SYSC 5906 G

Methodologies for Discrete Event Modeling and Simulation

A DEVS Based Simulation of Mitosis Using Cadmium

Team Members:

Hamnah Qureshi, 101225634

Naomi Amaya Lovett, 101223562

Grant Achuzia, 101222695

Due: March 2nd, 202

Table of Contents

PART I: Defining the Mitosis System

Model Choice Overview

Cell division through mitosis is a biological process that uses a finite sequence of phases (interphase, prophase, metaphase, anaphase, telophase, and cytokinesis). Each phase is made up of discrete structural changes in cellular components throughout the cell.

The goal of this project is to model mitosis as a Discrete Event Dynamic System (DEDS) with the DEVS formalism. The model will aim to capture the following:

- The sequence of mitotic phases
- The event driven interactions between cellular parts
- The state changes of each component throughout mitosis

This model can be used to simulate a normal mitotic progression, verify that the phases are ordered correctly, and serve as a foundation to extend the model to abnormal cell division scenarios.

Model Diagram

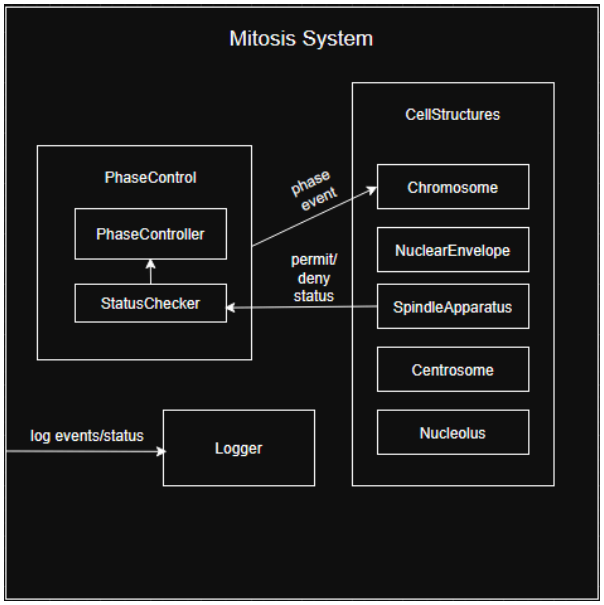


Figure 1: Draft MitosisSystem Model Diagram

Model Structure

Level 1: Top Coupled Model

MitosisSystem: Top level model representing the complete mitosis process. It connects control logic, cell components, and outputs. It does not have state or behavior of its own, rather, it defines structures and connections

Level 2: Coupled Subsystems

- **PhaseControl:** Coordinates the system's attempts to move from one phase to the next, letting the system know if a transition is allowed.
- **CellStructures:** Groups the models that represent the physical parts of the cell involved in mitosis. Submodels respond to phase changes.
- **Logger**: Records phase transitions and events in the simulation.

Level 3: Atomic Models in Subsystems

- **Chromosome:** Tracks the varying behaviors/actions in the chromosome. Updates state based on phase events and reports changes in the chromosome.
- **NuclearEnvelope:** Updates state based on phase and reports whether the envelope is broken or reformed.
- **SpindleApparatus:** Represents spindle formation and attachment to chromosomes. Reports when attachment is complete.
- **Centrosome:** Tracks centrosome duplication and movement to opposite poles. Reports whether centrosomes have reached opposite poles.
- **Nucleolus:** Represents the appearance/disapperance of the nucleolus as a visible indication of early and late mitosis phases.
- **PhaseController:** Maintains the current mitosis phase, requests transitions to the next phase, and broadcasts a phase event when the transition is allowed.
- **StatusChecker:** Receives updates from cell components and decides if the system can move on to the next phase. It sends a permit or deny signal to the PhaseController.

Technical Definitions

Chromosome: It is a tightly packed structure made of DNA and proteins. It carries the genetic information and needs to make sure the DNA is probably distributed during cell division. They condense in prophase, align in metaphase, separate in anaphase and decondense in telophase.

DNA: DNA is the molecule that carries the genetic instructions for how a cell functions and grows. DNA is organized differently depending on the stage of the cell cycle. During prophase, chromatin condenses into chromosomes. They align and separate in metaphase and anaphase, then decondense back into chromatin during telophase.

Nucleus: It controls the cell's activities and protects its DNA. During prophase, the nuclear membrane breaks down. It remains absent in metaphase and anaphase to allow chromosome movement, then reforms around the chromosomes in telophase.

Spindle apparatus: This component is made of microtubules that form during mitosis and pulls sister chromatids apart. This forms during prophase, attaches to chromosomes in metaphase, pulls sister chromatids in anaphase, and breaks down in telophase.

Centrosome: The centrosome organizes the spindle fibers for chromosome movement. It duplicates before mitosis, moves to opposite poles during prophase, helps pull chromosomes apart in anaphase, and remains at the poles until the end of telophase.

PART II: A DEVS Formal Specification of the Mitosis Model

Updated Model Diagram

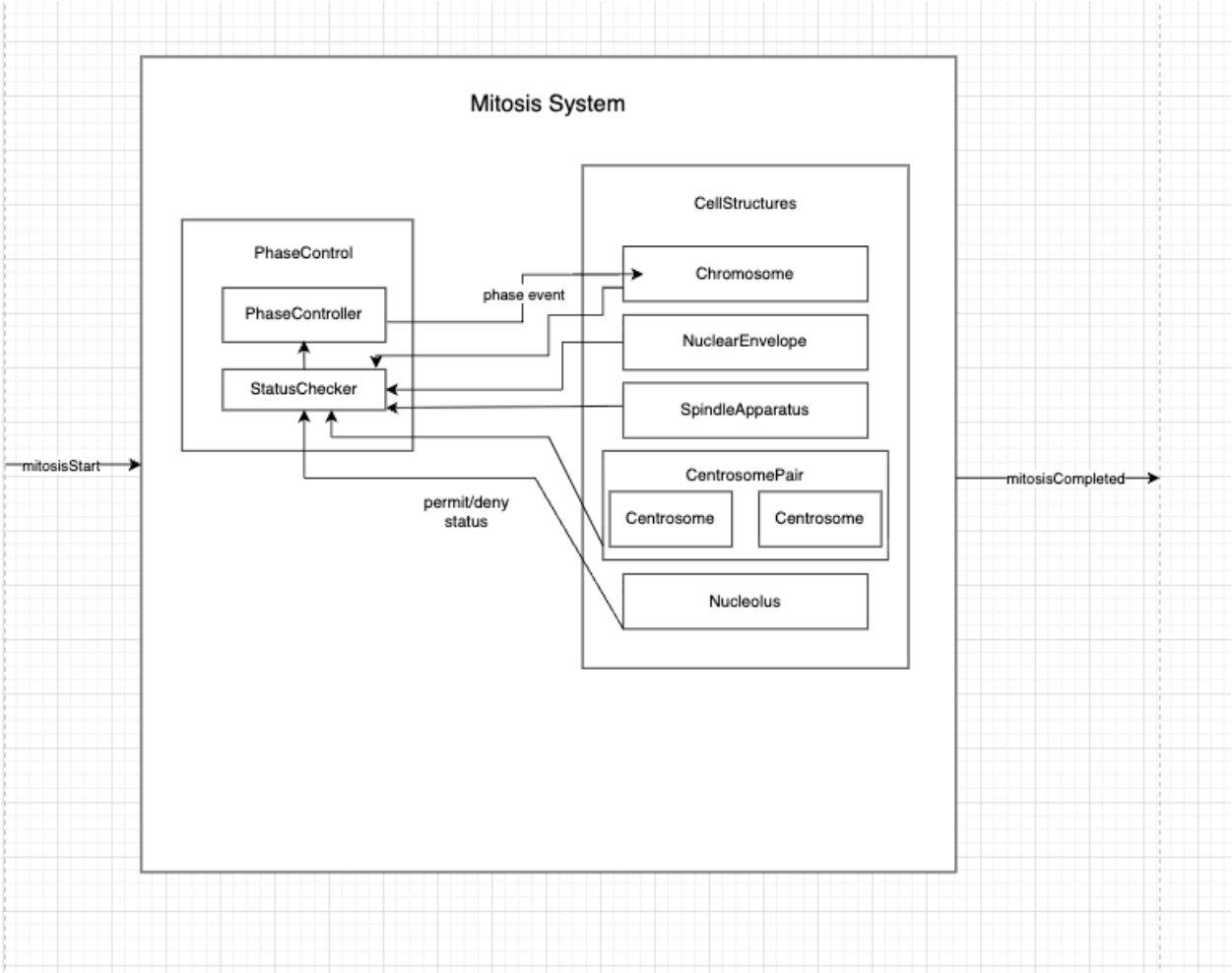


Figure 2: Updated Model Diagram

Formal Specifications for Coupled Models

For a coupled model, CM = <X, Y, D, {Md | d ∈ D}, EIC, EOC, IC, select> where , X is the input event set, Y is the output event set, D is the set of components, M_d is the component models, EIC is the external input couplings, EOC is the external output couplings, IC is the internal input couplings, and Select is the tiebreaker function.

MitosisSystem

X = {start}

Y = {mitosis_complete | mitosis_complete∈{true, false}}

D = {PhaseControl, CellStructures}

M_d = {$M_{PhaseControl}$, $M_{CellStructure}$}

EIC = {((Self, start), (PhaseController, start))}

EOC = {((PhaseControl, mitosis_complete), (Self, mitosis_complete))}

IC = {

((PhaseControl, phase_event), (CellStructures, phase_in)),

((CellStructures, chrom_status), (PhaseControl, chrom_status)),

((CellStructures, ne_status), (PhaseControl, ne_status)),

((CellStructures, sp_status), (PhaseControl, sp_status)),

((CellStructures, cen_status), (PhaseControl, cen_status)),

((CellStructures, nucleo_status), (PhaseControl, nucleo_status))

}

Select = {PhaseControl > CellStructures}

PhaseControl

X = {start}

Y = {phase_out ∈ {Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

D = {PhaseController, StatusChecker}

M_d = {$M_{PhaseController}$, $M_{StatusChecker}$}

EIC = {((Self, start), (PhaseController, start))}

EOC = {((PhaseController, phase_event), (Self, phase_out))}

IC = {((StatusChecker, allow_transition), (PhaseController, transition)),

((StatusChecker, deny_transition), (PhaseController, do_not_transition))}

Select = {StatusChecker > PhaseController}

CellStructures

X = {phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {chrom_status, ne_status, sp_status, cen_status, nucleo_status}

D = {Chromosome, NuclearEnvelope, SpindleApparatus, CentrosomePair, Nucleolus}

M_d = {$M_{Chromosome}$, $M_{NuclearEnvelope}$, $M_{SpindleApparatus}$, $M_{CentrosomePair}$, $M_{Nucleolus}$}

EIC = {((Self, phase_in),(d, phase_in))}

EOC = { ((Chromosome, status_out), (Self, chrom_status)),
((NuclearEnvelope, status_out), (Self, ne_status)),
((SpindleApparatus, status_out), (Self, sp_status)),
((CentrosomePair, status_out), (Self, cen_status)),
((Nucleolus, status_out), (Self, nucleo_status)) }

IC = ∅

Select = {Chromosome > NuclearEnvelope > SpindleApparatus > CentrosomePair > Nucleolus}

CentrosomePair

X ={phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

D = {CentrosomeTop, CentrosomeBottom}

M_d = {$M_{CentrosomeTop}$, $M_{CentrosomeBottom}$}

EIC = {((Self, phase_in),(d, phase_in))}

EOC = { (($M_{CentrosomeTop}$, status_out, $M_{CentrosomeTop}$, status_out), (Self, status_out))}

IC = ∅

Select = {$M_{CentrosomeTop}$ > $M_{CentrosomeBottom}$}

Formal Specifications for Atomics Models

For an atomic model, M=⟨X, Y, S, δ_{int}, δ_{ext}, λ, ta⟩ where X is the input event set, Y is the output event set, S is the set of states, δ_{int} is the internal transition function, δ_{ext} is the external transition function, λ is the output function, and ta is the time advance function.

Note: The port for phase events was originally called `input_phase` as seen in the atomic model diagrams below. However, the specifications have since been updated to use the name `phase_in`.

Chromosome

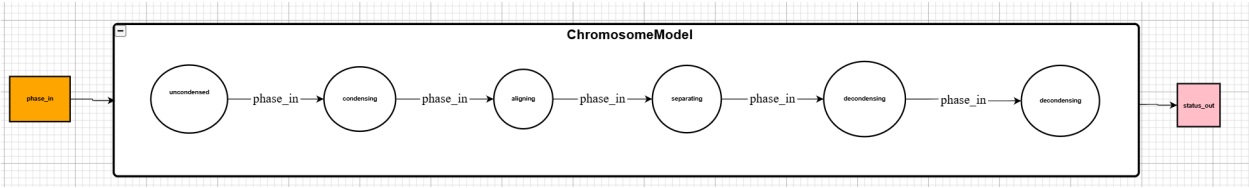


Figure 3: Chromosome Atomic Model Diagram

X ={phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

S = {uncondensed, condensed, aligned, separated, condensing, aligning, separating, decondensing}

$\boldsymbol{\delta}_{int}$ (Active) =

condensing → condensed

aligning → aligned

separating → separated

decondensing → uncondensed

$\boldsymbol{\delta}_{ext}$ =

Prophase → condensing

Metaphase → aligning

Anaphase → separating

Telophase → decondensing

Cytokinesis → decondensing

λ =

if (S == condensing || S == aligning || S == separating || S == decondensing)

status_out = ready

else

status_out = not_ready

ta =

ta(condensing) = t1

ta(aligning) = t1

ta(separating) = t1

ta(decondensing) = t1

ta(otherwise) = ∞

NuclearEnvelope

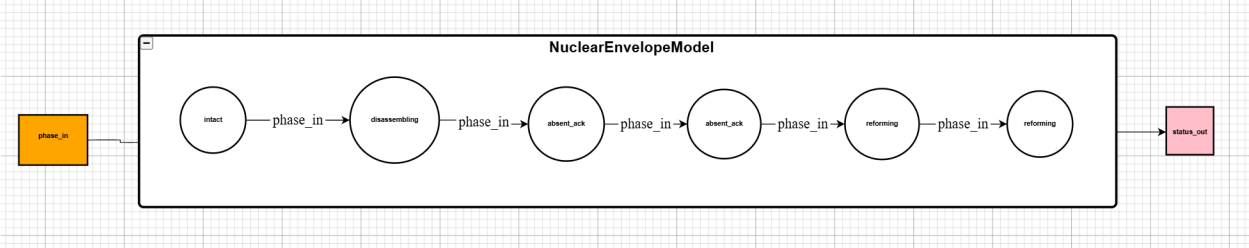


Figure 4: Nuclear Envelope Atomic Model Diagram

X = {phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

S = {intact, disassembling, absent_ack, absent, reforming, restored}

$\boldsymbol{\delta}_{int}$ (Active) =

disassembling → absent

absent_ack → absent

reforming → restored

$\boldsymbol{\delta}_{ext}$ =

Interphase → intact

Prophase → disassembling

Metaphase & Anaphase → absent_ack

Telophase & Cytokinesis → reforming

λ =

if (S == disassembling || S == reforming || S== absent_ack)

status_out = ready

else

status_out = not_ready

ta =

ta(disassembling) = t_1

ta(reforming) = t_1

ta(absent_ack) t_2

ta(otherwise) = ∞

SpindleApparatus

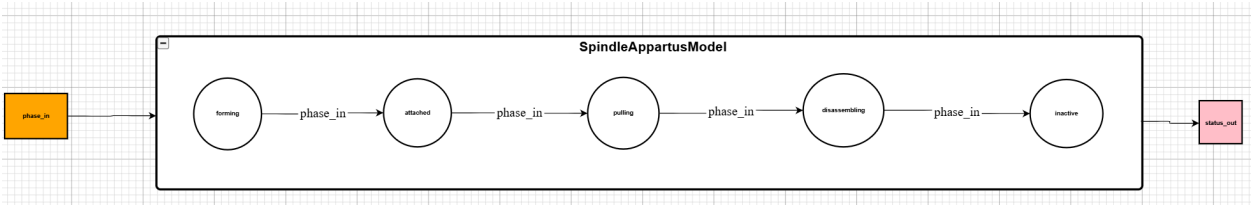


Figure 5: SpindleApparatus Atomic Model Diagram

X = {phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

S = {forming, attached, pulling, disassembling, inactive}

$\boldsymbol{\delta}_{int}$ (Active) = Passive

$\boldsymbol{\delta}_{ext}$ =

Prophase → forming

Metaphase → attached

Anaphase → pulling

Telophase → disassembling

Cytokinesis → inactive

λ =

if (S == || S== reforming || S== absent_ack)

status_out = ready

else

status_out = not_ready

ta =

ta(forming) = t_1

ta(attached) = t_2

ta(pulling) = t_2

ta(disassembling) = t_2

ta(inactive) = 0.0

ta(otherwise) = ∞

Centrosome

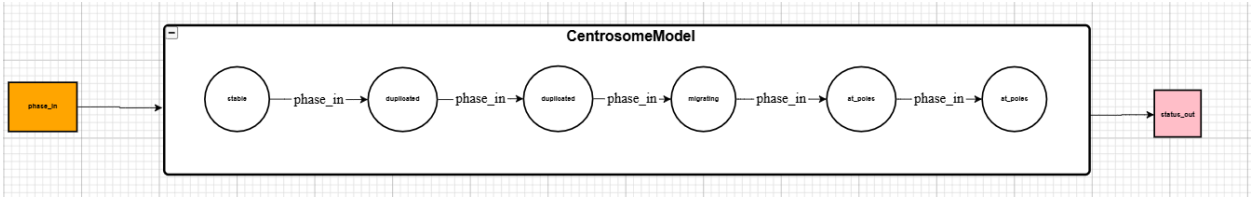


Figure 6: Centrosome Atomic Model Diagram

X = {phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

S = {duplicated, migrating, at_poles, stable}

$\boldsymbol{\delta}_{int}$ (Active) = Passive

$\boldsymbol{\delta}_{ext}$ =

Interphase -> stable

Prophase -> duplicated

Metaphase -> migrating

Anaphase -> at_poles

Telophase -> at_poles

λ =

```
if (S == at_poles || S == stable || S == duplicated || S == migrating):
        status_out = ready
else:
        status_out = not_ready
```

ta =

ta(duplicated) = t_1

ta(migrating) = t_2

ta(at_poles) = 0.0

ta(otherwise) = ∞

Nucleolus

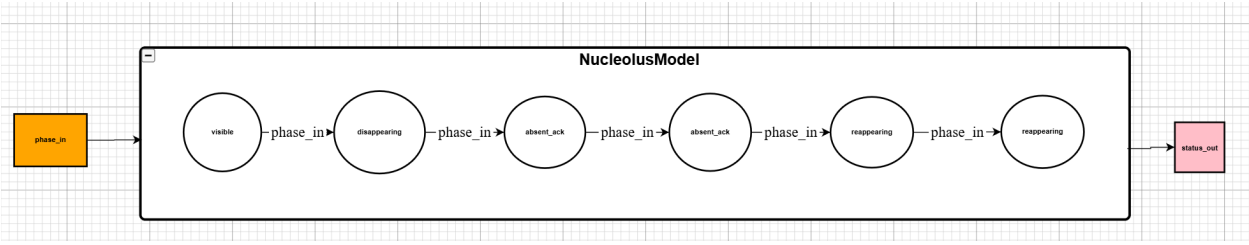


Figure 7: Nucleolus Atomic Model Diagram

X = {phase_in ∈ { Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}}

Y = {status_out ∈ {ready, not_ready}}

S = {visible, disappeared, reappearing, disappearing, absent_ack}

$\boldsymbol{\delta}_{int}$ (Active) =

disappearing → disappeared

absent_ack → disappeared

reappearing → visible

$\boldsymbol{\delta}_{ext}$ =

Interphase → visible

Prophase → disappearing

Metaphase & Anaphase → absent_ack

Telophase & Cytokinesis → reappearing

λ =

if (S == disappearing || S == absent_ack || S == reappearing):

status_out = ready

else:

status_out = not_ready

ta =

ta(disappeared) = t_1

ta(reappearing) = t_2

ta(absent_ack) = t_3

ta(otherwise) = ∞

StatusChecker

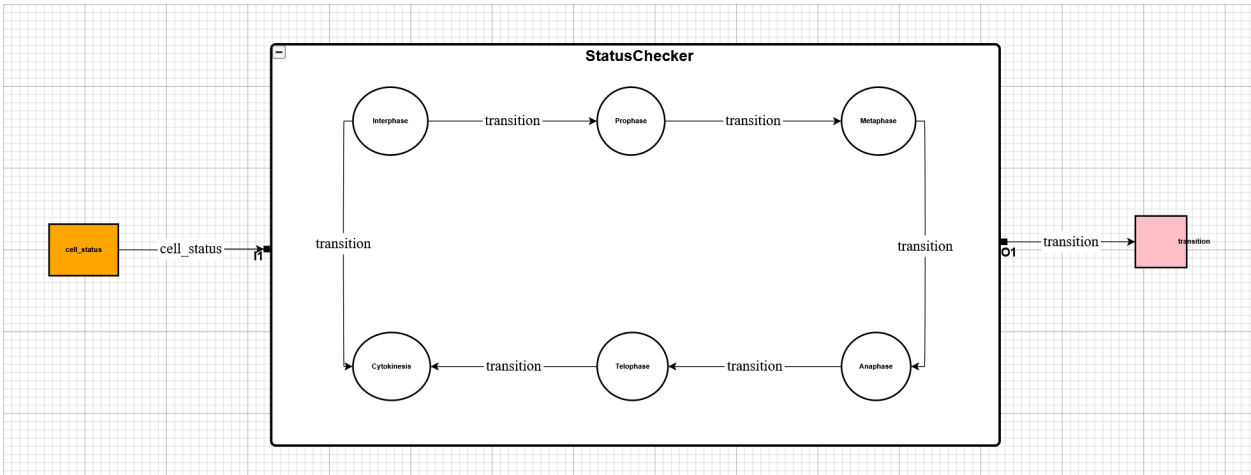


Figure 8: Status Checker Atomic Model Diagram

X = cell_status ∈ {chrom_status, ne_status, sp_status, cen_status, nucleo_status}, start_in ∈ {start}

Y = transition ∈ {allow_transition, deny_transition}

S = {Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}

$\boldsymbol{\delta}_{int}$ (Active) = Passive

$\boldsymbol{\delta}_{ext}$ = When all of the inputs are set to “ready”, switch to the next state

Interphase → Prophase

Prophase → Metaphase

Metaphase → Anaphase

Anaphase → Telophase

Telophase → Cytokinesis

λ =

```
if ALL cell_status == ready:
        if S == Cytokinesis return "Simulation Complete"
   return allow_transition
else:
   stay in deny_transition
```

ta = ∞, depends on the external inputs from the cell structures

PhaseController

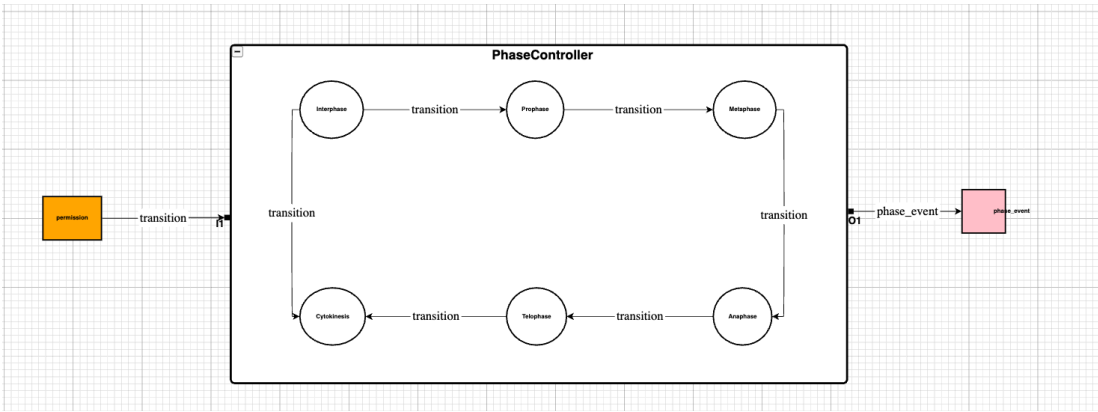


Figure 9: Phase Controller Atomic Model Diagram

X = permission ∈ {transition, do_not_transition}

Y = phase_event ∈ {Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}

S = {Interphase, Prophase, Metaphase, Anaphase, Telophase, Cytokinesis}

$\boldsymbol{\delta}_{int}$ (Active) = Passive waiting

$\boldsymbol{\delta}_{ext}$ =

if transition:

Interphase → Prophase

Prophase → Metaphase

Metaphase → Anaphase

Anaphase → Telophase

Telophase → Cytokinesis

Cytokinesis → Interphase

else if do_not_transition:

stay in current state

λ = outputs current phase event

ta = ∞ (waiting), 0 on transition

Atomic Model Test Cases

Chromosome Test Cases

Initial State: uncondensed

TC-C1: Prophase Tigger

Starts from the uncondensed state and goes to condensed based on a single Prophase at t =0. After an internal delay to aligned, it should output `not_ready` until the full progression is done; then it goes to ready.

Item	Value
Initial State	uncondensed
Input	prophase at t=0
Expected State	condensed
After ta(condensed) = t1	aligned
Output	not_ready until completed
Final Output	ready when completed

TC-C2: Full Phase Progression

Feeds the full phase sequence (Prophase -> Metaphase -> Anaphase -> Telophase) and checks that the chromosome goes through its full sequence of states (and outputs ready when completed).

Input sequence:

```
0.0 Prophase
10.0 Metaphase
15.0 Anaphase
20.0 Telophase
```

Expected Output:

```
condensed → aligned → separating → decondensing → completed
```

- Output = ready when progression complete

TC-C3: Invalid Phase Jump

This test sends an Anaphase at t = 0, which should cause the chromosome to stay uncondensed because it never received Prophase/Metaphase (so no transition).

Input Sequence

```
0.0 Anaphase
```

Expected Output

NuclearEnvelope Test Cases

TC-NE1: Prophase Breakdown

One Prophase at t =0, which means the nuclear envelope should go to disassembling then (after ta) to absent and output `not_ready`.

Input Sequence:

```
0.0 Prophase
```

Expected Output:

- State = assembling
- Eventually absent
- Output = not_ready

TC-NE2: Telophase Reforming

One Telophase at t =0; tests that the nuclear envelope goes from reforming to restored and outputs ready.

Input Sequence:

```
0.0 Telophase
```

Expected Output:

```
reforming → restored
```

- Output = ready

SpindleApparatus Test Case

TC-SP1: Normal Sequence

This test goes through the full phase sequence for the spindle apparatus (forming -> attached -> pulling -> disassembling -> inactive) and outputs ready when it is complete.

Input sequence:

```
0.0 Prophase
10.0 Metaphase
20.0 Anaphase
30.0 Telophase
```

Expected Output:

```
forming → attached → pulling → disassembling → inactive
```

- Output = ready when progression complete

Centrosome Test Case

TC-CT1: Normal Pole Migration

The test feeds the centrosome the phases Interphase, then Prophase, Metaphase, then Anaphase at the given times and checks that its state goes from stable-> duplicated -> migrating -> at_poles, and that it reports ready once it is at_poles.

Input sequence:

```
0.0 Interphase
5.0 Prophase
10.0 Metaphase
15.0 Anaphase
```

Expected Output:

```
stable → duplicated → migrating → at_poles
```

- Output = ready when at_poles

Nucleolus Test Case

TC-NU1: Disappearance and Reappearance

Sends Prophase at time 0 and Telophase at time 20 and checks that the nucleolus goes from visible -> disappeared -> reappearing -> visible. It should then report ready once that full sequence is done.

Input sequence:

```
0.0 Prophase
20.0 Telophase
```

Expected Output:

```
visible → disappeared → reappearing → visible
```

- output = ready when progression complete

StatusChecker Test Case

TC-SC1: All Components Ready

This test gives the StatusChecker “ready” from all components (chromosome, nuclear envelope, spindle, centrosome, nucleolus) at the same time and checks that it outputs allow_transition.

Inputs at the same time:

```
chrom_status = ready
ne_status = ready
sp_status = ready
cen_status = ready
nucleo_status = ready
```

Expected Output:

- Output: allow_transition

PhaseControllerTest Case

TC-PC1: All Components Ready

The test sends “transition” and checks that the PhaseControllermoves to the next phase (e,g, from Interphase -> Prophase), outputs that phase, and that it’s next internal transition happens immediately.

Input:

```
Permission = transition
```

Expected Output:

- Output: phase_event at which the model is transitioning
- Ex : Interphase → Prophase
- ta = 0
- Ex: Phase_event = Prophase

TC-PC2: Deny Transition

The test sends a “do not transition” and checks that the PhaseController does not change phase, does not produce an output, and keeps waiting until it receives “transition”.

Inputs at the same time:

```
Permission = do_not_transition
```

Expected Output

- Output: None
- ta = ∞ (waiting)
- deny transition until Permission = transition

Note: The test result files for these simulation results can be found at https://github.com/GAchuzia/Cadmium-Mitosis/tree/main/simulation_results

PART III: Execution Results and Analysis of the Mitosis Simulation

Simulation Results

For the atomic models, we ran the above mentioned test cases and for all of them received the expected output. This helped ensure the Cadmium DEVS atomic models were done correctly but that still meant we can run into issues when combining these atomic models to work together synchronously. Before testing out the entire model, we tested out the coupled models separately to make sure their behavior is correct and ensure our time advance values were ideal.

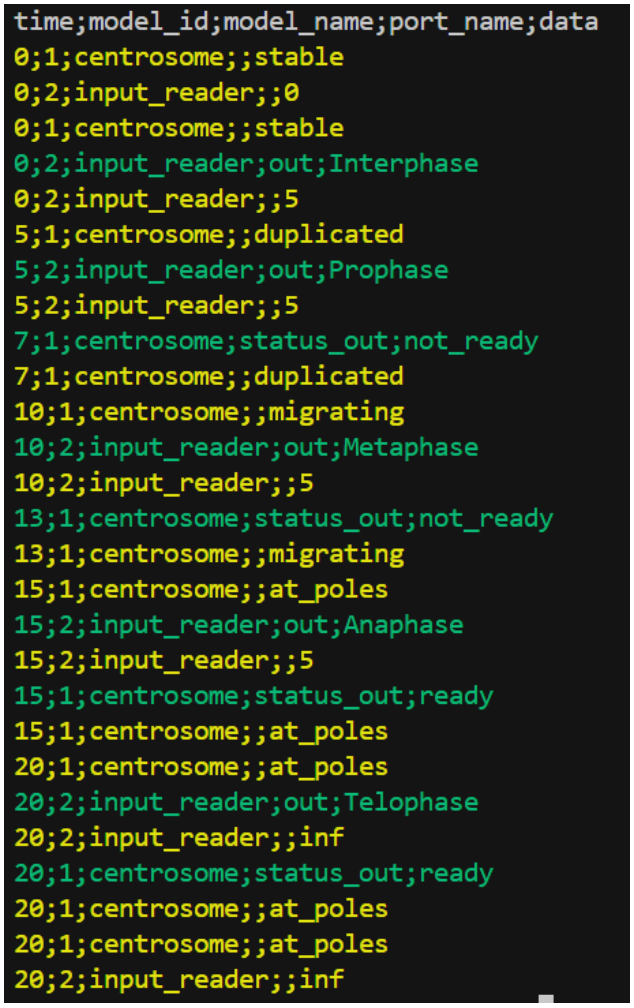
CentrosomePair (Coupled Model)

The CentrosomePair is made up of two Centrosome atomic models, $\text{Centrosome}_{\text{Top}}$ and $\text{Centrosome}_{\text{Bottom}}$. These models are synched by the CentrosomePair as it takes in input_phase and passed into the both atomic models as their input. The model takes the output of both Centrosomes which returns either "ready" or "not_ready". If both of them are "ready", it will return "ready". If only one of them or both are not, it will return "not_ready".

Input Sequence:

```
0.0 Interphase
5.0 Prophase
10.0 Metaphase
15.0 Anaphase
20.0 Telophase
```

Simulation result:



```
time;model_id;model_name;port_name;data
0;1;centrosome;;stable
0;2;input_reader;;0
0;1;centrosome;;stable
0;2;input_reader;out;Interphase
0;2;input_reader;;5
5;1;centrosome;;duplicated
5;2;input_reader;out;Prophase
5;2;input_reader;;5
7;1;centrosome;status_out;not_ready
7;1;centrosome;;duplicated
10;1;centrosome;;migrating
10;2;input_reader;out;Metaphase
10;2;input_reader;;5
13;1;centrosome;status_out;not_ready
13;1;centrosome;;migrating
15;1;centrosome;;at_poles
15;2;input_reader;out;Anaphase
15;2;input_reader;;5
15;1;centrosome;status_out;ready
15;1;centrosome;;at_poles
20;1;centrosome;;at_poles
20;2;input_reader;out;Telophase
20;2;input_reader;;inf
20;1;centrosome;status_out;ready
20;1;centrosome;;at_poles
20;1;centrosome;;at_poles
20;2;input_reader;;inf
```

Figure 10: Simulation Results for CentrosomePair

As we can see, at 0s, both Centrosome atomic models start at "stable" state. At 5s, the input_phase is "Prophase". Then, both models switch to "duplicated" state and status_out is printed as "not_ready" at 7s as the time advance is 2.0 for that state. At 10s, the input_phase is "Metaphase", then the model goes to "migrating" state and status_out is printed as "not_ready" at 13s as the time advance is 3.0 for that state. At 15s, the input_phase is "Anaphase", then the model goes to "at_poles" state and status_out is printed as "ready" as the time advance is 0.0 for that state so it would log it. The same thing happens at t=20 when input_phase is "Telophase".

PhaseControl(Coupled Model)

The PhaseControl model is made up of two atomic models: the PhaseController and the StatusChecker models. The StatusChecker takes in all five outputs from the CellStructures model and compares their status as ready / not ready. Once all five CellStructure outputs are set to ready, the StatusChecker outputs an “allow_transition” message to the PhaseController which then sends out the next mitosis state name to the CellStructures as “phase_event”.

Input sequence example:

Spindle input:

```
0.0 ready
10.0 not_ready
20.0 ready
35.0 not_ready
45.0 ready
60.0 not_ready
70.0 ready
```

Centroid input:

```
0.0 ready
5.0 not_ready
20.0 ready
30.0 not_ready
45.0 ready
55.0 not_ready
70.0 ready
```

… same for the remaining three cell structures.

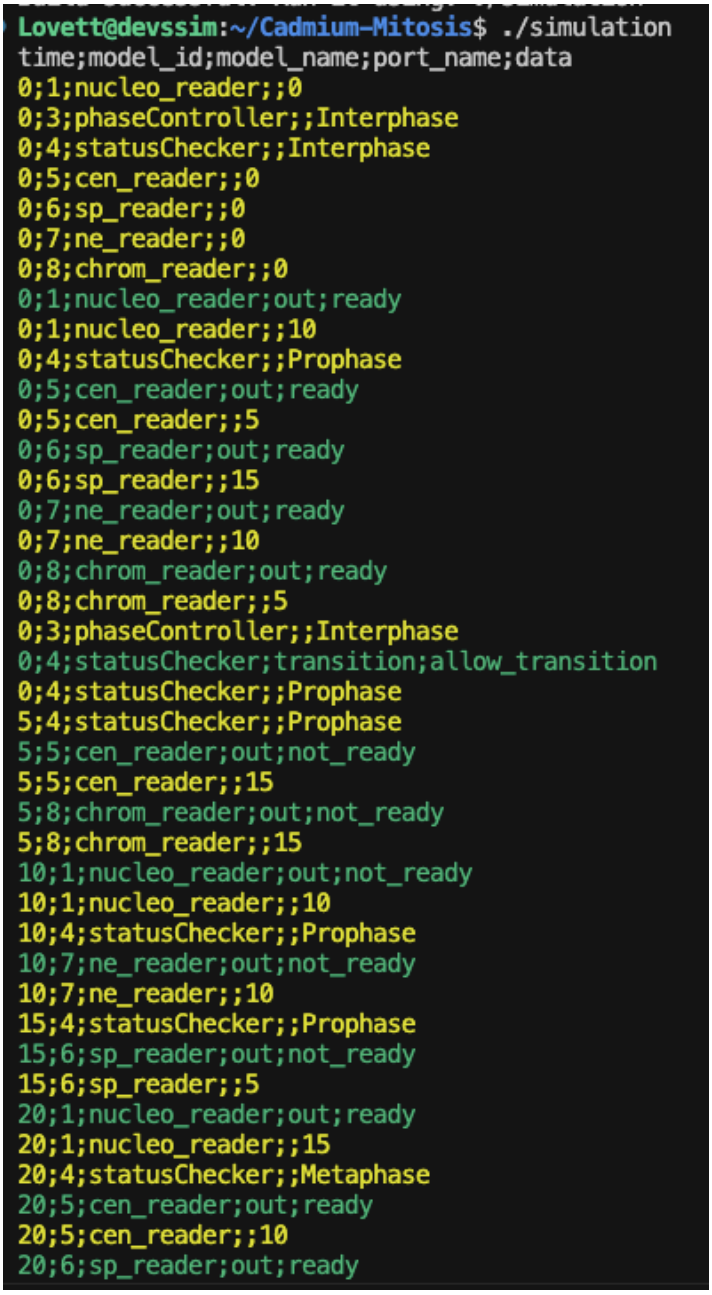


Figure 11: PhaseControl Simulation Results

At t = 0s, the system initializes in Interphase. The PhaseControl outputs Interphase, and the statusChecker is also in the Interphase state. All structure readers (nucleo_reader, cen_reader, sp_reader, ne_reader, and chrom_reader) initially report 0, then output “ready” to show that during the Interphase all cellular structures satisfy the readiness condition. Once they are all ready, the statusChecker produces a transition;allow_transition event. This enables the

phaseController to move forward. At t = 10s , structures report not_ready which happens while the cell components are progressing. The statusChecker remains in Prophase since ther readiness condition is still false. At t = 20s, all cell components are “ready” again and the statusChecker transitions to the Metaphase, confirming that all the conditions defined in the formal specification are satisfied.

CellStructures (Coupled Model)

The CellStructures model is composed of four cell part atomic models, and the coupled centrosome model: CentrosomePair, Chromosome, NuclearEnvelope, Nucleolus, and Spindle. It takes its input from the PhaseControl output as the “phase_event” or the name of the phase the model is simulating, and outputs “ready” for each Atomic part that finishes its internal transition.

Input sequence example:

```
0.0 Interphase
```

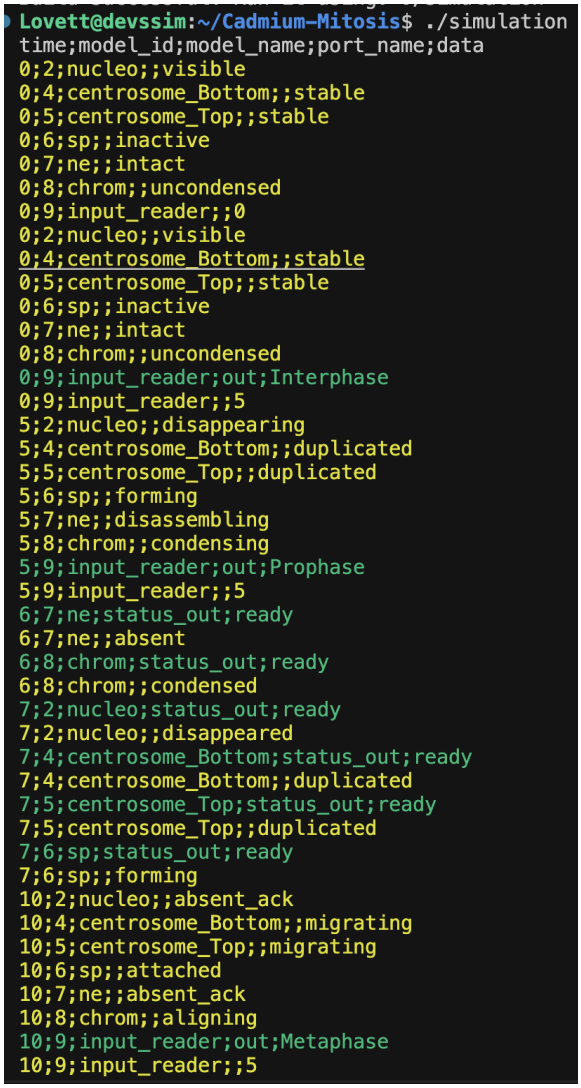


Figure 12: CellStructures Simulation Results

As observed in the execution log, at t=0s the CellStructures model receives the initial phase input (Interphase). All subcomponents initialize in their starting biological states where chromosomes are uncondensed, centrosomes are stale, spindle is inactive, and the nuclear envelope is intact. As the structures progress through the phases, the state of their biological states changes. Namely at t = 5s, Prophase is received and the changes begin. The nucleolus transitions to disappearing, centrosomes move to duplicated, the spindle begins forming, and chromosomes begin condensing. And so on and so forth through the rest of the timing, where each phase corresponds to some kind of change or resting state for each subcomponent of the cell.

Analysis of Output of Top Model

The top model runs a single simulation on a fixed time horizon (t = 100), with multiple atomic models in parallel. Each atomic model is driven by an IEStream reader that injects phase events from the same input files used in atomic atomic test cases. The coordinator advances simulation time and processes internal and external events for all components together, so the global log is one timeline with events from every model interwoven by time.

```
Qureshi@devssim:~/Cadmium-Mitosis$ ./build_sim.sh
Build successful! Run it using: ./simulation
Qureshi@devssim:~/Cadmium-Mitosis$ ./simulation
time;model_id;model_name;port_name;data
0;3;nucleo;;visible
0;5;centrosome_Bottom;;stable
0;6;centrosome_Top;;stable
0;7;sp;;inactive
0;8;ne;;intact
0;9;chrom;;uncondensed
0;11;phaseController;;Interphase
0;12;statusChecker;;Interphase
0;13;start_reader;;1
1;12;statusChecker;;Prophase
1;13;start_reader;out;"start"
1;13;start_reader;;inf
1;11;phaseController;;Prophase
1;12;statusChecker;transition;allow_transition
1;12;statusChecker;;Prophase
1;3;nucleo;;disappearing
1;5;centrosome_Bottom;;duplicated
1;6;centrosome_Top;;duplicated
1;7;sp;;forming
1;8;ne;;disassembling
1;9;chrom;;condensing
1;11;phaseController;phase_event;Prophase
1;11;phaseController;;Prophase
2;8;ne;status_out;ready
2;8;ne;;absent
2;9;chrom;status_out;ready
2;9;chrom;;condensed
2;12;statusChecker;;Prophase
3;3;nucleo;status_out;ready
3;3;nucleo;;disappeared
3;5;centrosome_Bottom;status_out;ready
3;5;centrosome_Bottom;;duplicated
3;6;centrosome_Top;status_out;ready
3;6;centrosome_Top;;duplicated
3;7;sp;status_out;ready
3;7;sp;;forming
3;12;statusChecker;;Metaphase
3;11;phaseController;;Metaphase
3;12;statusChecker;transition;allow_transition
3;12;statusChecker;;Metaphase
3;3;nucleo;;absent_ack
3;5;centrosome_Bottom;;migrating
3;6;centrosome_Top;;migrating
3;7;sp;;attached
3;8;ne;;absent_ack
Qureshi@devssim:~/Cadmium-Mitosis$ ./simulation
3;8;ne;;absent_ack
3;9;chrom;;aligning
3;11;phaseController;phase_event;Metaphase
3;11;phaseController;;Metaphase
3.1;3;nucleo;status_out;ready
3.1;3;nucleo;;disappeared
3.1;8;ne;status_out;ready
3.1;8;ne;;absent
3.1;12;statusChecker;;Metaphase
4;9;chrom;status_out;ready
4;9;chrom;;aligned
4;12;statusChecker;;Metaphase
6;5;centrosome_Bottom;status_out;ready
6;5;centrosome_Bottom;;migrating
6;6;centrosome_Top;status_out;ready
6;6;centrosome_Top;;migrating
6;7;sp;status_out;ready
6;7;sp;;attached
6;12;statusChecker;;Anaphase
6;11;phaseController;;Anaphase
6;12;statusChecker;transition;allow_transition
6;12;statusChecker;;Anaphase
6;3;nucleo;;absent_ack
6;5;centrosome_Bottom;;at_poles
6;6;centrosome_Top;;at_poles
6;7;sp;;pulling
6;8;ne;;absent_ack
6;9;chrom;;separating
6;11;phaseController;phase_event;Anaphase
6;11;phaseController;;Anaphase
6;5;centrosome_Bottom;status_out;ready
6;5;centrosome_Bottom;;at_poles
6;6;centrosome_Top;status_out;ready
6;6;centrosome_Top;;at_poles
6;12;statusChecker;;Anaphase
6.1;3;nucleo;status_out;ready
6.1;3;nucleo;;disappeared
6.1;8;ne;status_out;ready
6.1;8;ne;;absent
6.1;12;statusChecker;;Anaphase
7;9;chrom;status_out;ready
7;9;chrom;;separated
7;12;statusChecker;;Anaphase
9;7;sp;status_out;ready
9;7;sp;;pulling
9;12;statusChecker;;Telophase
9;11;phaseController;;Telophase
9;12;statusChecker;transition;allow_transition
```

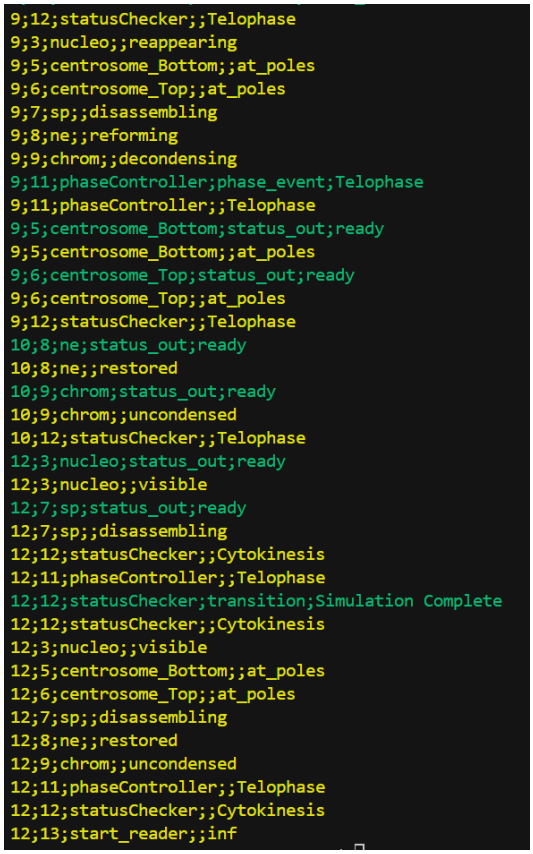


Figure 13: Final Mitosis System Simulation Results

From the three above images, we can see the simulation results of the entire mitosis system. As discussed earlier, the stages of mitosis are interphase, prophase, metaphase, anaphase, telophase and cytokinesis. The system is initialized with interphase and once it reads the input.txt file to start simulation. Then the phase controller and status checker will move to the next phase, prophase. Once in prophase, the atomic models will do their external transition and time advance and output ready once they are done. This will move the system to metaphase, and repeat the same process and will move on to the next phase when all models in cell structures are ready. Once it reaches the Cytokinesis phase, the status checker will print out "Simulation Complete" and the status of all the models will print out which is their last state. From this simulation, we can see that atomic and coupled models are able to work together in the DEVS framework. It displayed the phases of the mitosis system and states for each part of the cell as the phases progressed.

Conclusion

In conclusion, applying the DEVS (Discrete Event System Specification) framework to mitosis was an invaluable learning experience for the team as it was able to show how theoretical system design can be used to model biological complexity. We were able to validate the output behavior involved comparing the simulation's event log against experiment data to ensure internal transition and time advance were producing biologically accurate results.