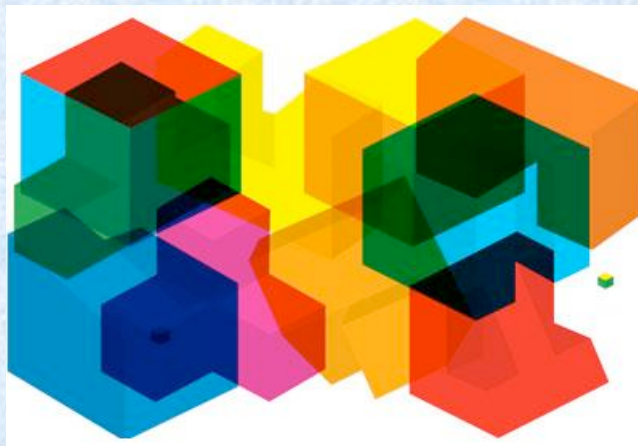


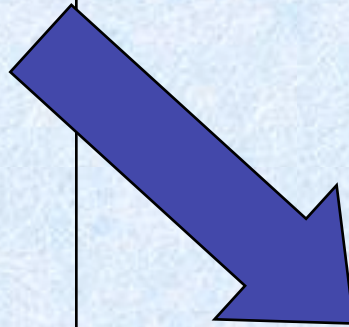
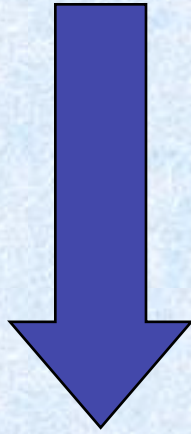
# libSBML

## Modularization



[http://www.sbml.org/SBML\\_Projects/libSBML/Modularization](http://www.sbml.org/SBML_Projects/libSBML/Modularization)

**libSBML 3.1.1**



**libSBML 4.0.0**



libSBML

core

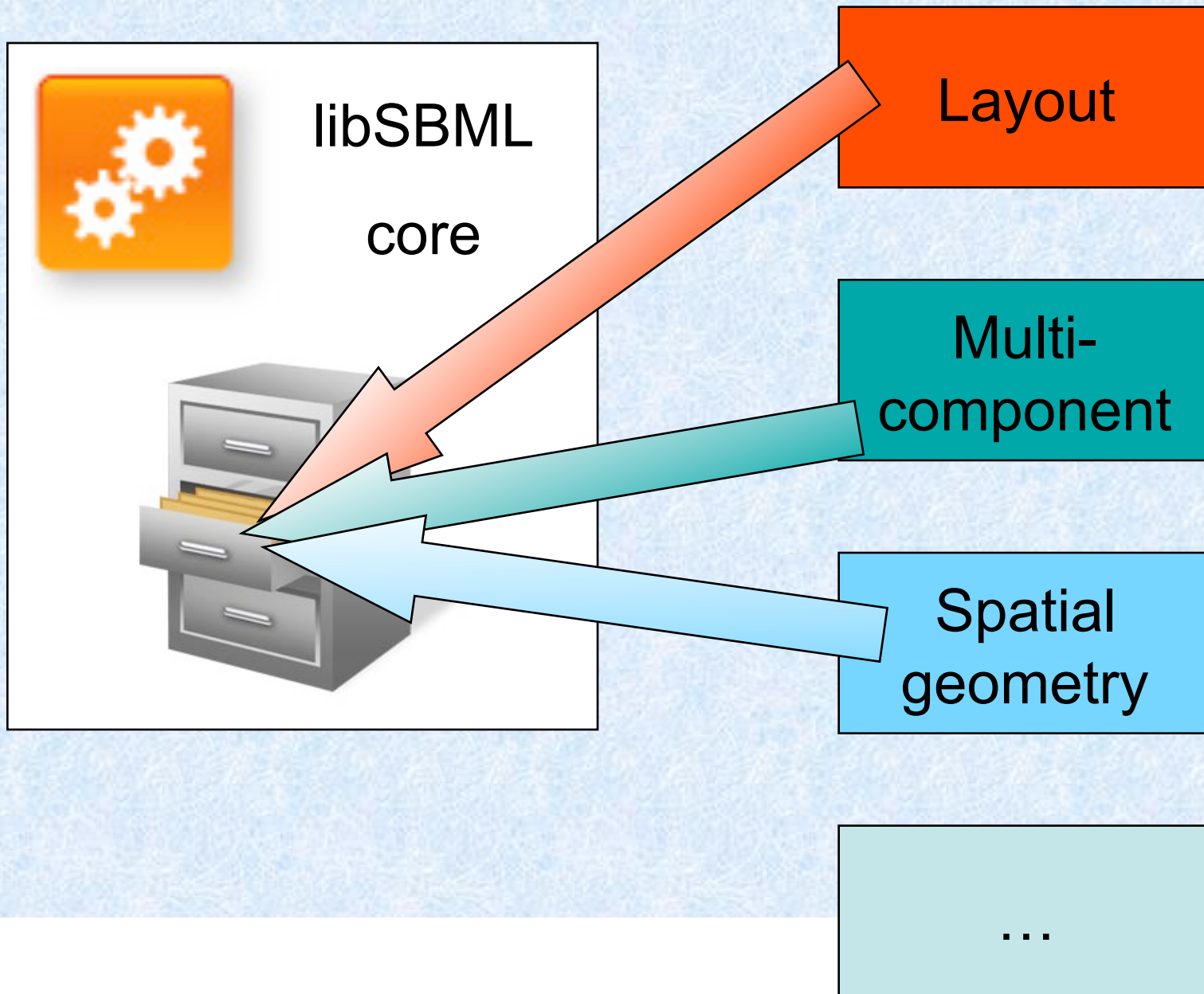
Layout

Multi-  
component

Spatial  
geometry

...

# Build ...



# Run ...



libSBML  
core



```
Initialize('Layout')
```

```
{
```

```
  checkForLibrary('Layout')
```

```
  loadLibrary('Layout')
```

```
}
```



# Issue 1

What ??

**Module name**

**Namespace**

**Prefix**

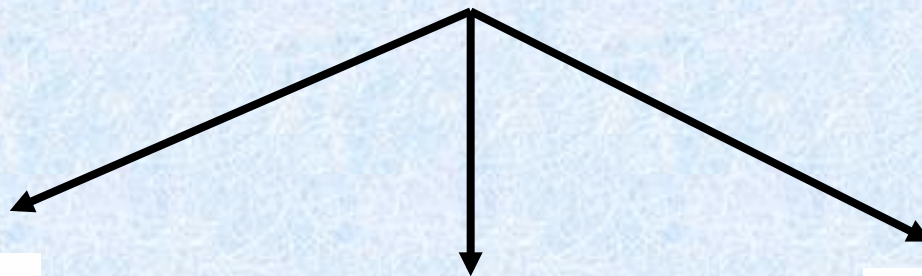
**Dependencies**

...

How ??

# Issue 2

loadLibrary



# Functionality

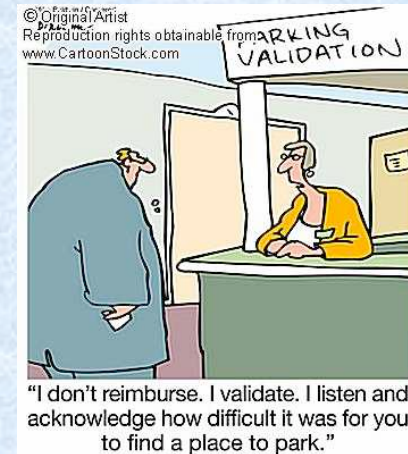
- read



- write



- validate



- create



- convert

L1 to L2v2

```
void  
SBase::read (  
{
```

```
...
```

```
    if (object)
```

```
    { ... }
```

```
    else if ( !( readOtherXML(stream) ... )
```

```
    { ... }
```

```
    else
```

```
    {
```

```
        passControlToOtherLibraries(stream);
```

```
    }
```

```
<listOfNewComponent>
```

```
    <newComponent id="nc1" .... />
```

```
    <newComponent id="nc1" .... />
```

```
</listOfNewComponents>
```

```
<species id="S1" layout:newAttribute="..."/>
```

# write

```
Species::writeAttributes(XMLAttributes& attributes)  
{  
    existing code  
  
    ....  
    passControlToOtherLibraries(attributes);  
}
```

# validate

```
SBMLDocument::checkConsistency()
```

```
{
```

```
    existing code
```

```
    ....
```

```
    passControlToOtherLibraries();
```

```
}
```

# convert

Issue 3

- setLevelAndVersion
- validation/compatability
- write

## Issue 4

# create

- default level/version/namespace

Level 2 Version 3

```
Model *model = new Model();  
model->setName("foo");  
Document *document = new SBMLDocument();  
document->setLevelAndVersion(1, 2);  
document->addModel(model);
```

Level 1 Version 2

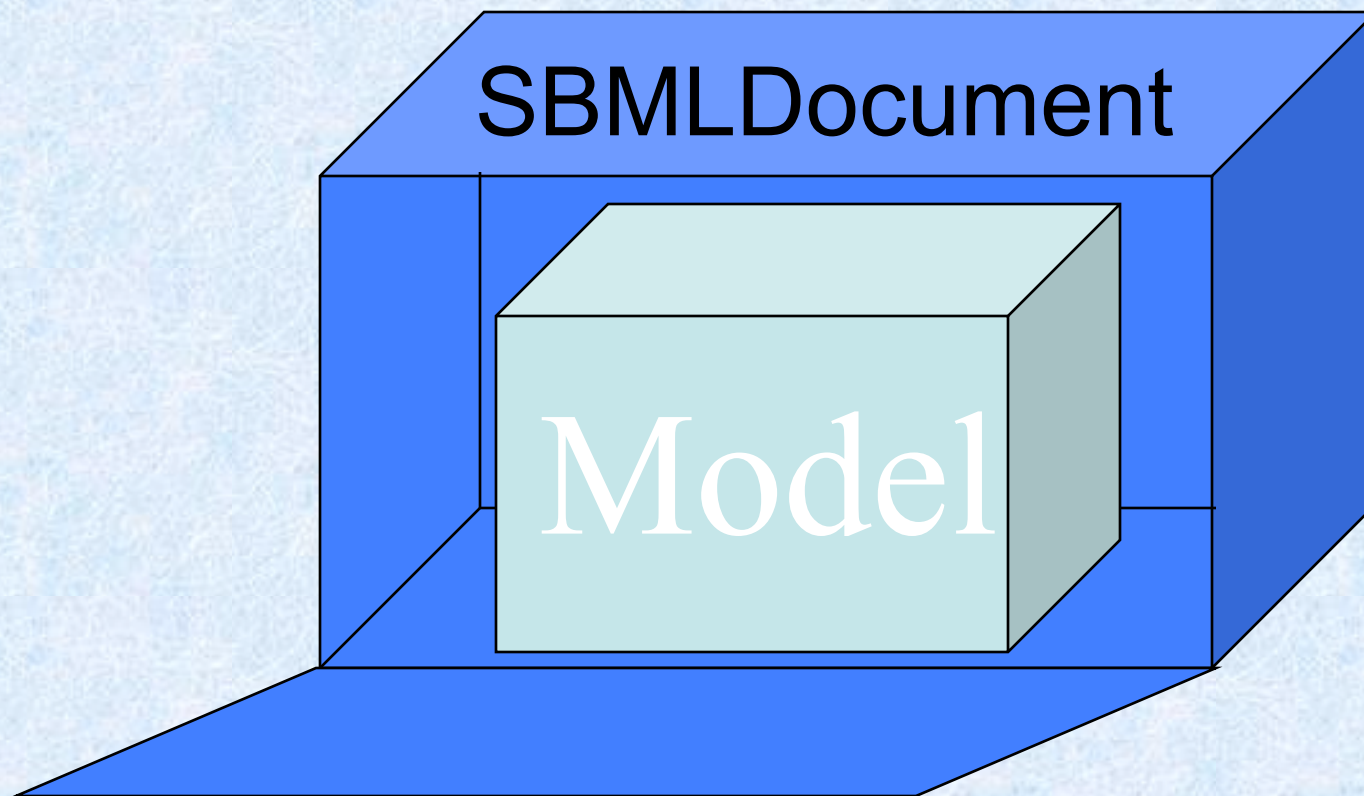
# Issue 5

**SBMLDocument**



[http://www.sbml.org/SBML\\_Projects/libSBML/Modularization](http://www.sbml.org/SBML_Projects/libSBML/Modularization)

# Issue 5



[http://www.sbml.org/SBML\\_Projects/libSBML/Modularization](http://www.sbml.org/SBML_Projects/libSBML/Modularization)

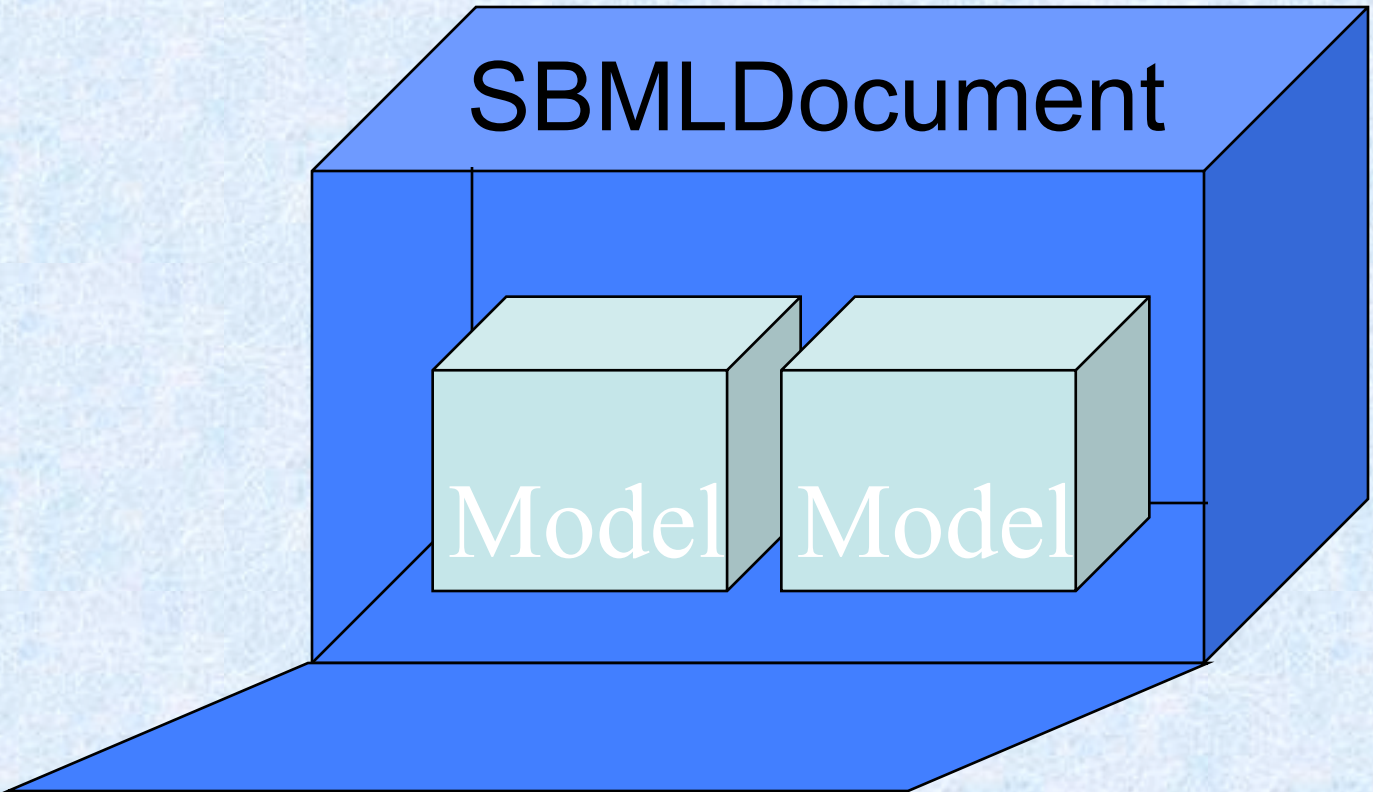
# Issue 5

**SBMLDocument**



[http://www.sbml.org/SBML\\_Projects/libSBML/Modularization](http://www.sbml.org/SBML_Projects/libSBML/Modularization)

# Issue 5



# Language

- C++
- C implementation
- local.i – language bindings

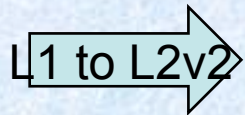
# Issues



- internal registry



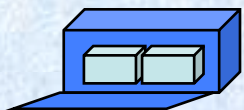
- multi-platform



- conversion



- default level/version/namespace



- multiple models per document