

Modularizing SBML

◎ The future of SBML

- Major extensions have been requested and are sorely needed
- But SBML is complicated enough
 - Only the largest-scale efforts have implemented most of Level 2

◎ Idea (a natural one): modularize

- Layer additional language extensions on top of a core
 - Similar to XML: XML is core, then there's XPath, MathML, etc.
- Additional benefits:
 - Can decouple development of individual extensions
 - Software can recognize & work with extensions it needs

Candidate mechanism: XML namespaces

```
<sbml xmlns="http://www.sbml.org/sbml/level3/version1"
      xmlns:space="http://www.sbml.org/extensions/space/version1">
  <model id="example">
    <listOfSpeciesTypes>
      <speciesType id="A">
        <space:mesh>
          ...
        </space:mesh>
      </speciesType>
      ...
    </listOfSpeciesTypes>
  </model>
</sbml>
```

What will be the core?

- ⊙ Expected to be *mostly* Level 2 Version 3, but ...
 - Chicken-and-egg problem: don't know if will need crucial features in the core to support extensions until we know what the extensions are
- ⊙ Current plan:
 1. Develop draft proposals for major extensions on the table
 2. Examine them together
 3. *Then* decide what needs to go into SBML Level 3 Core

What are expected Level 3 extensions?

- ⊙ Diagrams (layout + rendering)
 - Layout proposal is quite mature
- ⊙ Multicomponent/multistate species (= rule-based models or not?)
- ⊙ File inclusion
- ⊙ Model composition
- ⊙ Spatial geometry
- ⊙ Arrays
- ⊙ Dynamic structures