March 5, 2020: Hi, I just moved this over to a wiki on github:
https://github.com/MathCancer/PhysiCell/wiki/XML-based-Cell-Definitions

The document below is superseded by the wiki above.

Goals:

1. Simplify model setup
2. Reduce repetition (see 1) -- no longer need to type base parameters over and over again for each cell type
3. Easier to enforce best practices and avoid PhysiCell quirks
    a. Same custom data in all cell definitions, starting in cell defaults (avoid data output quirks)
    b. Cell cycle, phenotype set up first in cell defaults (best practice)
4. After this, users focus on writing custom functions and attaching them to the right cell definitions. No more hand typing / creating these cell definitions.
5. Make it easier to read in initial cell positions (for a future release) -- they can list (x,y,z), ID, and which cell definition. Then perhaps any unique initialization.
6. Easier bridge to linking SBML / molecular models -- specific path to SBML in the cell definition
7. Reduce number of custom parameters needed (part of this is cell def setup)
8. facilitate future where basic models can be created entirely at runtime in some sort of python+jupyter+PhysiCell environment. OR a sort of “PhysiCell studio”

Intended implementation (when parsing):

1. Read and create a default cell definition
    a. Base / default parameters fully specified here.
    b. Custom data also specified here.
2. All other cell definitions copy the default (at time of creation in PhysiCell instance).
    a. Inheritance: all other cell definitions inherit default parameters from the default definition.
3. All other cell definitions (in XML) are truncated -- only specific parts that deviate from the default are written
4. Create a master list of all cell definitions (unique IDs and character names). This data structure can be queried.

Anticipated ecosystem impacts:

1. xml2jupyter based tools will need a new tab for cell definitions
    a. Maybe subtabs to facilitate easy browsing of each cell individual definition
    b. Inheritance will need to be accounted for in the tab somehow. Changing a value in defaults needs to “percolate” to all other cell defs, unless overwritten by user.

Document history:
Feb 24, 2020: Paul Macklin starts this outline of the draft specification.
Feb 24, 2020: Edits opened up to MathCancer lab XML core team (Randy Heiland, Nick Mowery, Daniel Mishler)

intended timeline
Feb 2020: comments: All MathCancer Lab
late Feb 2020: comments by key collaborators at BCS, Institut Curie (especially PhysiBoSS)

late Feb 2020: draft implementation in PhysiCell
Week 1 March 2020: rework sample projects using the new spec
Week 2 March 2020: test making new projects from template 2D and template 3D projects

early Mar 2020: virtual town hall
target release: late March 2020

late March 2020: plan a training module using the XML ??

Draft XML on next page.

Overall structure (by example)

```
<cell_definitions>
     <cell_definition name=”default”>
          <phenotype>
               <cycle />
               <death />
               <volume />
               <!-- geometry omitted from now, as auto-calculated from volume -->
               <mechanics />
               <motility />
               <secretion />
               <molecular />
          </phenotype>
          <custom_data>
               <elastic_coefficient length=”1” units=”1/min”>1.0</elastic_coefficient>
               <attachment_point length=”3” units=”micron”>-12.8,13.9,0.0</attachment_point>
          </custom_data>
     </cell_definition>

     <cell_definition name=”cancer”>
          <phenotype>
               <cycle>
                    <transition_rates units=”1/min”>
                         <rate start_index=”0” end_index=”1”>0.0001</rate>
                    </transition_rates>
               </cycle>
               <volume>
                    <total units=”micron^3”>2.15e3</total>
               </volume>
     </cell_definition>
</cell_definitions>
```

Details on individual blocks on following pages

```
<cycle model_ID="0"> <!-- advanced_Ki67_cycle_model : codes are defined in PhysiCell_constants.* -->
     <transition_rates units="1/min"> <!-- units here since all transition rates have same units -->
          <rate start_index="0" end_index="1">0.00001</rate>
          <rate start_index="1" end_index="2">0.00001</rate>
          <rate start_index="2" end_index="0">0.00001</rate>
     </transition_rates>
</cycle>


<death />
<!-- give baseline rates for the various death models -->
<!-- is this would be where we should specify if necrosis is stochastic or deterministic?? -->
     <!-- or is that more of a custom parameter or option that belongs elsewhere? -->
<!-- probably also give the altered fluid and solid gain/loss rates -->
<!-- perhaps also set cell lysis parameters, and parameters on how much of internal contents are released at lysis (if tracked)

<!-- give the "mature" or "target" cell volume here, and fluid / solid change rates -->
<volume>
     <total units="micron^3" />
     <fluid_fraction units="dimensionless" />
     <nuclear units="micron^3" />
     <cytoplasmic units="micron^3" />
     <fluid_change_rate units="1/min" />
     <cytoplasmic_solid_change_rate units="1/min" />
     <nuclear_solid_change_rate units="1/min" />
<!-- anything else? maybe nuclear : cytoplasmic is an internaly calculated thing, instead of specified -->
</volume>

<!-- minimal parameters here for now -->
<mechanics>
     <cell_cell_adhesion_strength units="micron/min" />
     <cell_cell_repulsion_strength units="micron/min" />
     <relative_maximum_adhesion_distance units="dimensionless" />
</mechanics>

<motility>
     <options enabled="true" 2D="true" />
     <speed units="micron/min" />
     <persistence_time units="min" />
     <migration_bias units="dimensionless" />
     <chemotaxis enabled="true" substrate_index="0" direction="1" />
          <!-- index matches one of the microenvironment indices above -->
          <!-- 1 if up gradient, -1 if against gradient -->
</motility>
```

```
<!-- this is going to be the most laborious part for people -->
<!-- we ***could** include 3 in template by default, and just don't parse the later entries if there are fewer substrates -->
<!-- we could also use a shorter, vector form like <secretion_rates units="1/min">0,10.2,0.0</secretion_rates> →
<!-- we could also use the convention that unspecified rates are set to zero. In this case, only set the nonzero rates -->
<!-- moreover, users could leave this section blank, and only set nonzero rates for specific cell types. -->
<secretion>
      <!-- generally use this OR net export rates -->
      <secretion_rates units="1/min">
            <secretion_rate index="0">0.0</secretion_rate>
            <secretion_rate index="1">10.2</secretion_rate>
            <secretion_rate index="2">0.0</secretion_rate>
      </secretion_rates>
      <secretion_targets units="substrate density">
            <secretion_target index="0">1.0</secretion_target>
            <secretion_target index="1">1.0</secretion_target>
            <secretion_target index="2">1.0</secretion_target>
      </secretion_rates>
      <uptake_rates units="1/min">
            <uptake_rate index="0">10.0</uptake_rate>
            <uptake_rate index="1">1.0</uptake_rate>
            <uptake_rate index="2">0.0</uptake_rate>
      </uptake_rates>
      <!-- generally use this OR secretion_rates -->
      <net_export_rates units="total substrate/min">
            <net_export_rate index="0">0.0</net_export_rate>
            <net_export_rate index="1">1.0</net_export_rate>
            <net_export_rate index="2">-3.0</net_export_rate>
      </net_export_rates>
</secretion>

<!-- secretion rate and target, uptake rate, etc. for each substrate.-->
      <!-- dimensions: 1/time, density, 1/time -->
<!-- also for future: net export rate. dimensions: substance/time -->

<molecular />
<!-- mostly a placeholder, but name of SBML model -->
<!-- eventually some index mapping here -->
```