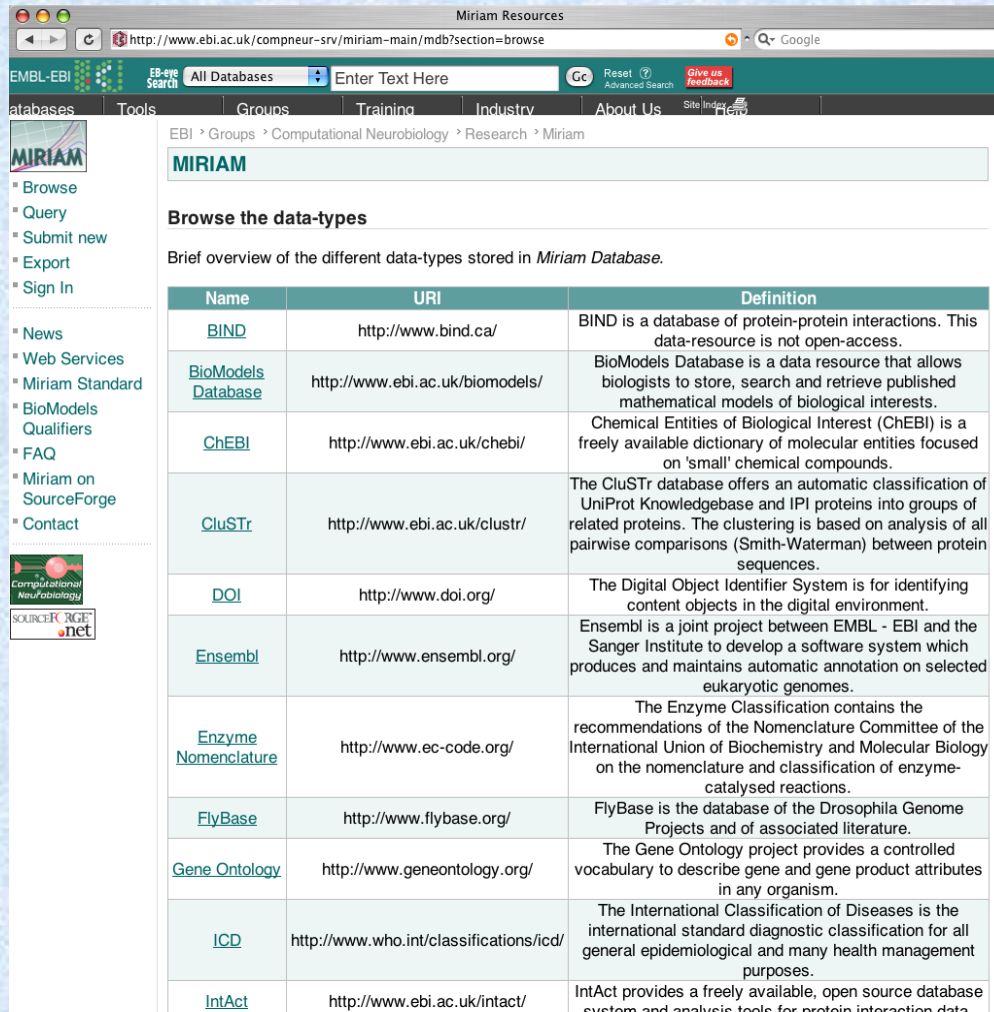
</rdf:Bag>

</bqbiol:is>

# CVTerm API



# MIRIAM Resources



The screenshot shows the MIRIAM Resources website. The main content area is titled "MIRIAM" and "Browse the data-types". Below this, there is a brief overview of the different data-types stored in the Miriam Database. A table lists the following databases:

Name	URI	Definition
<a href="#">BIND</a>	<a href="http://www.bind.ca/">http://www.bind.ca/</a>	BIND is a database of protein-protein interactions. This data-resource is not open-access.
<a href="#">BioModels Database</a>	<a href="http://www.ebi.ac.uk/biomodels/">http://www.ebi.ac.uk/biomodels/</a>	BioModels Database is a data resource that allows biologists to store, search and retrieve published mathematical models of biological interests.
<a href="#">ChEBI</a>	<a href="http://www.ebi.ac.uk/chebi/">http://www.ebi.ac.uk/chebi/</a>	Chemical Entities of Biological Interest (ChEBI) is a freely available dictionary of molecular entities focused on 'small' chemical compounds.
<a href="#">CluStr</a>	<a href="http://www.ebi.ac.uk/clustr/">http://www.ebi.ac.uk/clustr/</a>	The CluStr database offers an automatic classification of UniProt Knowledgebase and IPI proteins into groups of related proteins. The clustering is based on analysis of all pairwise comparisons (Smith-Waterman) between protein sequences.
<a href="#">DOI</a>	<a href="http://www.doi.org/">http://www.doi.org/</a>	The Digital Object Identifier System is for identifying content objects in the digital environment.
<a href="#">Ensembl</a>	<a href="http://www.ensembl.org/">http://www.ensembl.org/</a>	Ensembl is a joint project between EMBL - EBI and the Sanger Institute to develop a software system which produces and maintains automatic annotation on selected eukaryotic genomes.
<a href="#">Enzyme Nomenclature</a>	<a href="http://www.ec-code.org/">http://www.ec-code.org/</a>	The Enzyme Classification contains the recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzyme-catalysed reactions.
<a href="#">FlyBase</a>	<a href="http://www.flybase.org/">http://www.flybase.org/</a>	FlyBase is the database of the Drosophila Genome Projects and of associated literature.
<a href="#">Gene Ontology</a>	<a href="http://www.geneontology.org/">http://www.geneontology.org/</a>	The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism.
<a href="#">ICD</a>	<a href="http://www.who.int/classifications/icd/">http://www.who.int/classifications/icd/</a>	The International Classification of Diseases is the international standard diagnostic classification for all general epidemiological and many health management purposes.
<a href="#">IntAct</a>	<a href="http://www.ebi.ac.uk/intact/">http://www.ebi.ac.uk/intact/</a>	IntAct provides a freely available, open source database system and analysis tools for protein interaction data.

[www.ebi.ac.uk/MIRIAM](http://www.ebi.ac.uk/MIRIAM)



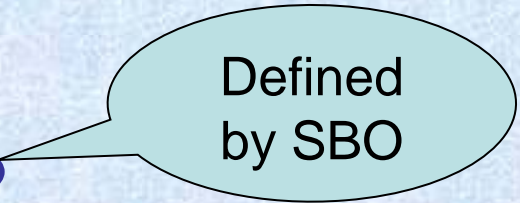
# Need to annotate

Standard scheme for machine-readable annotations

Guidelines for model quality

- Authorship, publication info
- Links to other data resources

Semantics of the mathematics



Defined  
by SBO



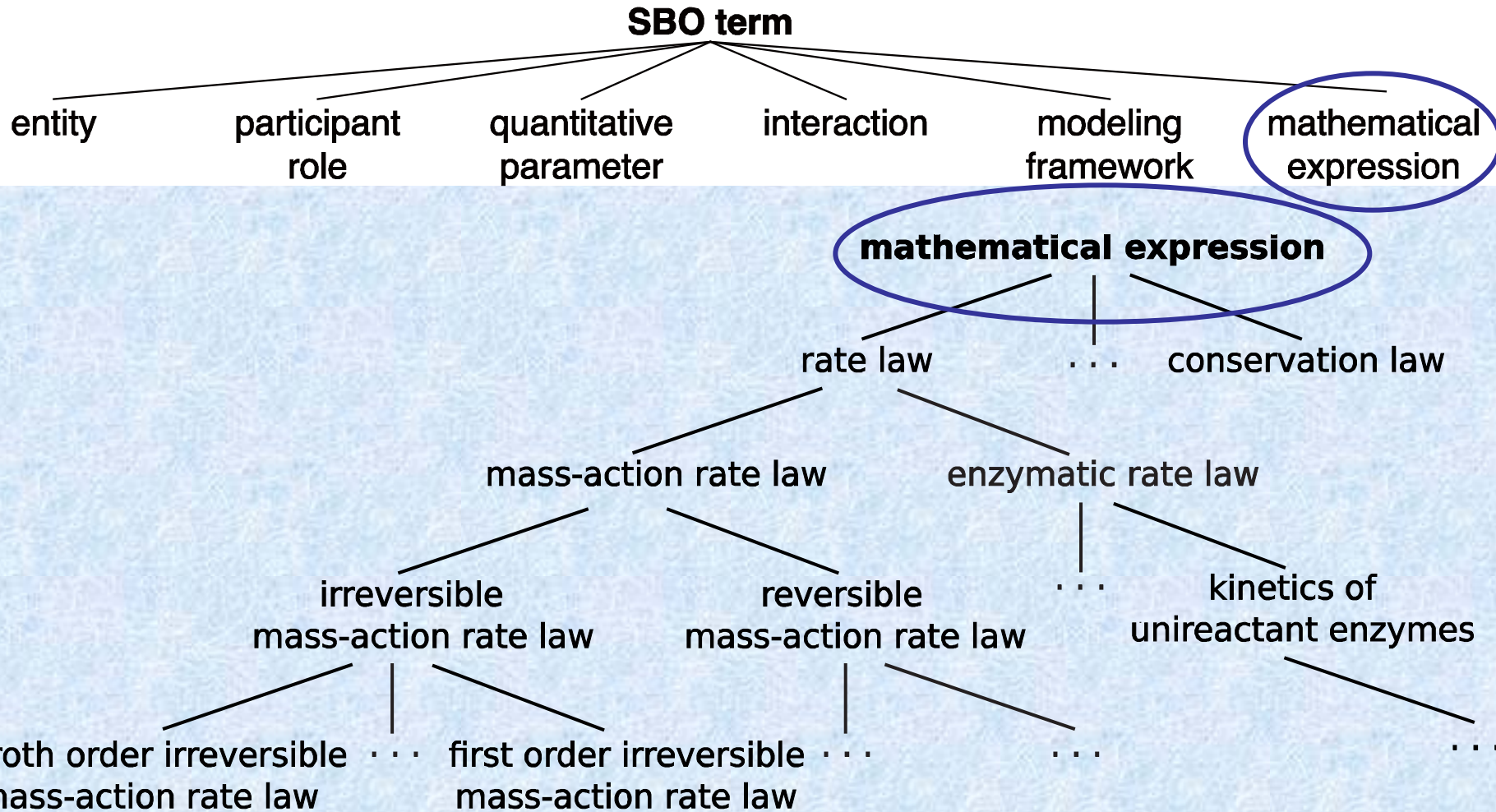
SBO

**S**ystems

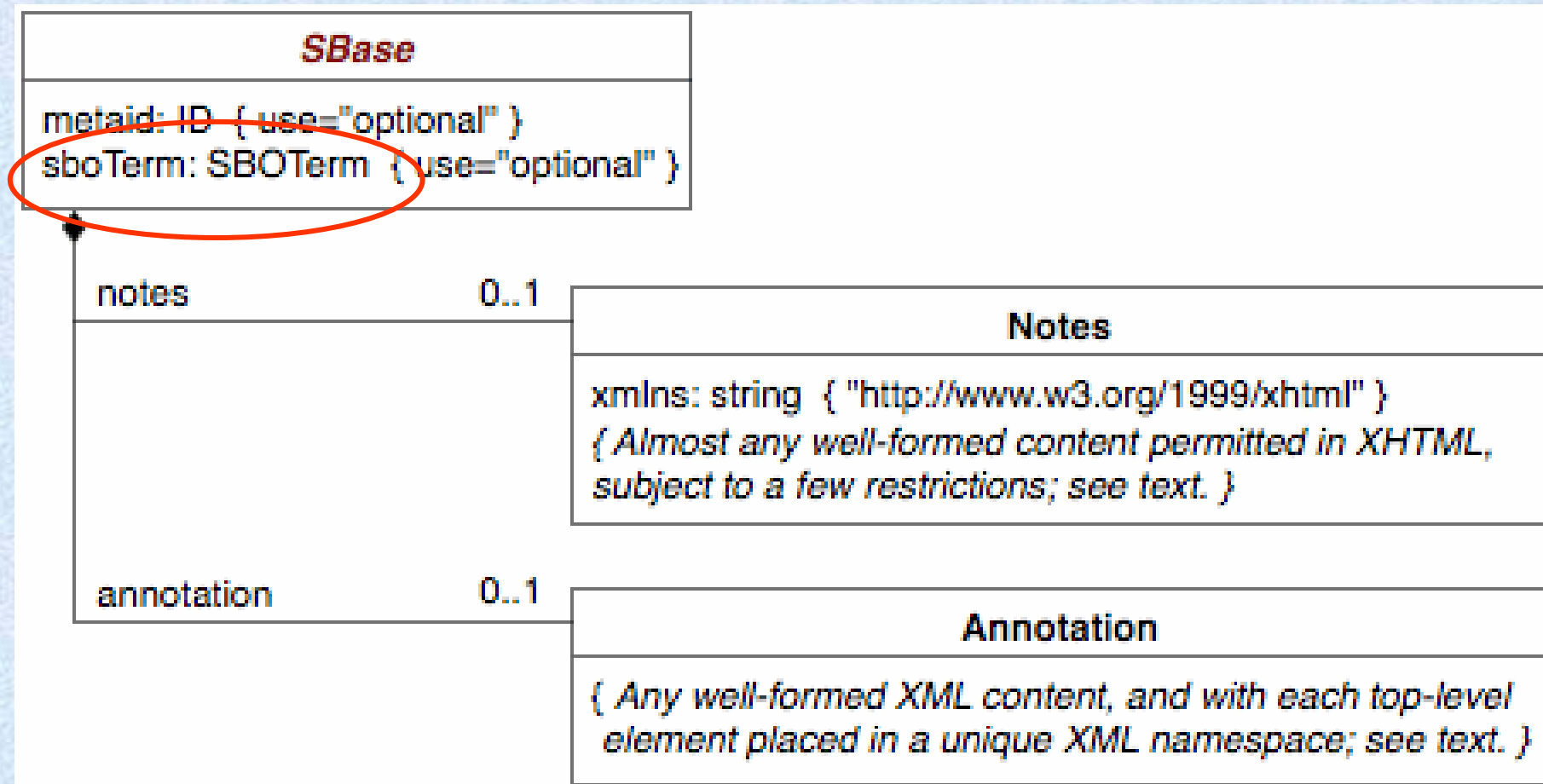
**B**iology

**O**ntology

# SBO



# SBase – the SBML base class

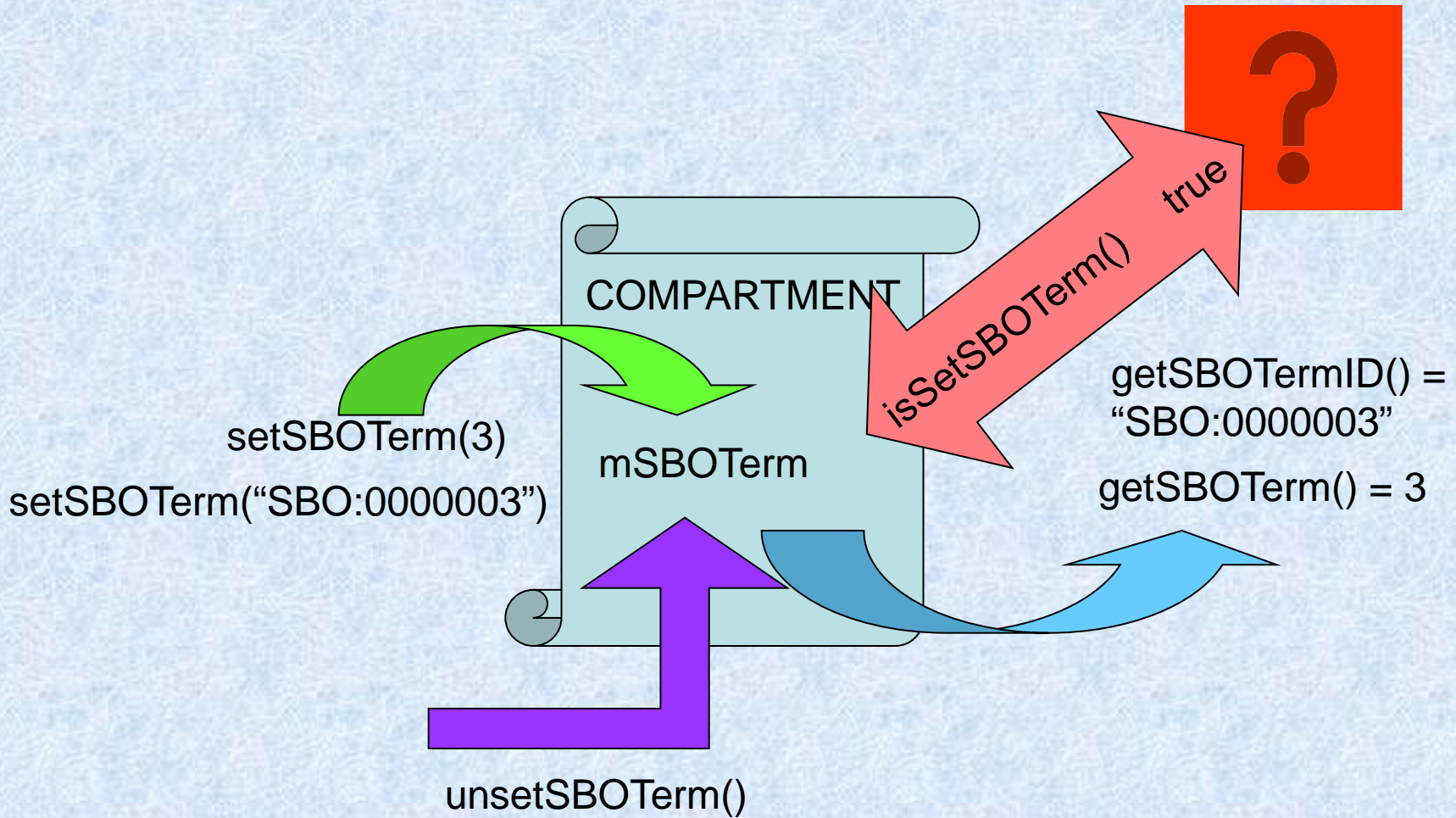


# SBase – the libSBML base class

```
SBase.h* Start Page
(Unknown Scope)
2868
2869
2870     std::string      mMetaId;
2871     XMLNode*        mNotes;
2872     XMLNode*        mAnnotation;
2873     SBMLDocument*   mSBML;
2874     SBMLNamespaces* mSBMLNamespaces;
2875
2876     int mSBOTerm;
2877
2878     unsigned int mLine;
2879     unsigned int mColumn;
2880
2881     /* store the parent SBML object */
2882     SBase* mParentSBMLObject;
2883
2884     /* storing annotations */
2885     List * mCVTerms;
2886     ModelHistory* mHistory;
2887
2888     /* flag that allows object to know it has been deleted
2889     * for OS where the memory is still readable after a delete
2890     */
2891     bool mHasBeenDeleted;
```



# SBOTerm API



# Acknowledgements - MIRIAM

## ■ EBI

- Nicolas Le Novère
- Alexander Broicher
- Mélanie Courtot
- Marco Donizelli
- Arnaud Henry
- Chen Li
- Lu Li
- Camille Laibe
- Nicolas Rodriguez

## ■ SBML team (Caltech)

- Michael Hucka
- Andrew Finney
- Benjamin Borstein
- Harish Dharuri
- Enuo He
- Sarah Keating
- Maria Schilstra
- Bruce Shapiro

## ■ NCBS (Bangalore)

- Upinder Bhalla
- Harsha Rani

## ■ University of Washington

- Herbert Sauro

## ■ Vienna TBI

- Rainer Machne

## ■ Systems Biology Institute (Tokyo)

- Hiroaki Kitano
- Akira Funahashi

## ■ JWS Online (Stellenbosh)

- Jacky Snoep

## ■ Virtual Cell (UCHC)

- Ion Moraru

## ■ Journals supporting BioModels Database

- Molecular Systems Biology
- All PLoS Journals
- All BioMedCentral Journals

## ■ Programs used for curation

- CellDesigner/SBMLodeSolver
- COPASI
- Jarnac/JDesigner
- MathSBML
- RoadRunner
- SBMLeditor
- XPP-Aut

The community of Systems Biology for their contributions of models and comments.



# Acknowledgements



Akiya Jouraku  
Keio, Japan



Frank Bergmann  
Caltech, USA



Mike Hucka  
Caltech, USA