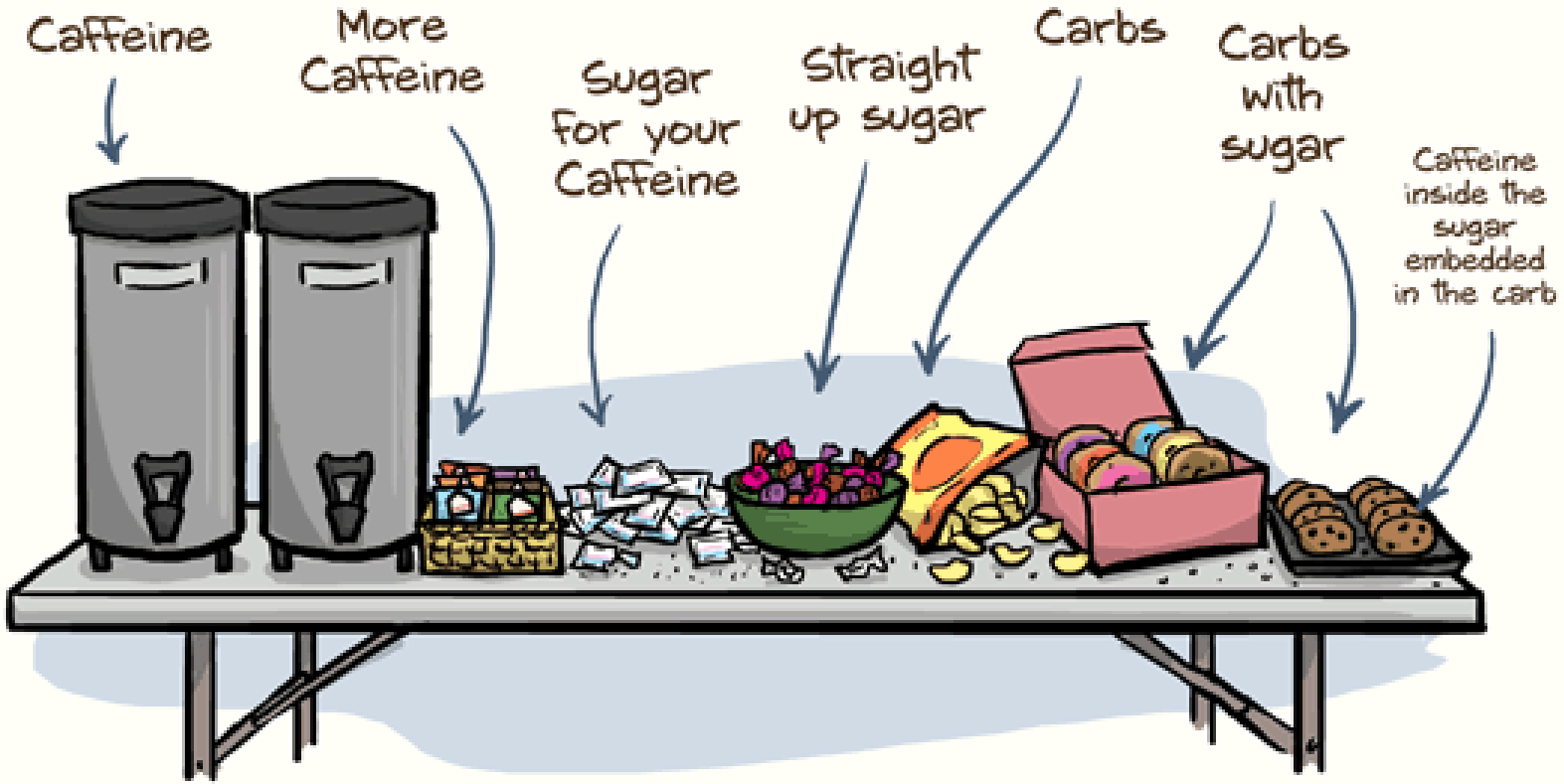
SBML L3v2 proposed changes

Lucian Smith

SEMINAR REFRESHMENTS!



Nothing says *"We are confident this seminar will be intellectually stimulating for you"* like a table full of things to help you stay awake.

All L3v1 errata

- http://sbml.org/Documents/Specifications/SBML_Level_3/Version_1/Core/Confirmed_issues_in_the_Level_3_Version_1_Core_Specification
- Typos, unclear sections, etc.
- Moral: get your corrections in!

No more required children

- ListOf's may be empty
- Reactions may have no reactants or products
- All <math> children are optional
- The <model> is optional
- The event trigger is optional
- The <lambda> in functionDefinition <math> is optional (*maybe)

No more required children

- Justifications:
  - A package may define what is missing (i.e. distrib with function definition math)
  - The modeler may not have that information, but still want to add annotations (i.e. kinetic laws)

Add an ‘id’ to things that lack it

- InitialAssignment
- Rule
- Trigger
- Priority
- Delay
- EventAssignment
- Constraint

Add an ‘id’ to things that lack it

- Justification:
  - Many packages have different reasons to refer to elements, such as comp (for deletions/replacements) and layout (for display).

Allow some SIdRefs to reference package IDs

- symbol/variable of:
  - AssignmentRule
  - RateRule
  - InitialAssignment
  - EventAssignment
- Any <ci> element
- Referent must have mathematical meaning
- Package not understood = rule not understood

Allow some SIdRefs to reference package IDs

- *NOT:*
  - ‘compartment’ of Species
  - ‘compartment’ of Reaction
  - ‘species’ of SimpleSpeciesReference
  - ‘conversionFactor’ of Model
  - ‘conversionFactor’ of Species

Allow some SIdRefs to reference package IDs

- Justification: much easier to change package variables if one can reference them directly.
- Still defines ‘fallback’ method for when package is not understood (‘just ignore it’, in most cases).

New mathML elements

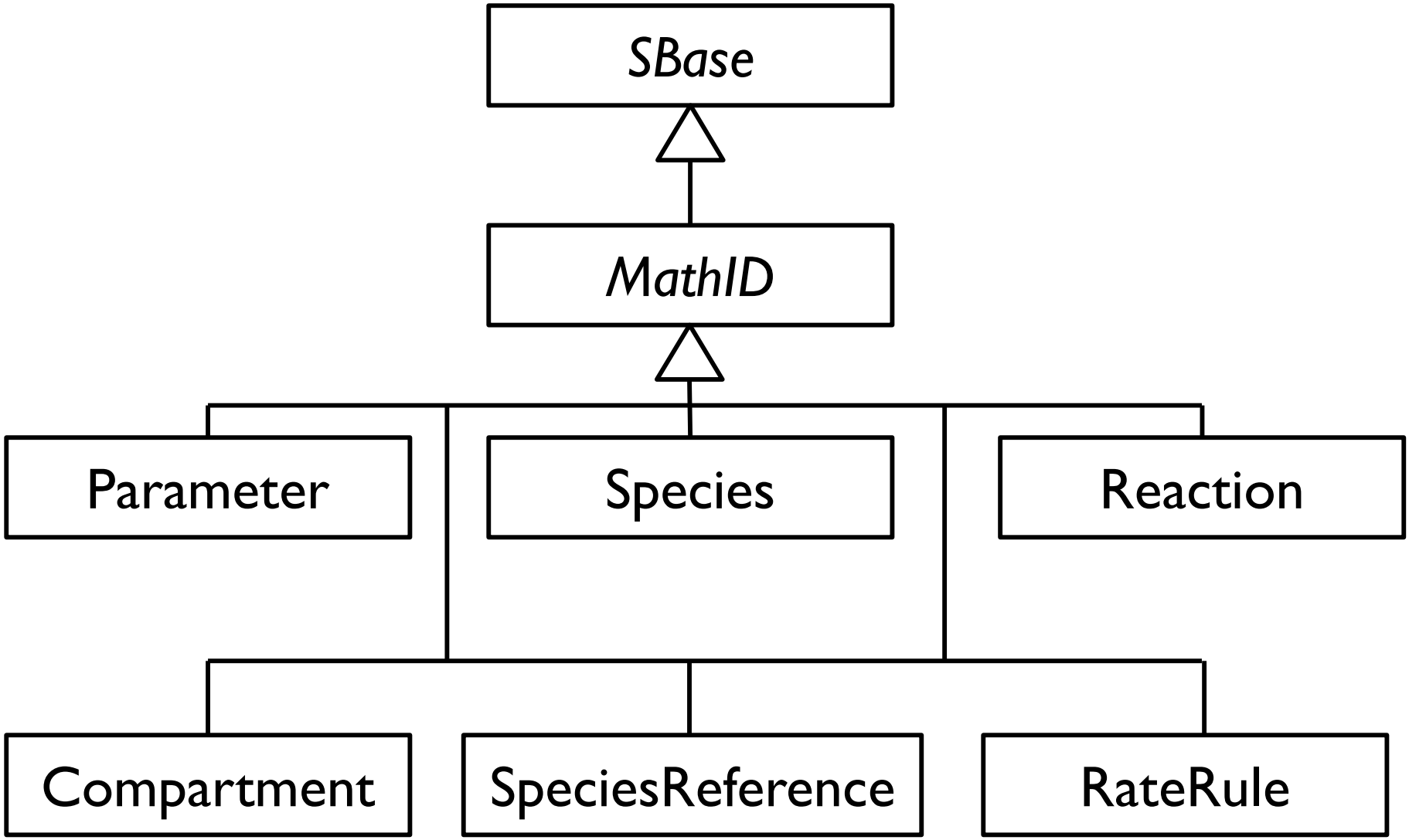
- Long discussion at http://sbml.org/Forums/index.php?t=msg&th=2154&rid=0#msg_7991
- No clear consensus, but propose:
  - Add specific MathML constructs to L3v2
  - Add more constructs with arrays/sets package
  - Add everything with an 'all MathML' package if necessary.
- Constructs to be added include:
  - <rem>, <quotient>, maybe others.

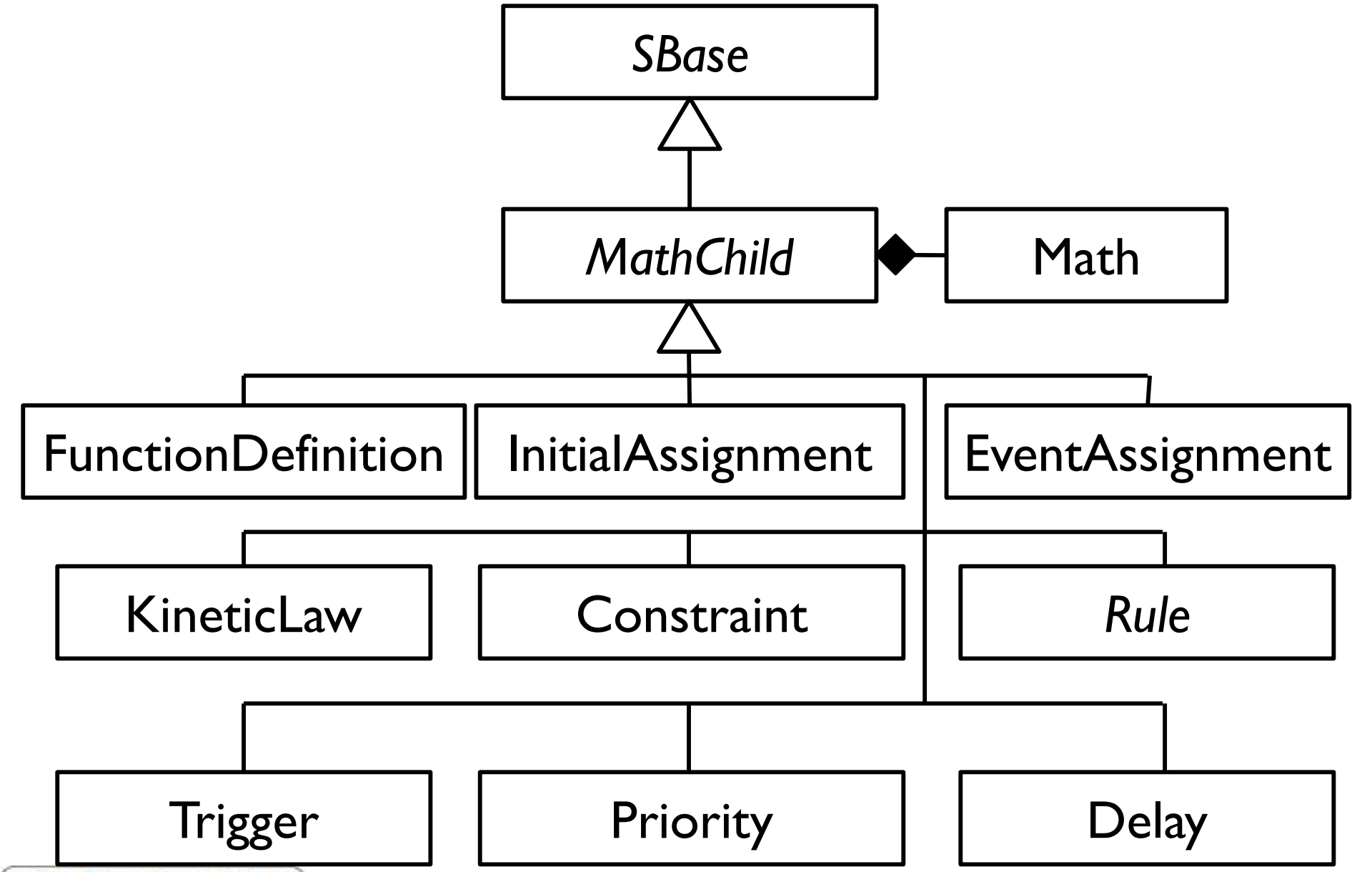
Allow Rate Rule IDs in MathML

- A RateRule ID would be equivalent to writing ‘dX/dt’ in an equation.
- Justification:
  - Many models use rates of change as independent variables in some equations.
  - A form of ‘dX/dt’ (often extended function definitions) already in use in tools such as Copasi.

Introduce intermediate class for math elements

- Introduce a 'MathID' class for elements whose ID has mathematical meaning (i.e. Parameters, Species, Reactions, etc.)
- Introduce a 'MathChild' class for elements with a <math> child.
- Justification: This will make it easier to define SIdRefs that must reference elements with mathematical meaning as packages introduce them; similarly with new package elements with <math> children.


SBase
MathID
Parameter
Species
Reaction
Compartment
SpeciesReference
RateRule


SBase
MathChild
Math
FunctionDefinition
InitialAssignment
EventAssignment
KineticLaw
Constraint
Rule
Trigger
Priority
Delay

Deprecate ‘fast’ flag?

- Some support for this in simulators.
- Actually used by modelers?
- Need feedback from community (you!)

Negative stoichiometries?

- Current spec says stoichiometry attribute 'should' be non-zero and positive, but no validation warning.
- Options:
  - Remove this from the main text
  - Add a validation warning
  - Discuss in 'best practices'.
  - Opinions?

More details at
http://sbml.org/Events/SBML_Editors%27_Meetings/During_HARMONY_2012_05_20/Summary

Current tracker items under discussion: click the 'L3v1 open discussions' search at
https://sourceforge.net/p/sbml/sbml-specifications/