

Presentation by Mike Hucka at the
CellML SBGN SBO BioPAX MIASE workshop
5 April 2009
Waiheke, New Zealand

A Very Brief Introduction to SBML and *Events* in SBML

Michael Hucka

*Senior Research Fellow, and co-director of the
Biological Network Modeling Center (BNMC)*

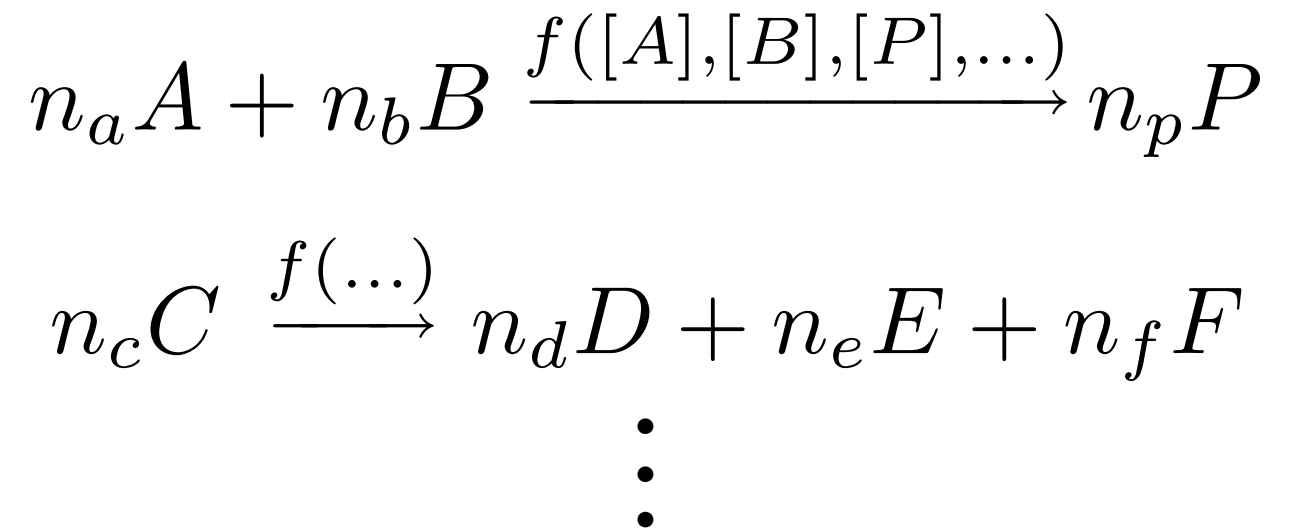
*California Institute of Technology
Pasadena, California, USA*



About SBML ...

SBML = Systems Biology Markup Language

- Machine-readable format for representing computational models
 - Can represent processes such as (but not limited to) biochemical reactions with arbitrary rate functions
- Can also include
 - Compartments (i.e., where substances are located)
 - Mathematical “extras” (e.g., additional assignments)
 - Discontinuous events with arbitrary triggers
- Neutral with respect to the framework into which the model is cast
- Aimed at being serialized in XML



SBML is widely supported

- Supported by >140 software systems

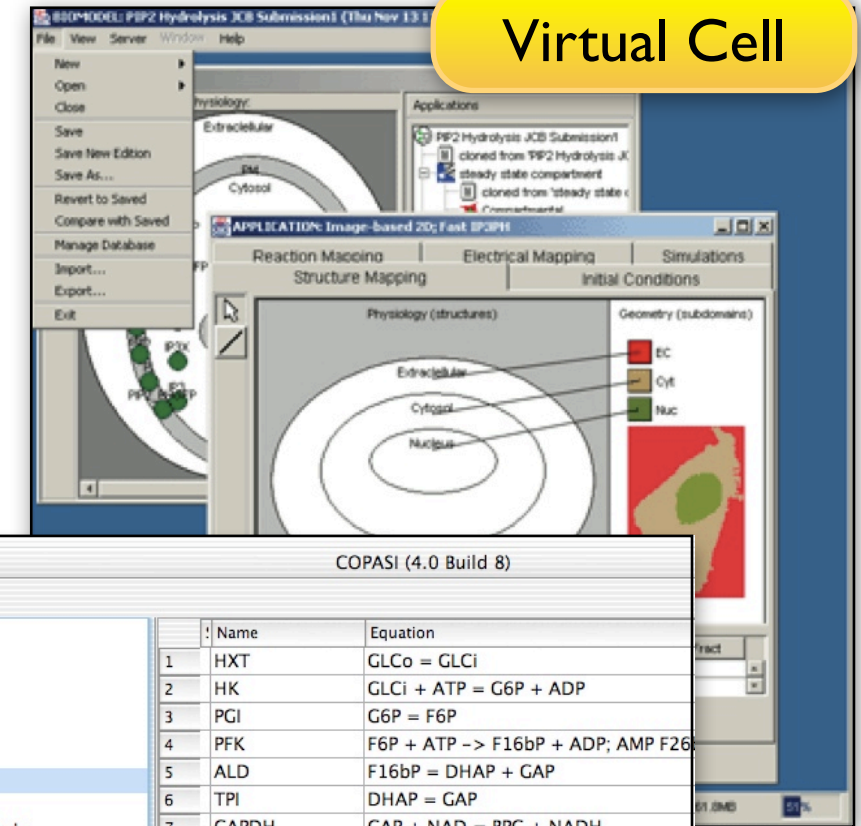
- Open-source & commercial
- General-purpose environments
 - Mathematica, MATLAB, etc.

- Special-purpose software

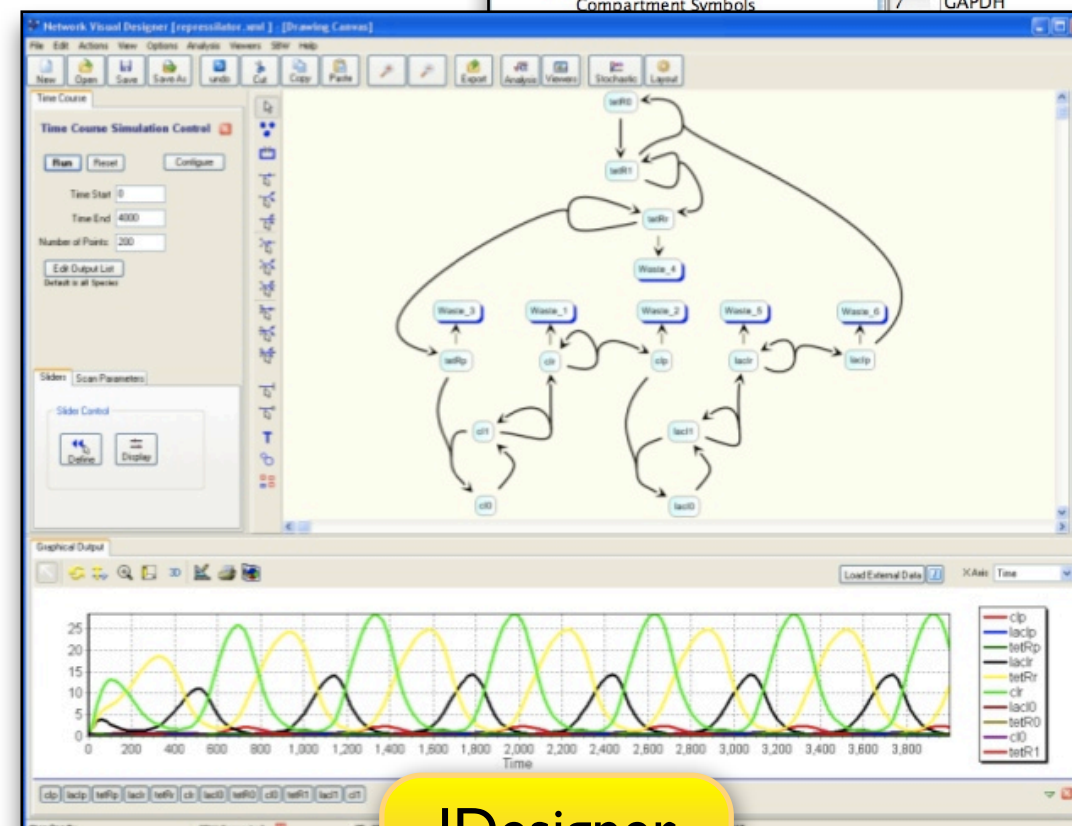
- model editing
- simulation
- analysis
- visualization

- (See sbml.org)

Virtual Cell



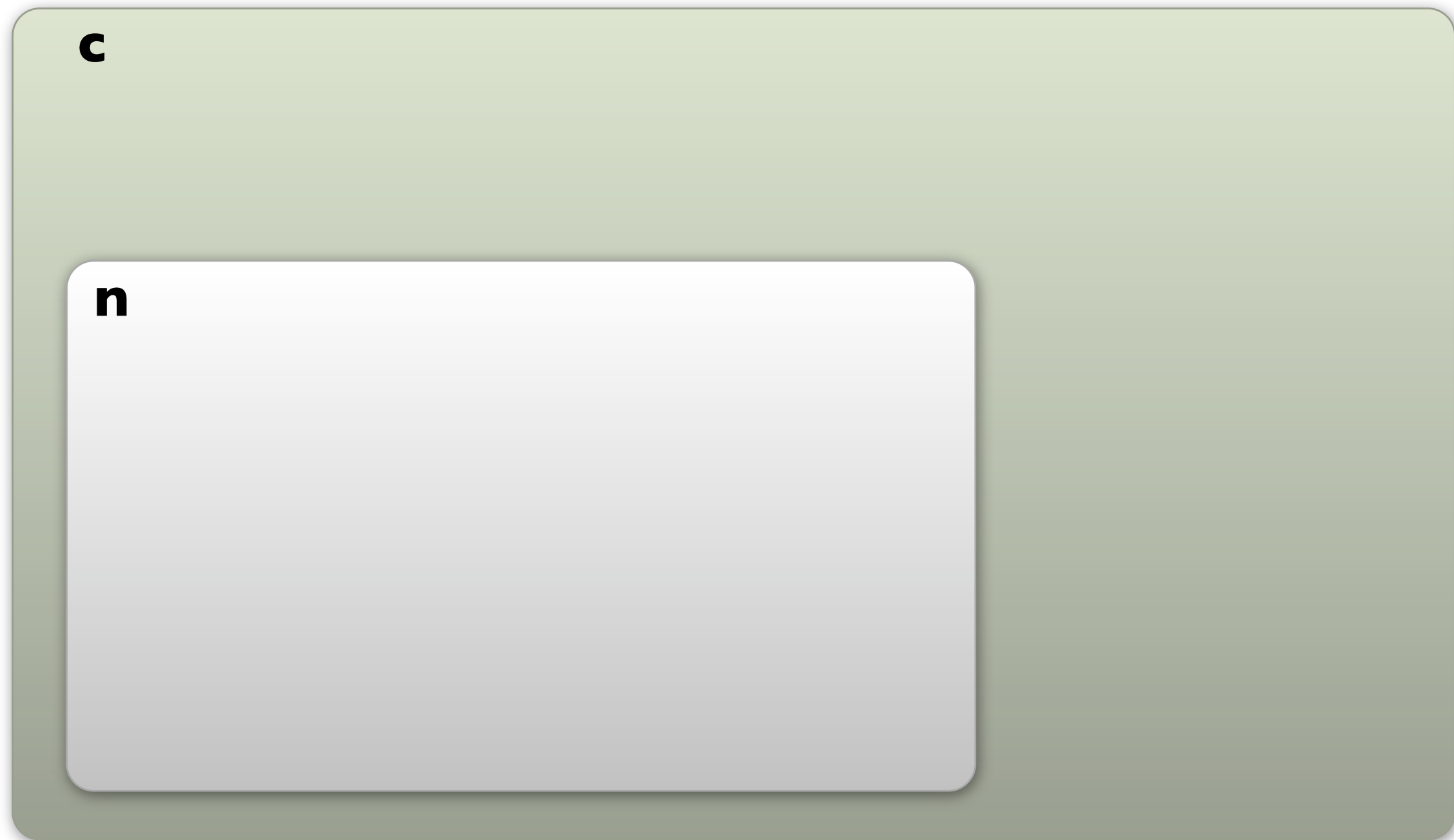
COPASI



JDesigner

Some basics of SBML model encoding

- Well-stirred compartments



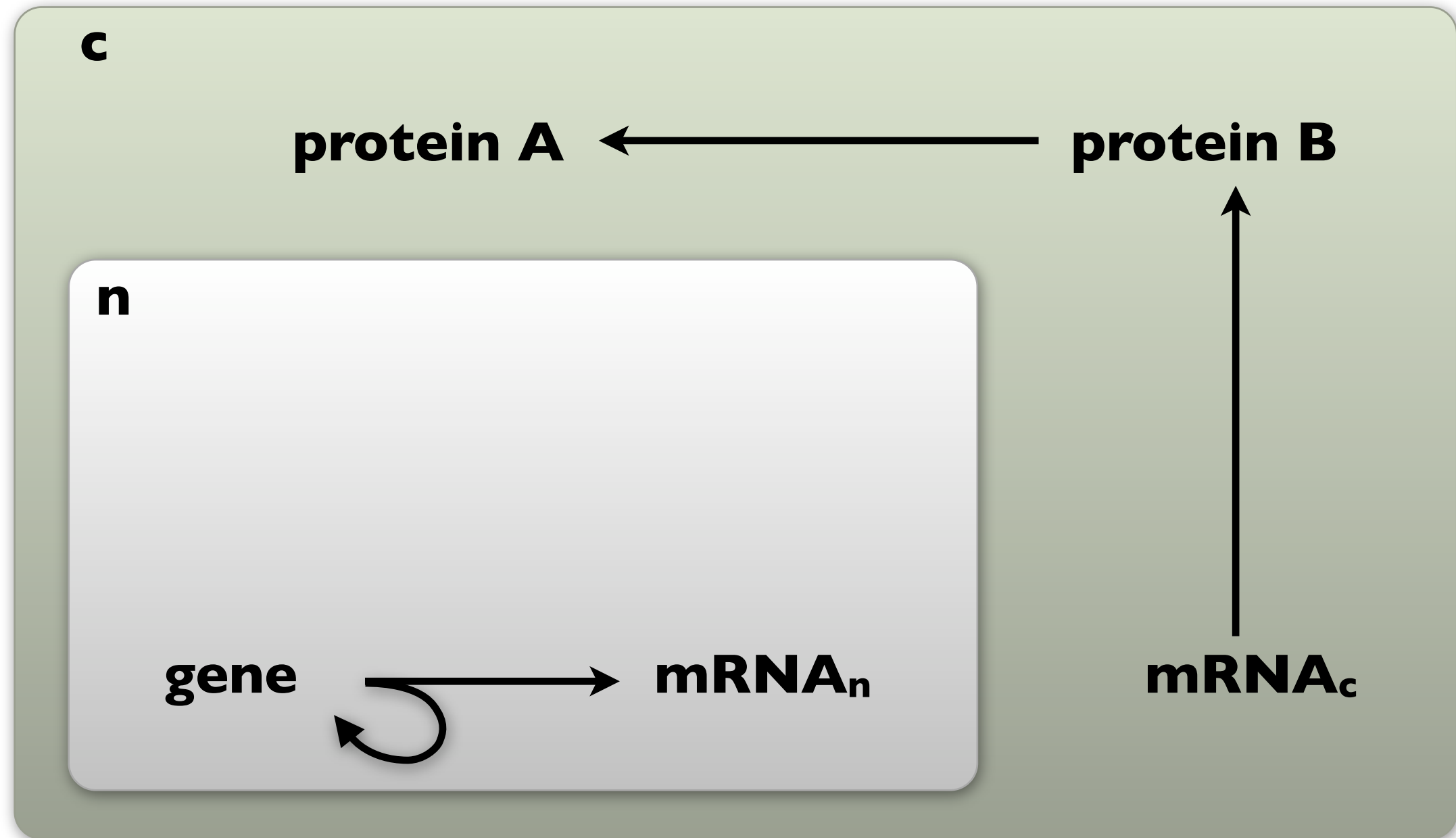
Some basics of SBML model encoding

- Species pools are located in compartments



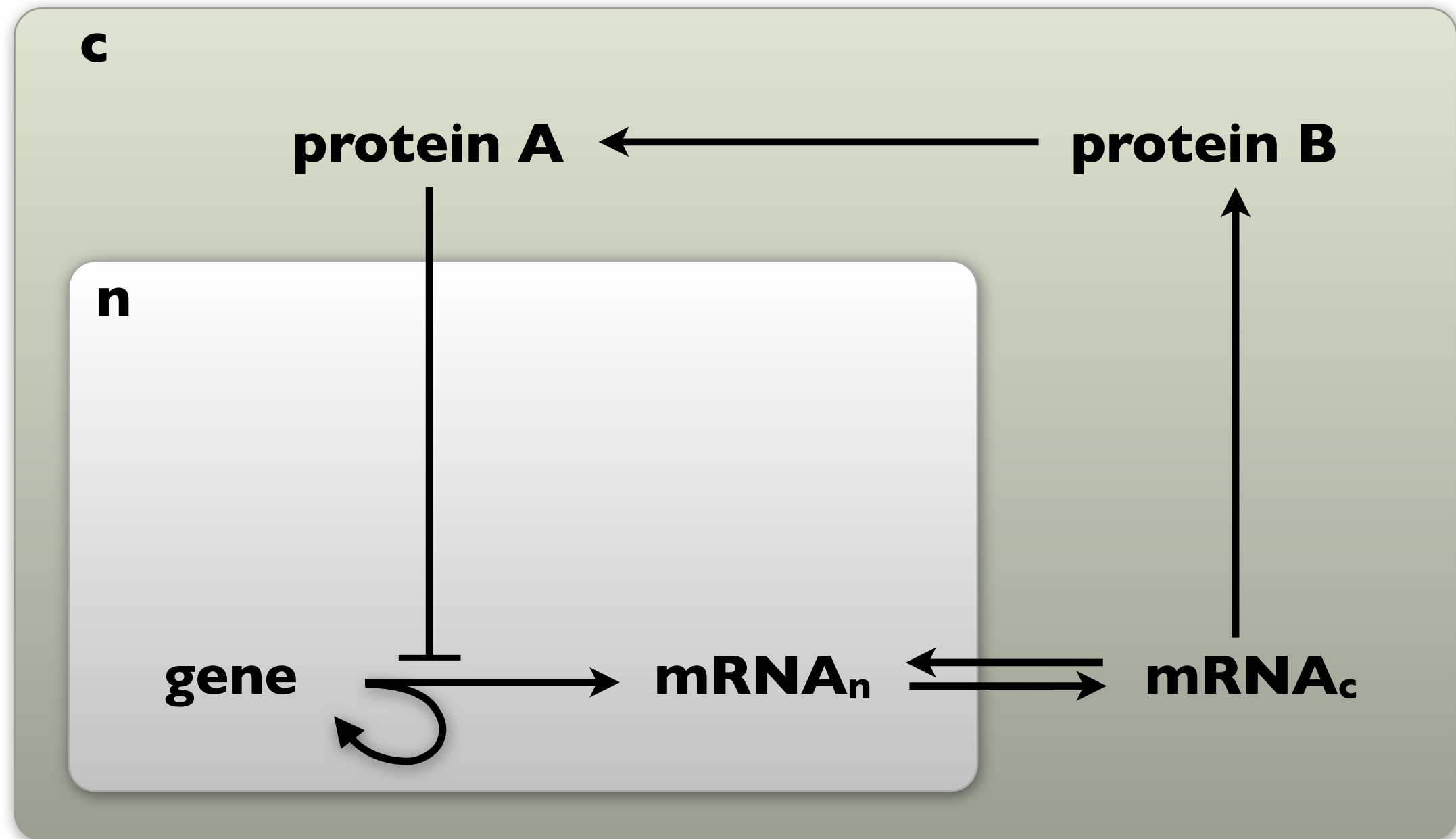
Some basics of SBML model encoding

- Reactions can involve any species anywhere



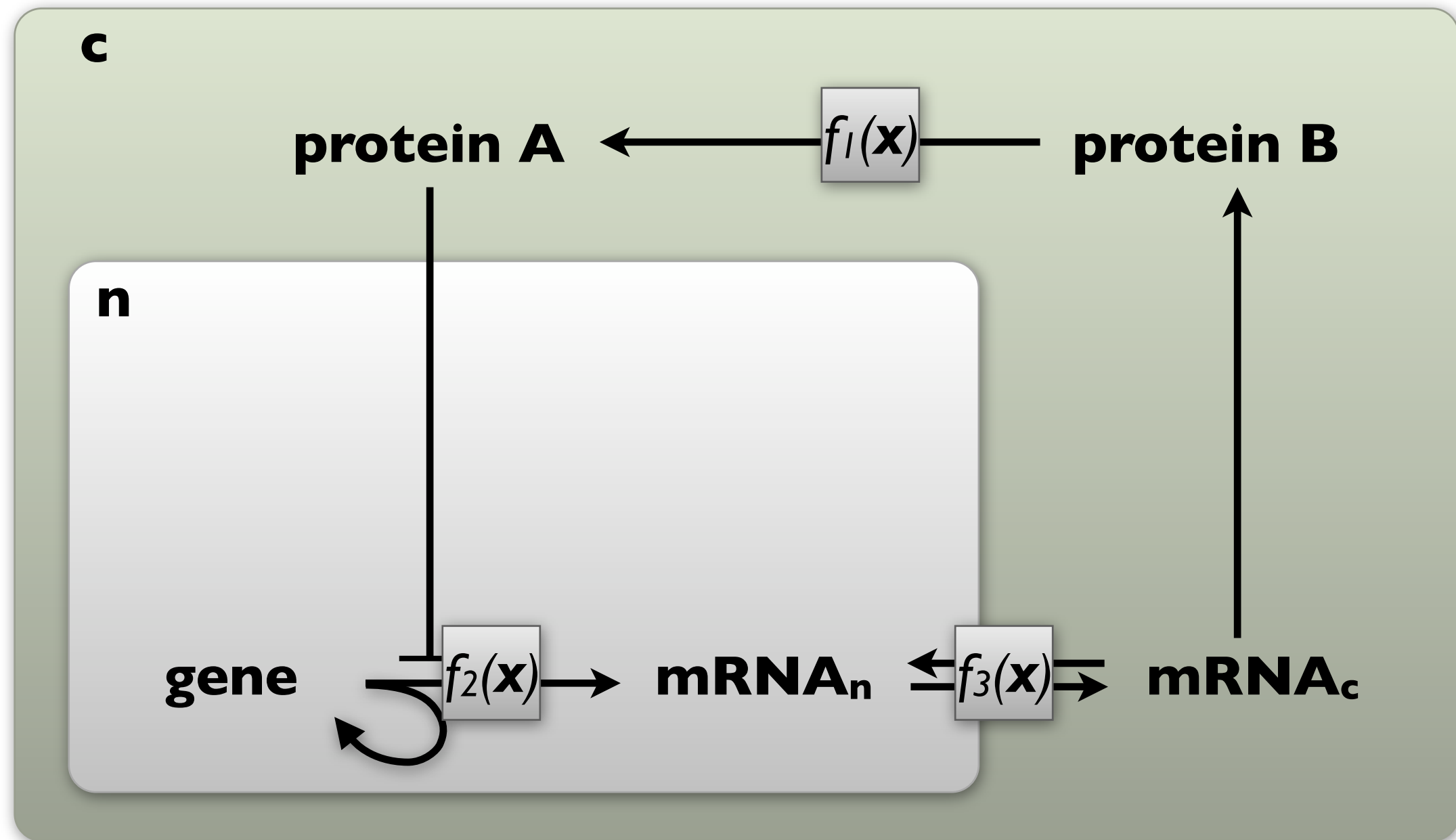
Some basics of SBML model encoding

- Reactions can cross compartment boundaries



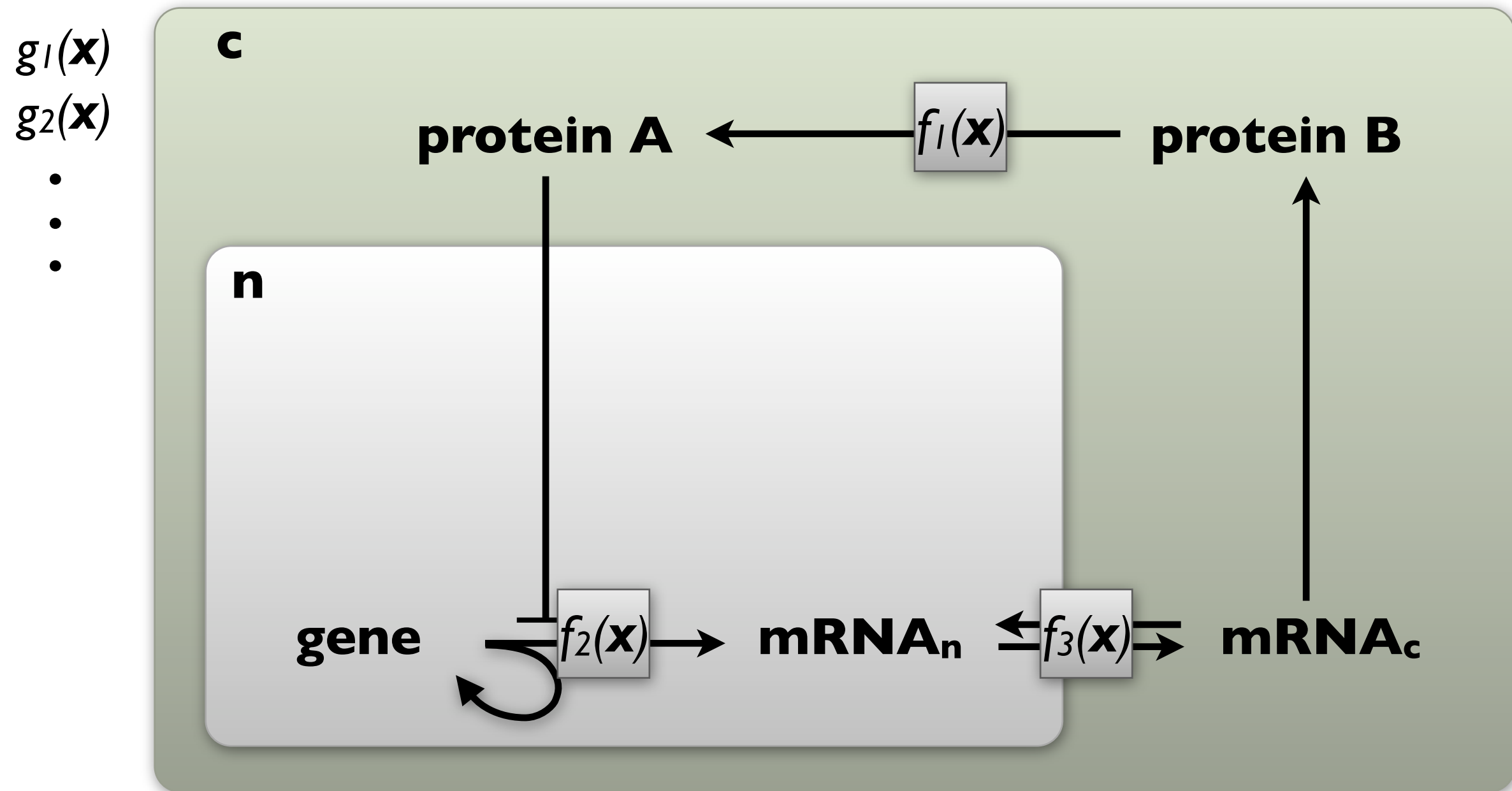
Some basics of SBML model encoding

- Reaction/process rates can be (almost) arbitrary formulas



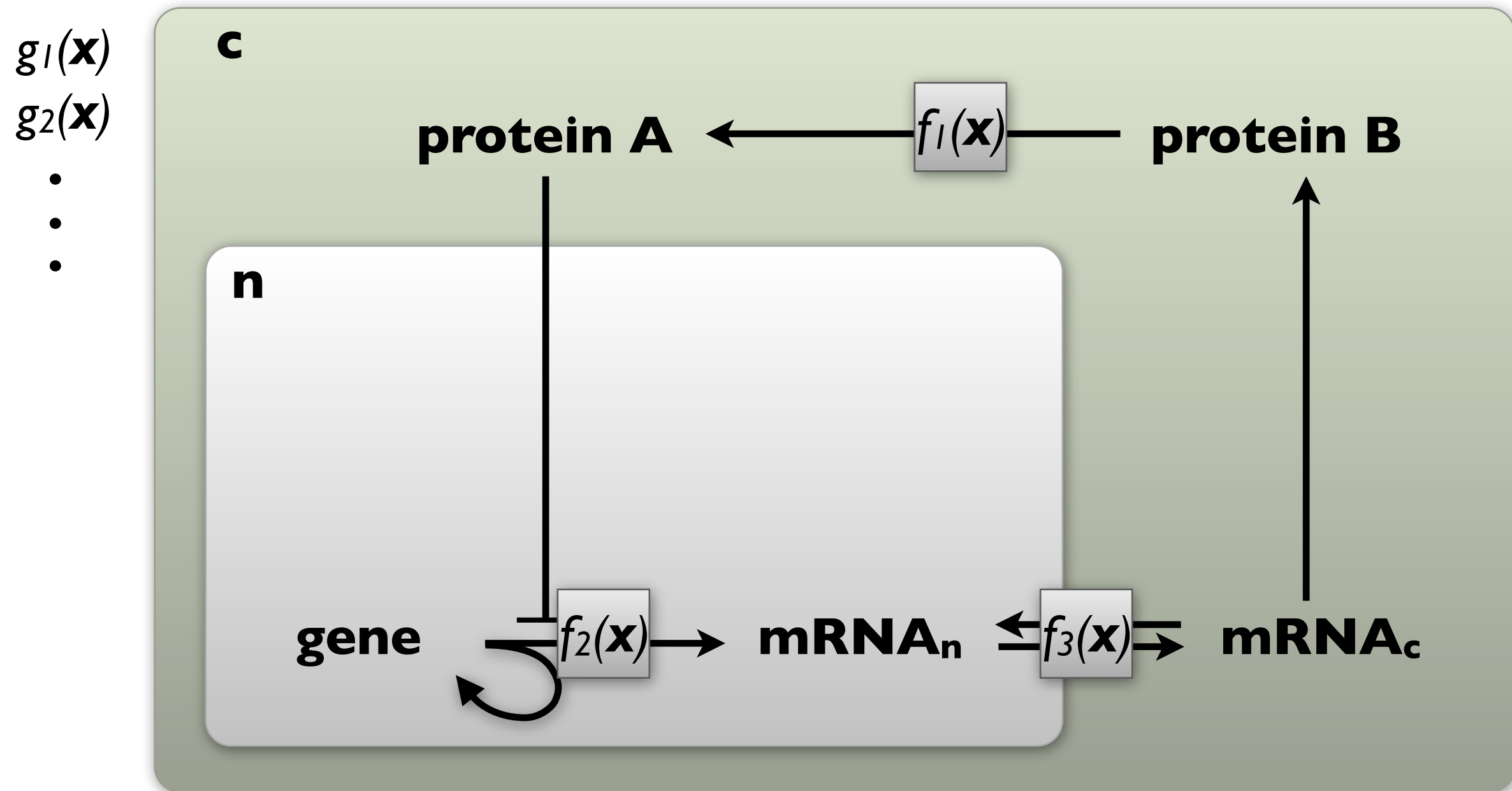
Some basics of SBML model encoding

- “Rules”: equations expressing relationships in addition to reaction sys.



Some basics of SBML model encoding

- “Events”: discontinuous actions triggered by system conditions



E_1 : when (...condition...),
do (...assignments...)

E_2 : when (...condition...),
do (...assignments...) ...

Not's

- ⦿ Declarative representation, not procedural
 - Not a simulation script
 - Something else must provide that (e.g., MIASE/SED-ML)
- ⦿ Not meant for humans to read/write
- ⦿ Not meant to be a software system's internal format
 - Software can translate to/from internal representation
- ⦿ Not for representing experimental or numerical results

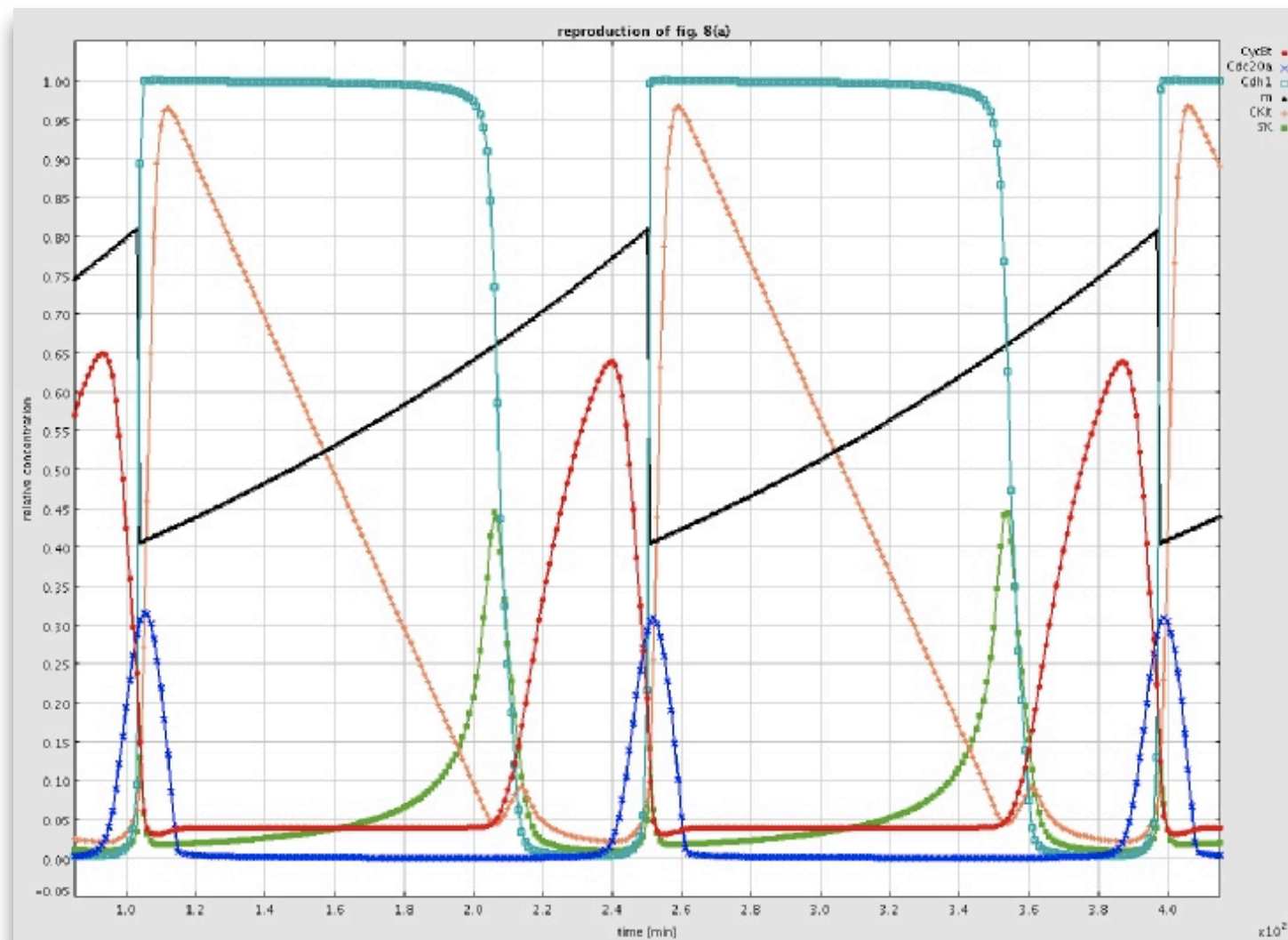
About events in SBML ...

Definition of events in SBML

- ◎ 3 main parts:
 - Trigger condition
 - MathML expression evaluating to a boolean
 - 1..* Event assignments
 - Each is a structure with 2 parts:
 - ▶ Variable to be assigned
 - ▶ MathML expression defining the value to be assigned
 - (Optional) Time delay, between firing and execution
 - MathML expression evaluating to a non-boolean

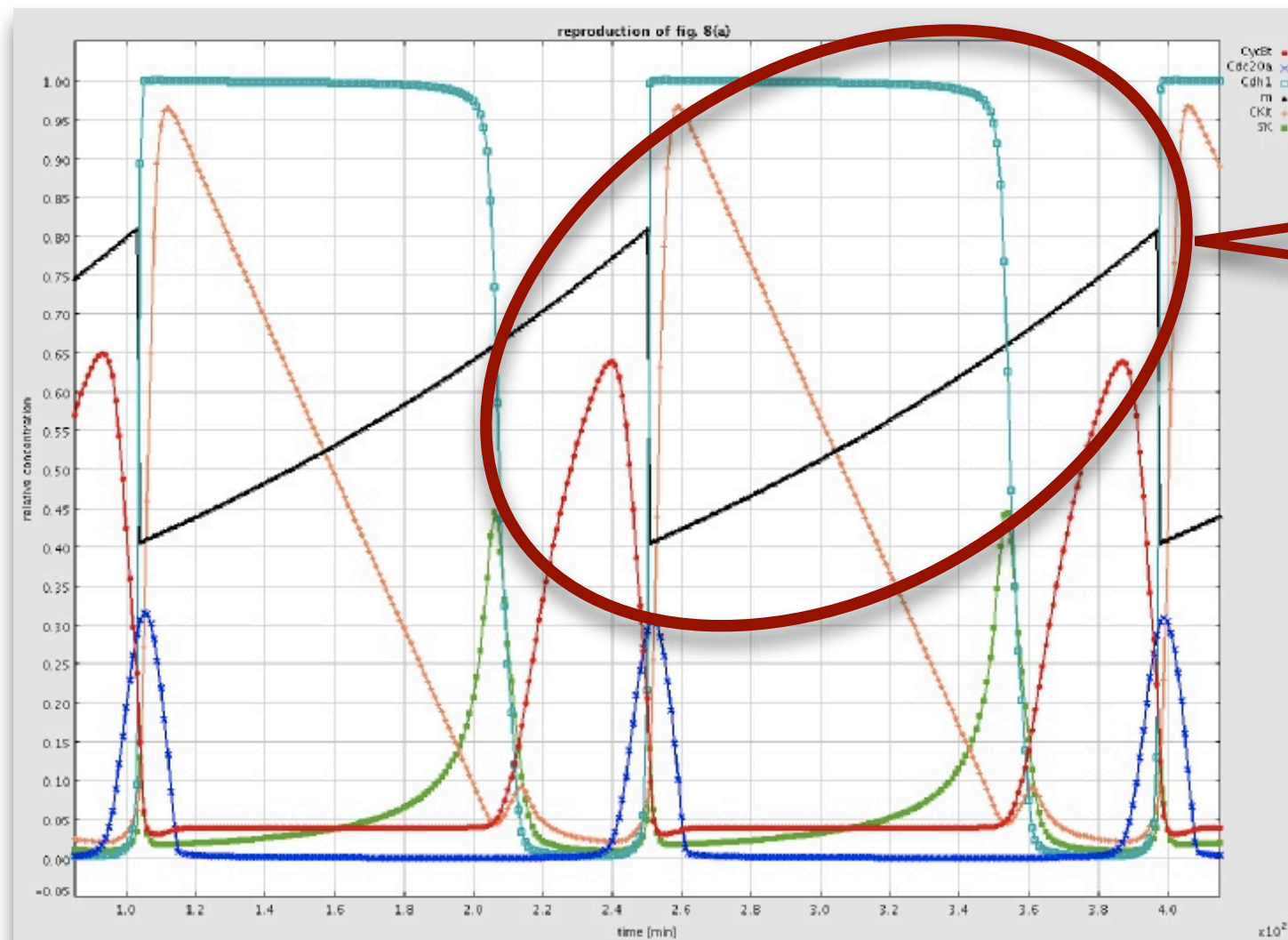
When does an event take place?

- Event “fires” when trigger expression transitions from “false” to “true”
 - Fires each time this happens (i.e., can happen multiple times)
- Example from BioModels Database:
 - Model BIOMD0000000195: *Tyson2001_Cell_Cycle_Regulation*



When does an event take place?

- Event “fires” when trigger expression transitions from “false” to “true”
 - Fires each time this happens (i.e., can happen multiple times)
- Example from BioModels Database:
 - Model BIOMD0000000195: *Tyson2001_Cell_Cycle_Regulation*



When: $CycB < 0.1$
Assign: $m = m/2$
Delay: 0

Event assignments

- ◎ Assignment occurs when the event is **executed**, not when it fires
 - Event *Delay* can separate **time-of-firing** from **time-of-execution**
- ◎ Each assignment sets the value of a species, compartment, or parameter
 - Restrictions:
 - No more than 1 assignment to the same variable in a given event
 - Object must not be flagged as “constant”
 - Object must not be subject of an SBML *Assignment Rule*
- ◎ *Limitation*: Cannot create or destroy objects
 - Coming as a separate capability in future SBML

3 more things

- ⦿ Need to specify **when** the event assignment formula is to be computed
 - Can be when event fires **or** when it is executed
 - (The possibilities arise as side-effect of having event delays)
 - Old SBML: always when it fires
 - Level 2 Version 4: boolean flag **useValuesFromTriggerTime**
- ⦿ Can an event fire at time ≤ 0 ?
 - No—can only be triggered immediately after, i.e., at time > 0
- ⦿ Can you have simultaneous events?
 - Yes—but SBML does not define a tie-breaking algorithm
 - Only states what to do about some cases of cascading events

Currently known event support

● ✓ = tested by Frank Bergmann; ■ = self-reported

Software	Events	Delayed events
iBioSim	✓	✓
PySCeS	✓	✓
MathSBML	✓	✓
ByoDyn	✓	✓
roadRunner	✓	coming
SBML ODE Solver	✓	
SBToolbox	■	■
Virtual Cell	yes (internally)	coming
CellDesigner	■	
GNU McSim	■	
JSim	■	
ProMoT	■	
XPPAUT	■	
PROTON	■	
COPASI	coming	coming