Some thoughts about the future of SBML

Nicolas Le Novère, EMBL-EBI

Disclaimer

- The following questions are not a tentative to undermine the development of support for SBML Level 3. SBML Level 3 is the latest official specification of SBML. SBML is the official community standard for encoding models in (systems) biology. SBML Level 3 will remain so for years to come.
- The following questions are meant to open discussions for the long term evolution of SBML.
- **IMPLEMENT SUPPORT FOR SBML LEVEL 3**
- **DEVELOP PACKAGES FOR SBML LEVEL 3**

IMPLEMENT SUPPORT FOR SBML LEVEL 3!

Where is SBML coming from?

- NATO MCA workshop "Biotechnological and Medical Implications of Metabolic Control Analysis", Visegrad, April 1999. A number of modelers recognise that they should agree on some sort of standard format in which they could interchange metabolic models between the different simulation and analysis packages.
- 9 September 1999: Herbert Sauro announces on the usenet forum "bionet.metabolic-reg" the Portable Metabolic Binary Standard (pmb files), developed by the MMFF List Committee. This format is not an XML format. In addition to HS the MMFF group comprises Pedro Mendes.
- 18 September 1999: Igor Goryanin suggests to use XML instead of a binary format to describe metabolic models.
- Early 2000: Mike Hucka, who had worked on an XML format for neuroscience (future NeuroML) talks with Hamid Bolouri about an XML format for models in Systems Biology
- May 2000: XML standard for cellular models (MML). ERATO. 10 May 2000. Herbert Sauro presents the first release of an XML format to encode biochemical models, MML. It contains the principal features of SBML, and in particular the various lists, the kineticLaw etc.
- August 2000: Hucka M, Sauro H, Finney A, Bolouri H. An XML-based Model Description Language for Systems Biology Simulations. ERATO Kitano Systems Biology Project. Control and Dynamical Systems. 107-81 California Institute of Technology, Pasadena, CA 91125. August 8 2000. The first traceable version of a draft specification for the language called SBML.

Where is SBML coming from?

- NATO MCA workshop "Biotechnological and Medical Implications of Metabolic Control Analysis", Visegrad, April 1999. A number of modelers recognise that they should agree on some sort of standard format in which they could interchange metabolic models between the different simulation and analysis packages.
- 9 September 1999: Herbert Sauro announces on the usenet forum "bionet.metabolic-reg" the Portable Metabolic Binary Standard (pmb files), developed by the MMFF List Committee. This format is not an XML format. In addition to HS the MMFF group comprises Pedro Mendes.
- 18 September 1999: Igor Goryanin suggests to use XML instead of a binary format to describe metabolic models.

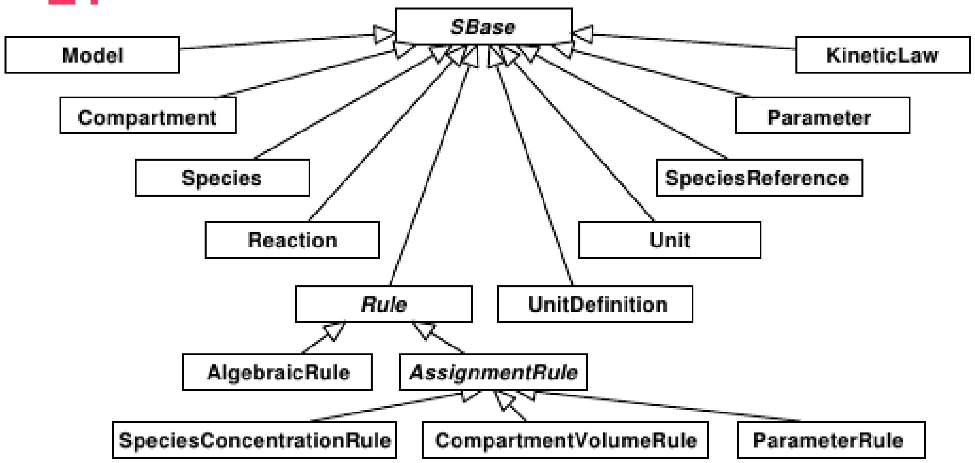
XML 1.0 became a W3C Recommendation on February 10, 1998!

NeuroML) talks with Hamid Bolouri about an XML format for models in Systems Biology

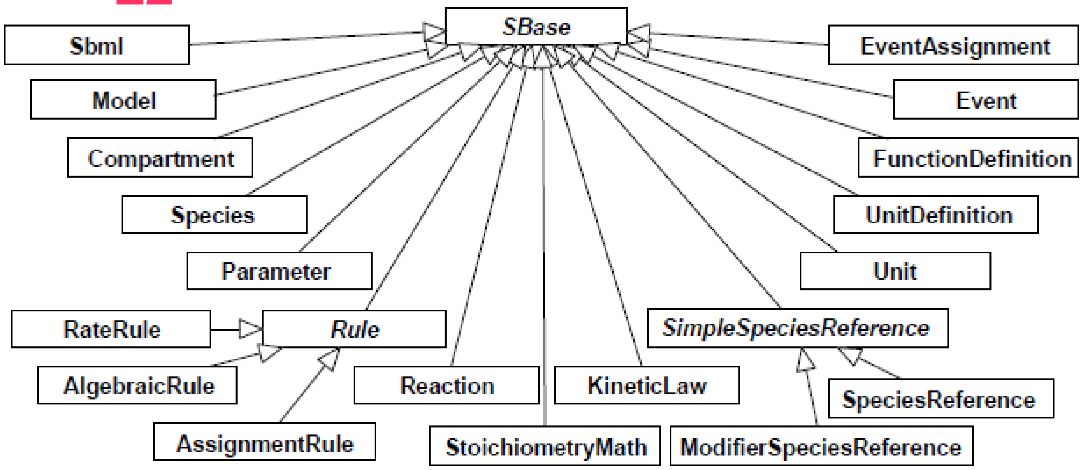
- May 2000: XML standard for cellular models (MML). ERATO. 10 May 2000. Herbert Sauro presents the first release of an XML format to encode biochemical models, MML. It contains the principal features of SBML, and in particular the various lists, the kineticLaw etc.
- **August 2000**: Hucka M, Sauro H, Finney A, Bolouri H. An XML-based Model Description Language for Systems Biology Simulations. ERATO Kitano Systems Biology Project. Control and Dynamical Systems. 107-81 California Institute of Technology, Pasadena, CA 91125. August 8 2000. The first traceable version of a draft specification for the language called SBML.

Similarities between SBML L1, L2 and L3

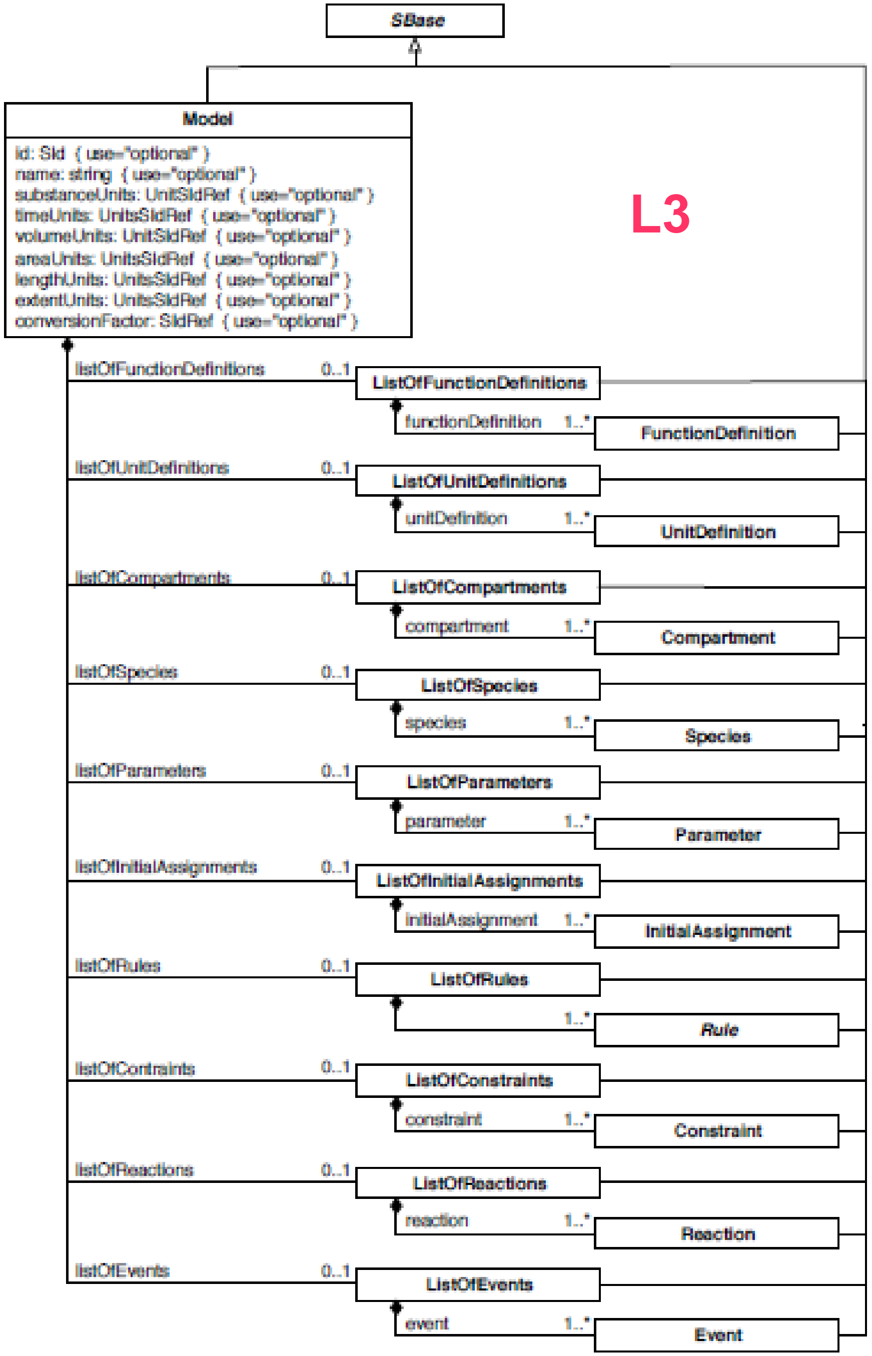
L1



L2



L3



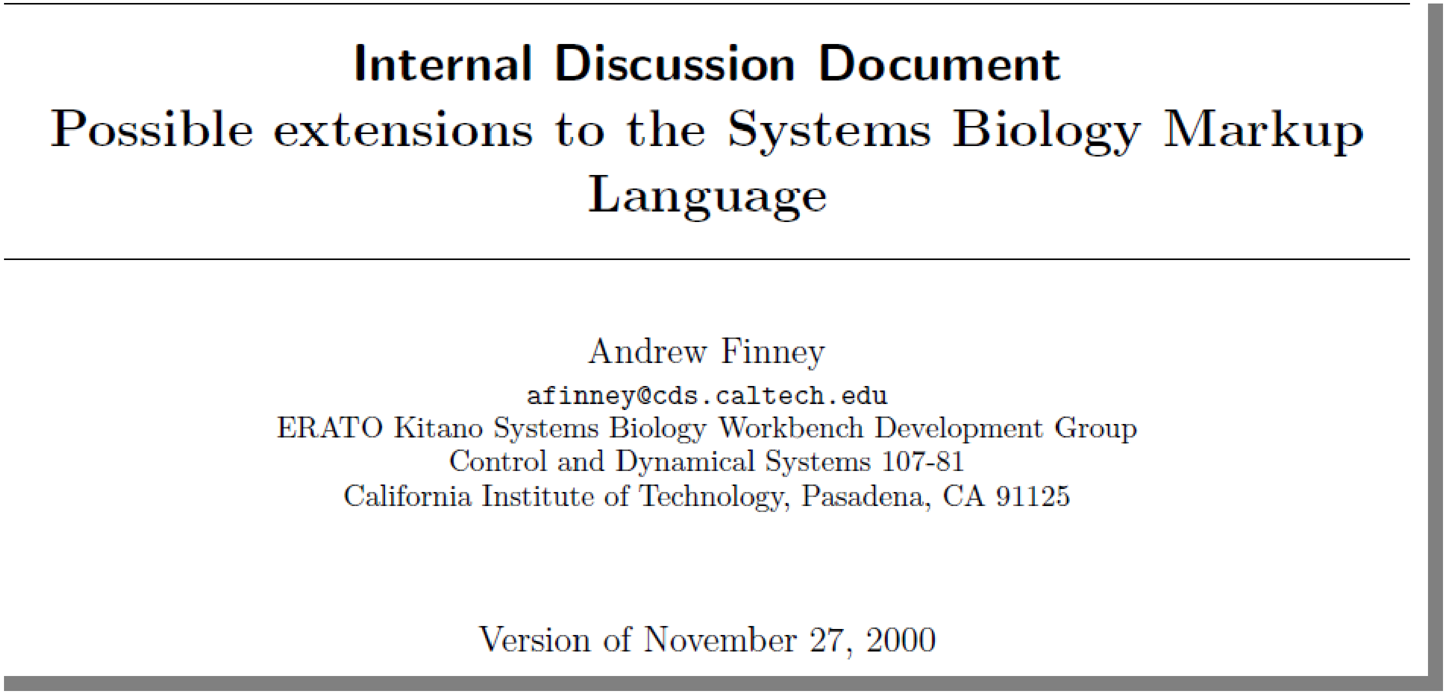
Differences between SBML L1, L2 and L3

Level 1 (devpt 2000-2003)	**Level 2** (devpt 2002-2008)	**Level 3** (devpt 2009-)
predefined functions	function definitions	function definitions
proprietary infix math notation	all math in MathML	all math in MathML
reserved namespaces for annotation	no reserved namespaces for annotations	no reserved namespaces for annotations
no controlled annotation	controlled RDF annotation	controlled RDF annotation
no discrete events	discrete events	discrete events
monolithic	monolithic	modular
default values	default values	no default values

Progressive simplification, generalisation and externalisation

~15 software	~135 software	>220 software

Where is SBML Level 3 coming from?



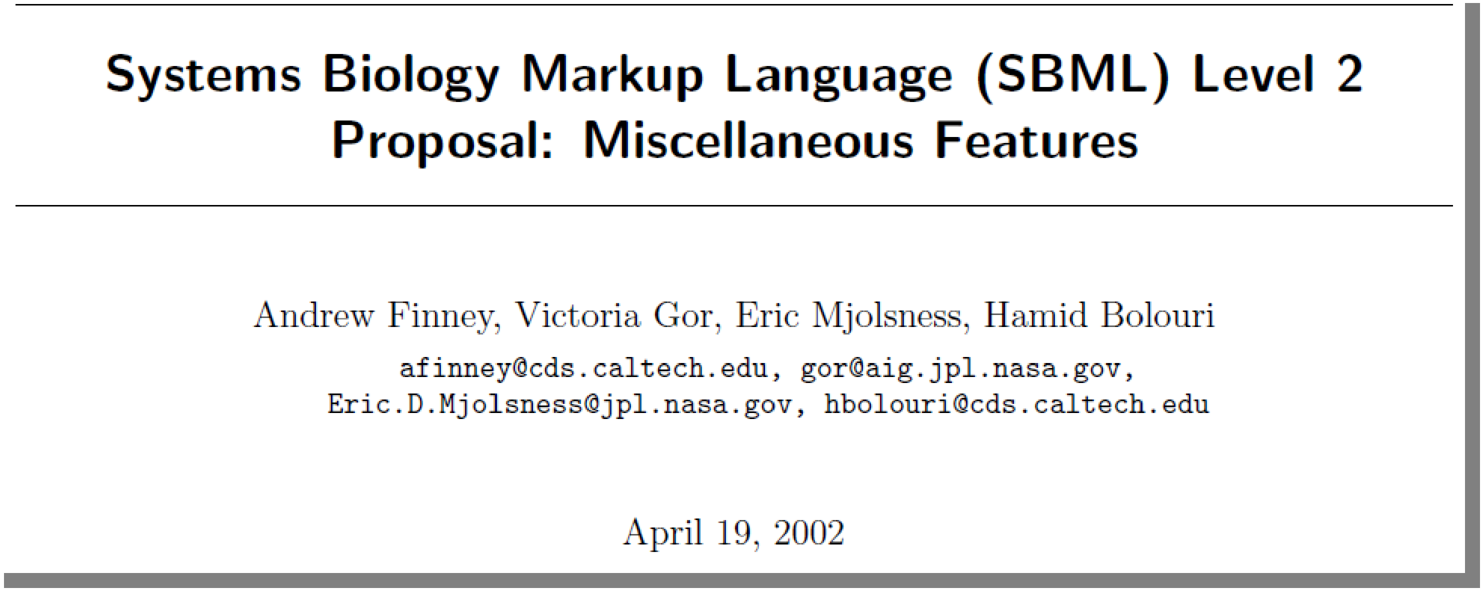
Internal Discussion Document
Possible extensions to the Systems Biology Markup Language

Andrew Finney
afinney@cds.caltech.edu
ERATO Kitano Systems Biology Workbench Development Group
Control and Dynamical Systems 107-81
California Institute of Technology, Pasadena, CA 91125

Version of November 27, 2000

Mentioned most of the packages, but also controlled annotations!

Proposed the mechanism of packages.



Systems Biology Markup Language (SBML) Level 2 Proposal: Miscellaneous Features

Andrew Finney, Victoria Gor, Eric Mjolsness, Hamid Bolouri
afinney@cds.caltech.edu, gor@aig.jpl.nasa.gov,
Eric.D.Mjolsness@jpl.nasa.gov, hbolouri@cds.caltech.edu

April 19, 2002

What does that tell us for the future?

- SBML Level 3 core is mostly a cleaned and glorified version of SBML Level 1 (this is a compliment!)
- Most of SBML structure has been designed while we had only a couple of years experience in SBML.

→ Would we design it the same way if starting today?
→ Would we design it the same way in 2020?

What does that tell us for the future?

- SBML Level 3 core is mostly a cleaned and glorified version of SBML Level 1 (this is a compliment!)
- Most of SBML structure has been designed while we had only a couple of years experience in SBML.

  → Would we design it the same way if starting today?
  → Would we design it the same way in 2020?

- First proposal of package-based structure: April 2002
- First official release of SBML Level 3 core: October 2010

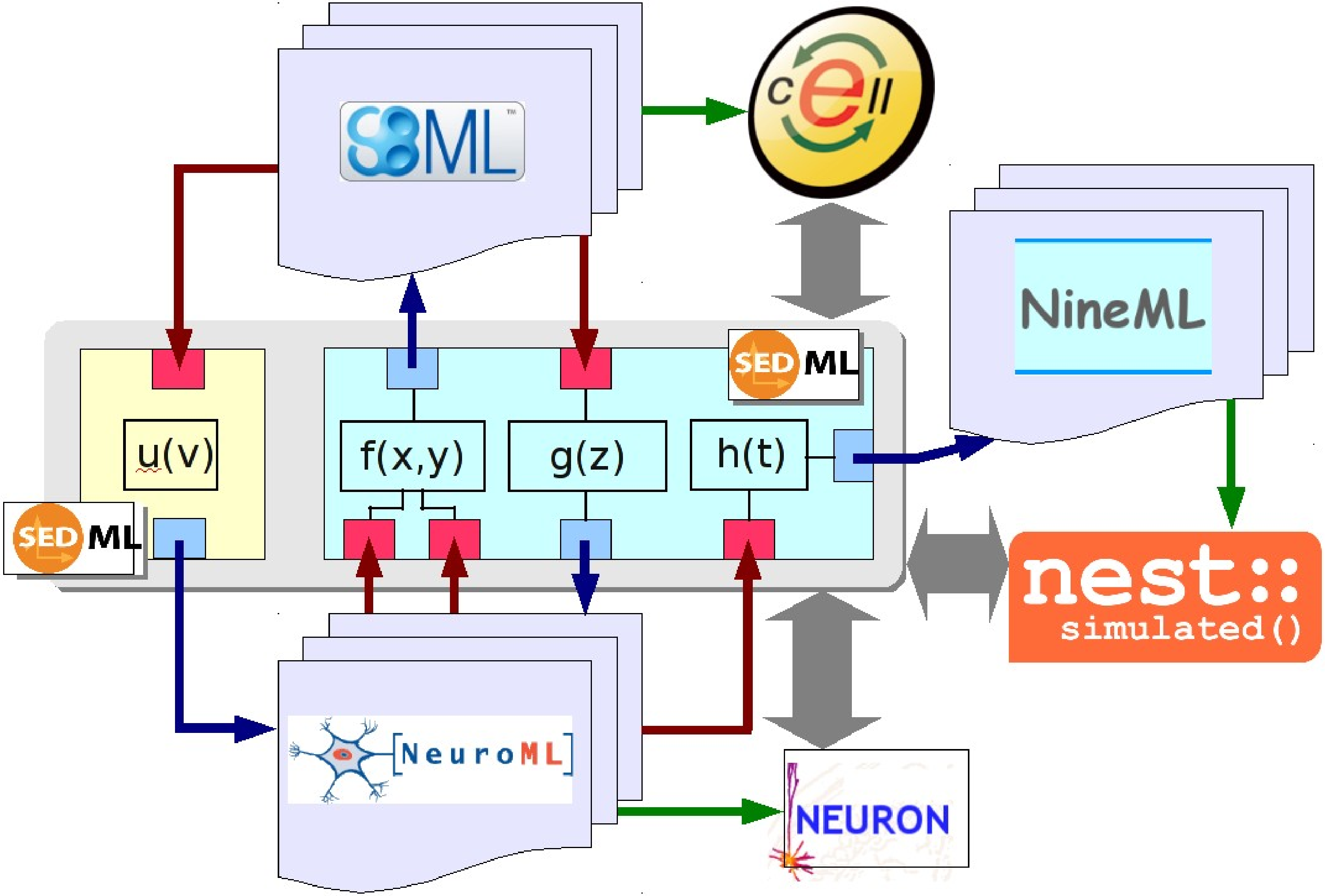
  → We need a few years head start if we want a new structure in 2020

One model, one SBML file, one simulation

- One of the main initial aims of SBML was to exchange information between modules of the Systems Biology Workbench (which was the actual project of the ERATO Kitano grant)
- Since then, the paradigm has always been that a software loads a model encoded in SBML and subsequently does something interesting with this model only.
- The ensemble of rules, reactions and events are meant to lead to a system of mathematical equations solved together.
- This is somehow mirrored in MIRIAM rule of instantiation
- Physiology, neurosciences, crop and ecology modeling require the use of different “models” at different scales, analysed using different approaches, synchronised by methods (mathematics) described outside of the elementary “models”
- NB. The 1-1-1 rule is also one of the reasons behind the abuse of events to represent sequential simulations (see Jonathan Cooper's talk)

→ Now that SED-ML is here, allowing to run a simulation experiment using several model description, we maybe need to revisit this paradigm

Example of multi-model simulation experiment



Is SBML about Process Description?

SBML directly inherits from the description of metabolic networks. Its paradigm is chemical kinetics, with processes consuming pools and producing pools. The mathematical system, whether ODEs or set of propensities, is largely generated from the set of reactions. So is SBML mainly about model represented as a set of processes?

→ **Yes**: Why do-we try to stretch it to cover rule-based models, logical models, statistical models, etc.?

→ **No**: Why are the elements necessary to the description of processes in the core?

One language or a federation of languages?

- Some model approaches are not meant to be used together, in a given system of equations. They are meant to be used on their own, and sometimes connected through the result of their simulations and analyses
- For instance the `multi` package is very complex, because the rules are really meant to be interpreted to generate processes acting on pools, and not mixed with them
- The `quali` package is very simple, because there are no points of contact between quali and core classes
- Furthermore, the communities using those models are almost not overlapping. They model different aspects of biology, have different questions (and have different meetings, journals etc.) “They” Vs. “Us”

→ Should it not be more sensible to have different representations in different SBML sublanguages or at least in different `model` elements?

The package system would still be used for different aspects of the same approach (e.g. spatial, constraints etc. for process description)

External initial conditions

- Reminder: One of the main initial aims of SBML was to exchange information between modules of the Systems Biology Workbench (which was the actual project of the ERATO Kitano grant). Hence the need to put everything in one representation.
- But participants, such as VCell and StochSim representatives, were already reluctant to put numbers in SBML. Models versus model instances
- With other modelling activities such as pharmacometrics, tying numbers to mathematics is becoming even more problematic.
- Externalising numbers would allow to re-use the same model with different parametrisations, allow to handle numbers in an homogeneous manner throughout the model life-cycle etc.

Modularity in core (1)

- Biology is modular.
- At the moment, modellers have to edit entire SBML files.
    - Problem of super-large models (e.g. jamboree models): maintenance (editing, fixing) is very difficult, visualisation is even more difficult
    - Development of those models may require the expertise or workforce of many groups. Much easier if we can distribute the modules.
    - Problem of multi-scale models meant to be simulated with several software
- Modularity allows robust encapsulation and re-use
- Many computer representation formats are modular
- We actually considered CellML structure as a solution back in 1999/2000. Deemed too complicated at the time (It is hard to develop partial support). Alternative were proposed by other early members of the community (e.g. ProMot/Diva). But the urgent need was a simple format to encode metabolic models. Nevertheless, CellML approach may be the right one from an engineering point of view, and we may now have better XML technologies to support that.

Modularity in core (2)

- Modules would allow to have different sets of units in a given model
- No complexity increase: The current L3 core would be a default module

NB: All those issues are not answered by the `comp` package! We are not talking about a model made of several models, with complex interface generation, replacement etc. But it would facilitate the job of comp

Attributes versus Elements

- SBML does not use XML elements with content. When properties of a component are necessary, the values are stored in attributes

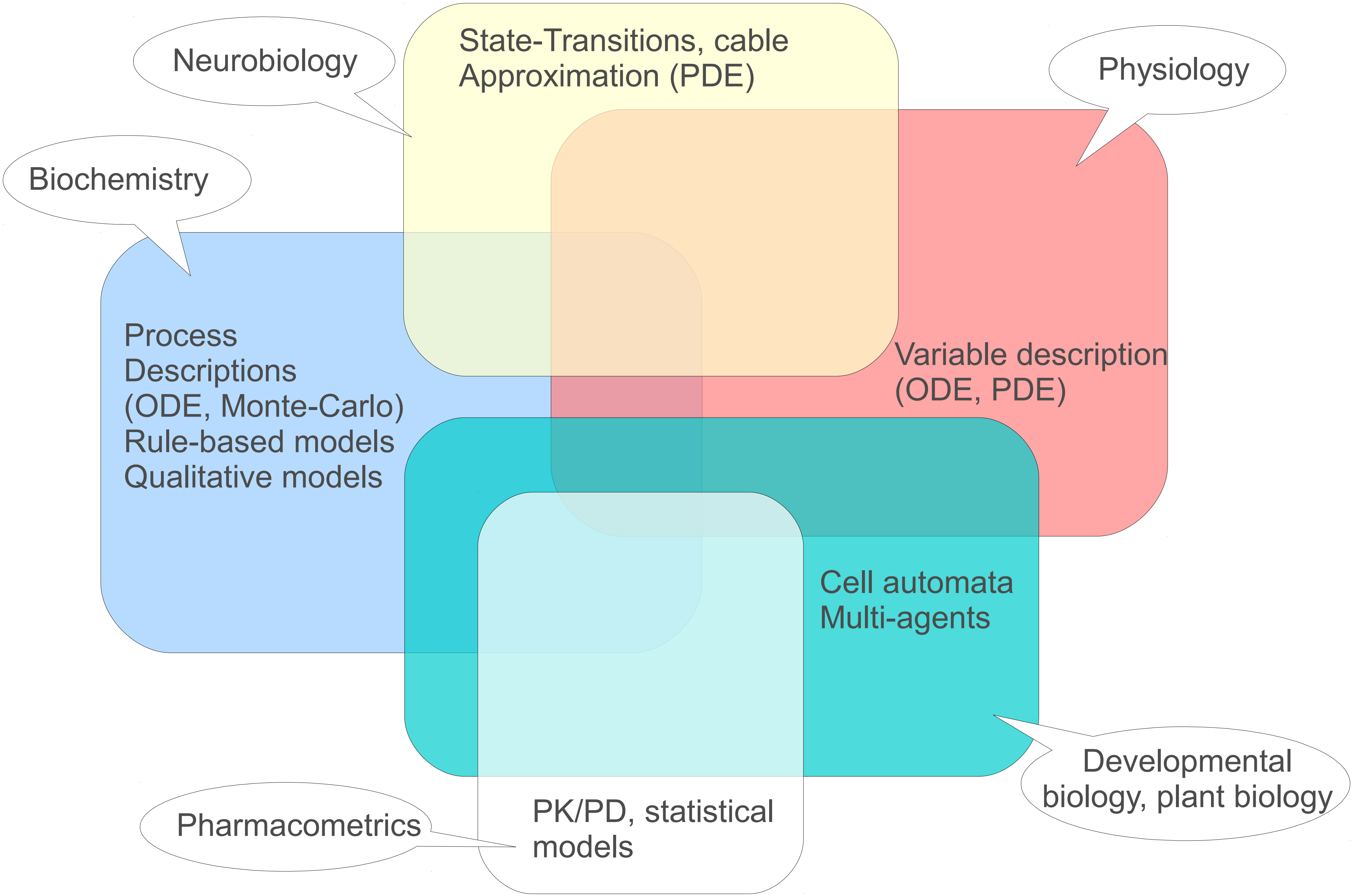
```
 <species metaid="X" id="X" compartment="comp1" initialAmount="0"
units=”myUnit”>
      <annotation>
        <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:bqmodel="http://biomodels.net/model-qualifiers/">
          <rdf:Description rdf:about="#_000003">
            <bqbiol:isDescribedBy>
              <rdf:Bag>
                <rdf:li rdf:resource="urn:miriam:pubmed:12345"/>
              </rdf:Bag>
            </bqbiol:isDescribedBy>
          </rdf:Description>
        </rdf:RDF>
      </annotation>
</species>
```

- Storing alternative values is not possible
- Relating the attributes is difficult (e.g. units ... of what?)
- Annotations cannot easily point to attributes. What is the article about? The location of the species? The initial concentration?

Miscellaneous issues

- Expansion of the maths
  - e.g. vector, matrix, sum, product
- Semantically loaded element names
  - `Reaction` (we want to represent all processes, not just "reaction"); `reactant`, `product` Not only some input are not strictly speaking reactant (transport ...), but the reaction can be negative, therefore consuming the products and producing the reactants.
  - `Species` (we want to represent all pools). Plus confusion with organism.
  - Export semantics to proper tools: ontologies
- Explicitly defined pointers
- No densities, everything expressed as amount?
- Make time an explicit variable?

Do-we want to cover all that? Will-we without "adapting to them"?



Neurobiology

State-Transitions, cable Approximation (PDE)

Physiology

Biochemistry

Process Descriptions (ODE, Monte-Carlo) Rule-based models Qualitative models

Variable description (ODE, PDE)

Cell automata Multi-agents

Developmental biology, plant biology

Pharmacometrics

PK/PD, statistical models

A note on “them” versus “us”

- The SBML community (and more largely COMBINE) is not defined *a priori* but grows out of the needs for encoding.
- SBML is a format to exchange models between tools. If we want to cover a new type of models, the developers of the tool that will exchange these models must be involved
  - They know better what to cover, and they know how they do it at the moment
  - Academic people will not use a format imposed from outside
- The more experienced members of the community must guide and help the newcomers. But one should always try to get new expertise on-board. We do NOT know better what people want or need. But we can help people realising what they want or need.