libSBML & SBML
Sarah Keating
SBML

A machine-readable format for representing computational models in systems biology
libSBML

- API library for working with SBML
libSBML

• read
libSBML

- read
- create
libSBML

- read
- create
- manipulate
libSBML

- read
- create
- manipulate
- convert between levels/versions
libSBML

- read
- create
- manipulate
- convert between levels/versions
- write
libSBML

- read
- create
- manipulate
- convert between levels/versions
- write
- validate
libSBML

- API library for working with SBML
and the benefits to SBML
Software supporting SBML

(counted in middle of each year)
Software supporting SBML

libSBML 1.0.0
Software supporting SBML

libSBML 1.0.0

(counted in middle of each year)
libSBML
libSBML

• closely mimic SBML structure


ModelHistory* mHistory;

ListOfFunctionDefinitions mFunctionDefinitions;
ListOfUnitDefinitions mUnitDefinitions;
ListOfCompartmentTypes mCompartmentTypes;
ListOfSpeciesTypes mSpeciesTypes;
ListOfCompartment mCompartments;
ListOfSpecies mSpecies;
ListOfParameters mParameters;
ListOfInitialAssignments mInitialAssignments;
ListOfRules mRules;
ListOfConstraints mConstraints;
ListOfReactions mReactions;
ListOfEvents mEvents;

ListFormulaUnitsData mFormulaUnitsData;

#ifdef USE_LAYOUT
   ListOfLayouts mLayouts;
#endif /* USE_LAYOUT */
id: SId
name: string { use="optional" }
compartmentType: SId { use="optional" }
spatialDimensions: int { use="optional" default="3" }
size: double { use="optional" }
units: UnitSId { use="optional" }
outside: SId { use="optional" }
constant: boolean { use="optional" default="true" }
class LIBSBML_EXTERN Compartment : public SBase
{

protected:
    /** @cond doxygen-libsbml-internal */

    /**
     * Subclasses should override this method to read values from the given
     * XMLAttributes set into their specific fields. Be sure to call your
     * parents implementation of this method as well.
     */
    virtual void readAttributes (const XMLAttributes& attributes);

    /**
     * Subclasses should override this method to write their XML attributes
     * to the XMLOutputStream. Be sure to call your parents implementation
     * of this method as well.
     */
    virtual void writeAttributes (XMLOutputStream& stream) const;

    std::string mCompartmentType;
    unsigned int mSpatialDimensions;
    double mSize;
    std::string mUnits;
    std::string mOutside;
    bool mConstant;

    bool mIsSetSize;

    /** @endcond doxygen-libsbml-internal */
};
class LIBSBML_EXTERN Compartment : public SBase
{

protected:
    /** @cond doxygen-libsbml-internal */

    /**
     * Subclasses should override this method to read values from the given
     * XMLAttributes set into their specific fields. Be sure to call your
     * parents implementation of this method as well.
     */
    virtual void readAttributes (const XMLAttributes& attributes);

    /**
     * Subclasses should override this method to write their XML attributes
     * to the XMLOutputStream. Be sure to call your parents implementation
     * of this method as well.
     */
    virtual void writeAttributes (XMLOutputStream& stream) const;

    std::string mCompartmentType;
    unsigned int mSpatialDimensions;
    double mSize;
    std::string mUnits;
    std::string mOutside;
    bool mIsSetSize;
    bool mConstant;

    /** @endcond doxygen-libsbml-internal */
};
libSBML

• closely mimic SBML structure

• different language bindings
- Standard ANSI C++
- C
- C#
- MATLAB
- Octave
- Perl
- Python
- Java
- Ruby
libSBML

• closely mimic SBML structure

• different language bindings

• XML parsers
XML Layer

XML Parser

XMLInput Stream

XMLOutput Stream

XMLToken

XMLNode

XMLAttributes

Expat Attributes

Xerces Attributes

LibXML Attributes
libSBML

- closely mimic SBML structure
- different language bindings
- XML parsers
- develop in line with specification
libSBML

- closely mimic SBML structure
- different language bindings
- XML parsers
- develop in line with specification
- backwards compatibility
Acknowledgements

• Frank Bergmann
• Bill Denny
• Christoph Flamm
• Akira Funahashi
• Ralph Gauges
• Martin Ginkel
• Alex Gutteridge
• Stefan Hoops
• Moriyoshi Koizumi
• Ben Kovitz
• Rainer Machné
• Nicolas Rodriguez
Acknowledgements

Ben Bornstein
JPL, USA

Akiya Jouraku
Keio, Japan

Mike Hucka
Caltech, USA