Biocellion Model Specification in XML

Biocellion models can be represented using the XML tags and attributes discussed in this document. Use `biocellion.py` to translate the XML model specification into C++ for compilation and execution.

1 XML Tags

The following tags are currently supported. Some tags are required.

1.1 `<?xml?>`

Begin every model XML file with the XML versioning tag.

```
<?xml version="1.0" encoding="utf-8" ?>
```

1.2 `<model>`

Every model must be contained in the `<model></model>` tag, with exactly one model per XML file. Each `<model>` may contain the following tags.

1.2.1 Allowed Subtags

Tag	Required Count	Notes
`<simulator>`	1	
`<input>`	0 or 1	Not supported yet.
`<solute>`	≥ 0	
`<molecule>`	≥ 0	
`<particle>`	≥ 0	
`<interaction>`	≥ 0	
`<world>`	1	
`<reaction>`	≥ 0	
`<molecularReactions>`	0	Not supported yet.
`<solver>`	≥ 1	
`<agentGrid>`	1	
`<species>`	≥ 0	

1.2.2 Allowed Attributes

No attributes are allowed.

1.2.3 Allowed Text Values

No text values are allowed, only subtags.

1.2.4 Example

```
<model>
  ...
</model>
```

1.3 <param>

<param> tags are used for specifying various parameters to define the model. For example, how many simulated hours to run the simulation. Parameters have a name attribute, and a text value. Most parameters also expect a unit attribute. Many unit conversions are supported.

1.3.1 Allowed Subtags

No subtags are allowed.

1.3.2 Allowed Parameters

No parameters are allowed.

1.3.3 Allowed Attributes

Attribute	Required	Notes
name	Yes	Name of the parameter.
unit	Recommended	Units of the value.

1.3.4 Allowed Text Values

The text values is usually a floating point number, but could be an integer, or a boolean specification.

1.3.5 Examples

```
<model>
  ...
  <simulator>
    <param name="outputPeriod" unit="hour">0.1</param>
    <param name="agentTimeStep" unit="hour">0.1</param>
    <param name="numStateAndGridTimeStepsPerBaseline">1</param>
```

```
    ...
  </simulator>
  ...
</model>
```

1.4 <model>.<simulator>

Every <model> must contain one <simulator> tag. This tag is used to control high-level time related settings for the simulation.

1.4.1 Allowed Subtags

Tag	Required Count	Notes
<param>		See required parameters below.
<timeStep>	1	

1.4.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
outputPeriod	Yes	float	time	Simulated time between data output.
agentTimeStep	Yes	float	time	Simulated time delta per baseline timestep
numStateAndGridTimeStepsPerBaseline	Yes	integer		Internal timesteps per baseline timestep.
restartPreviousRun	No	boolean		Not supported yet.
randomSeed	No	integer		Not supported yet.
chemostat	No	boolean		Not supported yet.
diffusionReactionOnAgentTime	No	boolean		Not supported yet.

1.4.3 Allowed Attributes

No attributes are allowed.

1.4.4 Allowed Text Values

No text values are allowed, only subtags.

1.4.5 Example

```
<model>
  ...
  <simulator>
    <param name="outputPeriod" unit="hour">0.1</param>
    <param name="agentTimeStep" unit="hour">0.1</param>
    <param name="numStateAndGridTimeStepsPerBaseline">1</param>
    ...
```

```
  </simulator>
  ...
</model>
```

1.5 <model>.<simulator>.<timeStep>

The <simulator> must contain one <timeStep> tag. This tag is used to control the end of the simulation.

1.5.1 Allowed Subtags

Tag	Required Count	Notes
<param>		See required parameters below.

1.5.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
endOfSimulation	Yes	float	time	Simulated time to end the simulation.
adaptive	No	boolean		Not supported yet.
timeStepIni	No	float	time	Not supported yet.
timeStepMin	No	float	time	Not supported yet.
timeStepMax	No	float	time	Not supported yet.

1.5.3 Allowed Attributes

No attributes are allowed.

1.5.4 Allowed Text Values

No text values are allowed, only subtags.

1.5.5 Example

```
<model>
  ...
  <simulator>
    ...
    <timeStep>
      <param name="endOfSimulation" unit="hour">1.0</param>
    </timeStep>
  </simulator>
  ...
</model>
```

1.6 <model>.<reaction>

Every <model> may contain multiple <reaction> tags. This tag is used to configure reactions between agents and solutes. These reactions are references inside of <model>.<solver> and <model>.<species> specifications.

1.6.1 Allowed Subtags

Tag	Required Count	Notes
<param>		See required parameters below.
<kineticFactor>	≥ 0	Describes a scaling factor of the reaction.
<yield>	≥ 0	Describes the consumption or production of a solute or particle.

1.6.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
muMax	Yes	float	inverse time	Reaction scaling factor.

1.6.3 Allowed Attributes

Attribute	Required	Notes
name	Yes	Name of the reaction.
class	Yes	ReactionFactor
catalyzedBy	Yes	<particle> reference, can be empty
catalyst	No	<species> reference, can be empty

1.6.4 Allowed Text Values

No text values are allowed, only subtags.

1.6.5 Example

```
<model>
  ...
  <reaction catalyzedBy="biomass" class="ReactionFactor" name="MyReaction1">
    <param name="muMax" unit="hour-1">3.6</param>
    <kineticFactor class="MonodKinetic" solute="MySolute1">
      <param name="Ks" unit="g.L-1">0.12e-3</param>
    </kineticFactor>
    <yield>
      <param name="MySolute1" unit="g.g-1">-0.1</param>
      <param name="MySolute2" unit="g.g-1">0.1</param>
    </yield>
  </reaction>
  ...
</model>
```

1.7 <model>.<reaction>.<yield>

Every <model>.<reaction> may contain one <yield> tag. This tag is used to configure solutes, particles and molecules that are consumed or produced by the reaction. They reference <model>.<particle>, <model>.<solute> and <model>.<molecule> specifications with the name attribute of a a <param> tag.

1.7.1 Allowed Subtags

Tag	Required Count	Notes
<param>		See required parameters below.

1.7.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
particle-name	No	float	mass.mass-1	amount produced (negative means consumed)
solute-name	No	float	mass.mass-1	amount produced (negative means consumed)
molecule-name	No	float	mass.mass-1	amount produced (negative means consumed)

1.7.3 Allowed Attributes

Attribute	Required	Notes
None	No	No attributes allowed

1.7.4 Allowed Text Values

The text is a numeric value. Positive values indicate production, and negative values indicate consumption.

1.7.5 Example

```
<model>
  ...
  <reaction catalyzedBy="biomass" class="ReactionFactor" name="MyReaction1">
    <param name="muMax" unit="hour-1">3.6</param>
    <kineticFactor class="MonodKinetic" solute="MySolute1">
      <param name="Ks" unit="g.L-1">0.12e-3</param>
    </kineticFactor>
    <yield>
      <param name="MySolute1" unit="g.g-1">-0.1</param>
      <param name="MySolute2" unit="g.g-1">0.1</param>
    </yield>
  </reaction>
  ...
</model>
```

1.8 <model>.<reaction>.<kineticFactor>

Every <model>.<reaction> may contain multiple <kineticFactor> tags. This tag is used to configure scaling factors for reaction rates. Several kinetic factors are possible. They are all multiplied together to calculate the strength of the reaction.

1.8.1 Allowed Subtags

Tag	Required Count	Notes
<param>		See required parameters below.

1.8.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
Ki	No	float	mass.volume-1	Constant used for some classes of kinetic factor
Ks	No	float	mass.volume-1	Constant used for some classes of kinetic factor
permeability	No	float	area.time-1	Constant used for some classes of kinetic factor

1.8.3 Kinetic Factor Styles

- FirstOrderKinetic The simplest factor, returns 1.0.
- SimpleInhibition High density of an item will restrict the reaction rate. Returns $K_i/(K_i + d)$ where d is the item's density.
- MonodKinetic Low density of an item will restrict the reaction rate. Returns $d/(K_s + d)$ where d is the item's density.
- LinearKinetic Low density of an item will restrict the reaction rate. Returns $K_s d$ where d is the item's density.
- KineticAgentSurfaceArea Only supported for species items. Returns the a/v where a is the agent's surface area and v is the agent's volume.
- KineticPermeability Returns permeability, as specified in the parameter.

1.8.4 Allowed Attributes

Attribute	Required	Notes
`class`	Yes	FirstOrderKinetic, SimpleInhibition, MonodKinetic, LinearKinetic, KineticAgentSurfaceArea, KineticPermeability
`molecule`	No	molecule's density to use in factor formula
`solute`	No	solute's density to use in factor formula
`species`	No	specie's density to use in factor formula

1.8.5 Allowed Text Values

No text values

1.8.6 Example

```
<model>
  ...
  <reaction catalyzedBy="biomass" class="ReactionFactor" name="MyReaction1">
    <param name="muMax" unit="hour-1">3.6</param>
    <kineticFactor class="MonodKinetic" solute="MySolute1">
      <param name="Ks" unit="g.L-1">0.12e-3</param>
    </kineticFactor>
    <yield>
      <param name="MySolute1" unit="g.g-1">-0.1</param>
      <param name="MySolute2" unit="g.g-1">0.1</param>
    </yield>
  </reaction>
  ...
</model>
```

1.9 `<model>.<solver>`

Every `<model>` must contain at least one `<solver>` tag. This tag is used to configure PDE solvers for diffusion. Even if there are no solutes to diffuse, a solver must still be specified. Required biocellion configuration parameters, such as refine-ratio are calculated from solver parameters.

1.9.1 Allowed Subtags

Tag	Required Count	Notes
`<param>`		See required parameters below.
`<reaction>`	≥ 0	References a `<model>.<reaction>`

1.9.2 Allowed Parameters

Parameter	Required	Type	Units	Notes
`active`	Yes	boolean	bool	Not used, but required.
`preStep`	Yes	integer	int	Number of pre-step iterations solver uses.
`postStep`	Yes	integer	int	Number of post-step iterations solver uses.
`bottomStep`	Yes	integer	int	Number of bottom-step iterations solver uses.
`nCycles`	Yes	integer	int	Number of iterations solver uses.
`pdeSolverType`	No	string	string	`StateSteady`, `TimeDependent`.
`refineRatio`	No	integer	int	AMR refinement ratio for solver.
`AMRLevels`	No	integer	int	Number of AMR levels for solver.
`numTimeSteps`	No	integer	int	Number of time steps.

1.9.3 Allowed Attributes

Attribute	Required	Notes
`name`	Yes	Name of the solver.
`class`	Yes	Not used, but required.
`domain`	Yes	`<ComputationDomain>` reference.

1.9.4 Allowed Text Values

No text values are allowed, only subtags.

1.9.5 Example

```
<model>
  ...
  <solver class="SolverSimple" domain="MyComputationDomain" name="solutes">
    <param name="active">true</param>
    <param name="preStep">150</param>
    <param name="postStep">150</param>
    <param name="bottomStep">150</param>
```

```
    <param name="nCycles">5</param>
    <param name="pdeSolverType">TimeDependent</param>
    <reaction name="MyReaction1"/>
    ...
  </solver>
  ...
</model>
```