Process Algebra with Layers: Multi-scale Integration Modelling applied to Cancer Therapy

Erin Scott[1](0000-0001-7190-1728), James Nicol[2], Jonathan Coulter[2](0000-0003-4112-2357), Andrew Hoyle[1](0000-0002-9117-7041), and Carron Shankland[1](0000-0001-7672-2884)(✉)

[1] Computing Science and Mathematics
University of Stirling, Stirling, FK9 4LA, UK
ces@cs.stir.ac.uk
[2] School of Pharmacy
Queen's University Belfast, BT7 1NN, UK

Abstract. We present a novel Process Algebra designed for multi-scale integration modelling: Process Algebra with Layers (PAL). The unique feature of PAL is the modularisation of scale into integrated layers: Object and Population. An Object can represent a molecule, organelle, cell, tissue, organ or any organism. Populations hold specific types of Object, for example, life stages, cell phases and infectious states. The syntax and semantics of this novel language are presented. A PAL model of the multi-scale system of cell growth and damage from cancer treatment is given. This model allows the analysis of different scales of the system. The Object and Population levels give insight into the length of a cell cycle and cell population growth respectively. The PAL model results are compared to wet laboratory survival fractions of cells given different doses of radiation treatment [1]. This comparison shows how PAL can be used to aid in investigations of cancer treatment in systems biology.

Keywords: Systems Biology, Formal Methods, Mathematical Modelling, Cell Cycle, DNA Damage

1 Scientific Background

Multi-scale modelling in systems biology is now commonplace and indeed essential to many investigations [2, 3]. Analysis of emergent properties arising from the interactions between scales of multi-scale systems is important as an aid in solutions to topical issues such as disease and climate change. Indeed the issue of cancer cell growth and damage from treatments is a multi-scale system as the damage affects the levels of intracellular proteins within a cell and therefore affects the cell population levels [1, 4]. There are a number of successful multi-scale models such as Powathil et al. [4] cellular automaton model to study the dynamics of chemotherapy drugs to cancer cell-cycle heterogeneity. Another example is the Met Office climate prediction model [5] which couples the atmospheric and oceanic scales together in one model.

There is no universally adopted theoretical/computational framework or language for the construction of multi-scale models. Most multi-scale models are specific to the problem they are addressing and are defined by integrated scales that are modelled in different mathematical and computational languages [2, 3]. These hybrid models use a combination of modelling approaches such as Ordinary Differential Equations (ODE) and Cellular automata (CA) to define specific scales of the model. For example, the Powathil et al. [4] model is hybrid because the intracellular proteins are defined in ODE and the cell populations are modelled in cellular automaton. The Met Office climate prediction model [5] is hybrid as it utilises different mathematical approaches to describe the different scales and is specific to the problem of climate change forecasting. These hybrid models make the structure and the analysis of the model difficult as the scales are defined in separate models. The modeller must create, or be knowledgeable in, integration techniques to link the models together.

Process algebra offers an ideal opportunity in systems biology [6]. It gives a high-level description of interactions, communications, and synchronizations between a collection of independent agents or processes. Its application provides many analysis techniques for systems' behaviour and properties. For example, time series simulations (to produce model predictions to compare with observed data), Markovian analysis (deriving a Continuous Time Markov Chain (CTMC) of all the possible states and evolutions of the model to be used for functional verification), model checking (to validate the model against a high level property specified in e.g. temporal logic), and model generation (creating models from time series data). The multiscale P-system framework of Romero-Campero et al [7] is also attractive for its ability to describe extremely abstract hierarchical systems within one formalism; however, the range of possible analyses is smaller.

There are only a few process algebra languages that are specifically designed for multi-scale systems. These include Parametric Stochastic Process Algebra with Hooks (psPAH) [8] and Performance Evaluation Process Algebra nets (PEPA nets) [9]. These multi-scale languages focus on the integration of spatial scales, assuming the same time scale. One important multi-scale modelling feature is allowing the easy definition of the addition and deletion of objects within the language to capture, for example, cell division and cell death. Neither psPAH nor PEPA nets include this specific modelling feature. As a result, the modeller needs to add many lines of code to add and delete objects, making the model difficult to construct and read. In this paper we propose a novel language, Process Algebra with Layers (PAL), which gives a convenient representation of multi-scale systems by putting these features directly into the syntax and semantics.

The unique features of PAL are the integrated layers: Object and Population. These novel layers allow PAL to include the easy definition of the addition and deletion of Objects through Population actions. These layers modularise the definitions of specific Object populations from the Object's internal system definition giving a more elegant model. The layers are generally applicable to a variety of multi-scale systems: see Section 2.1. We show how easily the layers of PAL can be applied to a novel mammalian cell cycle and DNA damage case study. The case study has two distinct layers. At one level, there is a population of growing and dividing cells. How cells move between these states is controlled by changes in cell mass and levels of selected proteins. The intracellular species and interactions between them form the second layer. The novel PAL model links together the established models of Zhang et al. [10] and Tyson et al. [11] to investigate the effect of DNA damage from radiation on the progression of the cell cycle. This has not been previously considered in the literature.

2 Materials and Methods

2.1 Process Algebra with Layers

PAL has two layers which are named Object and Population. Figure 1 shows a conceptual schematic of these layers.

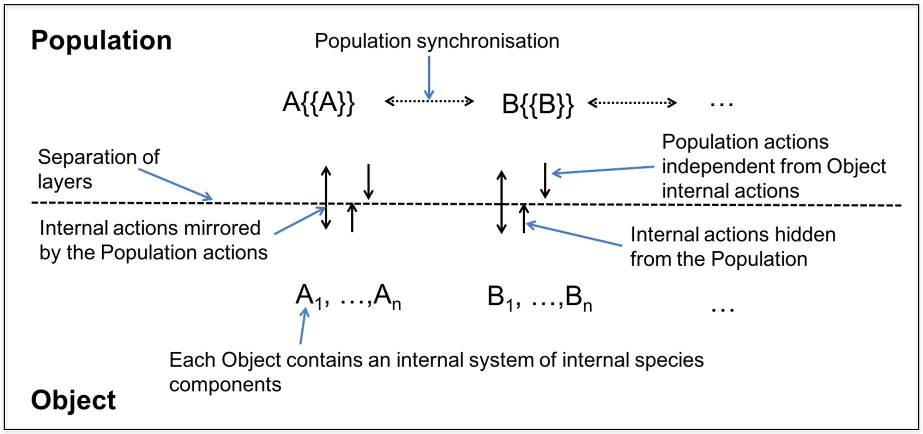
Population
Population synchronisation
A{{A}}
B{{B}}
…
Separation of layers
Population actions independent from Object internal actions
Internal actions mirrored by the Population actions
Internal actions hidden from the Population
$A_1, \ldots, A_n$
$B_1, \ldots, B_n$
…
Each Object contains an internal system of internal species components
Object

Fig. 1. Schematic of Object and Population Layers. The Object layer describes each specific Object. Objects have an internal system description of internal species components. The Population layer defines populations of Objects.

An Object is an individual system model at the lowest scale of interest. Objects comprise a number of internal species components. A PAL model may have multiple Object types, and multiple replications of these Objects. The Objects' internal species evolve dynamically over time via actions. Some actions are internal to the Object, and some impact on the next scale up, the Population. A Population is a collection of Objects. A PAL model must have at least one Population. There may be interaction between Populations and between layers.

Objects and Populations can stand for any scale the modeller chooses. In Section 2.5, for example, PAL is illustrated by a mammalian cell case study. The cells and their intracellular proteins (internal species) are the Objects. These drive the cell cycle. Populations define cells in specific phases such as growing and dividing. Mirrored actions connect Objects (e.g. when a protein reaches a specific level) with Populations (e.g. resulting in a state change from growing to dividing). Another example, from Chapter 5 of Scott [12], takes marine organisms as Objects, with their physiology in energy budgets as internal species. The organisms are held in Populations describing their life stage such as larvae, juvenile and adult. The organisms change life stage through mirrored actions (e.g. when mass reaches a certain threshold, larvae evolve to juveniles) and also can be removed from the system (e.g. fishing of the Population). Also at the organism level, we could consider a model of disease hosts (Objects) and their immune system and micro-parasite interaction (internal species). Populations of hosts are defined as susceptible, infected and recovered. Hosts change infectious states dependent on the number of parasites in their system (mirrored actions) and more hosts can be added due to immigration (independent Population action).

2.2 The Syntax of PAL

$$\begin{aligned}
P\{\{O\}\}_A &::= (\alpha, 1)\ PAL_{op}\ P\{\{O\}\}_A \mid P\{\{O\}\}_A\ +\ P\{\{O\}\}_A \mid D \\
\text{Where } PAL_{op} &= \downarrow\ \mid\ \uparrow\ \mid ((+)) \\
O &::= O\ \underset{L}{\bowtie}\ O \mid S(x) \\
S &::= (\alpha, \kappa)\ \mathit{op}\ S \mid S + S \mid C \\
\text{Where } \mathit{op} &= <<\ \mid\ >>\ \mid (+) \mid (-) \mid (\ .\ ) \\
M &::= M\ \underset{L_s}{\diamondsuit}\ M \mid P\{\{O\}\}_A
\end{aligned}$$

Fig. 2. Syntax of PAL.

The syntax of PAL is shown in Figure 2. PAL uses the same syntax as Biochemical-Performance Evaluation Process Algebra (Bio-PEPA) [13] to define the internal species components.

The component O, called an **Object component**, describes an internal system and the interactions among internal species components S. Species are named using $C = S$ to allow modular construction. The element x is a positive integer-valued parameter. Constants allow names to be assigned to patterns of behaviour associated with components. $(\alpha,\ \kappa)$ is the internal species prefix, where $\alpha \in$ SpeciesActions is the action type, k is the stoichiometry coefficient of the species in that reaction, and SpeciesActions is a modeller-defined set of action names. The prefix combinators *op* are: $<<$ indicating a reactant, $>>$ a product, $(+)$ an activator, $(-)$ an inhibitor and $(\ .\ )$ a generic modifier. $O\ \underset{L}{\bowtie}\ O$ denotes the cooperation between internal species over the cooperation set L. Set L determines those activities on which the cooperands are forced to synchronise.

The component $P\{\{O\}\}_A$ is called a **Population component** and represents a multi-set of Object components $P\{\{O\}\}_A = P\{[\![O_1, ..., O_n]\!]\}_A$. A multi-set is an unordered collection of Objects with repetitions. Populations are named using $D = P\{\{O\}\}_A$ to allow modular construction. $(\alpha,1)$ is the prefix, where α

$\in$ Actions is the action type and 1 is the stoichiometric coefficient of the Object in that action. The design choice of a stoichiometry of 1 was chosen to simplify the resulting states the action produces. There are three prefix combinators called PAL_{op} which represent the role of the Objects in the action. These are: $\downarrow$ indicates a deletion of an Object, $\uparrow$ an addition of an Object (an *initialO* element will be added to a specific Population with a specific initial set up relevant to its Population) and $((+))$ an Object which is an activator (the Object is involved in the action but does not change).

The choice operator $P\{\{O\}\}_A + P\{\{O\}\}_A$ and $S + S$ represents non-deterministic choice between actions whether these be Population actions or internal species actions. Once one branch is chosen the others are discarded, thus choice represents competition between actions depending on their rate.

The top-level component M, called a model component, describes the system and the interactions among Population components. The cooperation between Populations over the multi-scale action cooperation set L_s is expressed by $M \underset{L_s}{\diamondsuit} M$. Set L_s determines those actions on which the cooperands must synchronise. Each Population component must have a hidden action set A identifying internal species actions which are hidden from the Population component. Hidden actions should not be in the set L_s in a well defined PAL system.

2.3 The Semantics of PAL

A PAL system $\mathcal{P}$ is a septuple $\langle Pcomp, Ocomp, Scomp, F_R, \mathcal{K}, \mathcal{N}, \mathcal{M}\rangle$, where:

- *Pcomp* is the set of definitions of Population components;
- *Ocomp* is the set of definitions of Object components;
- *Scomp* is the set of definitions of internal species components;
- F_R is the set of functional rate definitions;
- $\mathcal{K}$ is the set of parameter definitions;
- $\mathcal{N}$ is the set of quantities describing each internal species;
- $\mathcal{M}$ is the model component describing the system.

The definition of the Object components in *Ocomp* must be defined in terms of the internal species components defined in *Scomp* and for each cooperation set L_i in O, $L_i \subseteq$ SpeciesActions (O). In a well-defined PAL system each element has to satisfy the following conditions. Set $\mathcal{N}$ has to contain all the internal species components. The functional rates are well defined if each variable in their definition refers to the name of a species component in the set $\mathcal{N}$ or a constant parameter in the set $\mathcal{K}$. The definition of the internal species components in *Scomp* must have sub-terms of the form (α, k)op S and the action types in each single component must be distinct. The definition of the Population components *Pcomp* must be defined in sub-terms of the form $(\alpha,1)$ PAL_{op} $P\{\{O\}\}_A$ and the action types in each single component must be distinct. The model component $\mathcal{M}$ must be defined in terms of the Population components defined in *Pcomp* and for each cooperation set L_{si} in $\mathcal{M}$, $L_{si} \subseteq$ Actions $(\mathcal{M})$.

prefixReac $((\alpha, k) << S)(l) \xrightarrow{(\alpha,\ [S :<< (l, \kappa)])} S(l - k) \quad k \leq l \leq N$

prefixProd $((\alpha, k) >> S)(l) \xrightarrow{(\alpha,\ [S :>> (l, \kappa)])} S(l + k) \quad 0 \leq l \leq (N - k)$

prefixMod $((\alpha, k) op\ S)(l) \xrightarrow{(\alpha,\ [S\ :\ op(l, \kappa)])} S(l)$ with op = (.), (+), (-) and $0 < l \leq N$ if $op = (+)$, $0 \leq l \leq N$ otherwise

where S is the name of the species component, op is the action type, l the level κ the stoichiometry coefficient, and N the maximum level of S.

choice1 $\dfrac{S_1(l) \xrightarrow{(\alpha,w)} S_1'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)} S_1'(l')}$ choice2 $\dfrac{S_2(l) \xrightarrow{(\alpha,w)} S_2'(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,w)} S_2'(l')}$

constant $\dfrac{S(l) \xrightarrow{(\alpha, S\text{: } [op(l,k)])} S'(l')}{C(L) \xrightarrow{(\alpha, C\text{: } [op(l,k)])} S'(l')}$ with $C = S$

coop1 $\dfrac{O_1 \xrightarrow{(\alpha,w)} O_1'}{O_1 \underset{L}{\bowtie} O_2 \xrightarrow{(\alpha,w)} O_1' \underset{L}{\bowtie} O_2}$ with $\alpha \notin L$

coop2 $\dfrac{O_2 \xrightarrow{(\alpha,w)} O_2'}{O_1 \underset{L}{\bowtie} O_2 \xrightarrow{(\alpha,w)} O_1 \underset{L}{\bowtie} O_2'}$ with $\alpha \notin L$

coop3 $\dfrac{O_1 \xrightarrow{(\alpha,w)} O_1' \quad O_2 \xrightarrow{(\alpha,w)} O_2'}{O_1 \underset{L}{\bowtie} O_2 \xrightarrow{(\alpha,w)} O_1' \underset{L}{\bowtie} O_2'}$ with $\alpha \in L$

where w is a list recording the species that participate in the reaction and L is the cooperation action set

Fig. 3. Rules for Bio-PEPA included in the semantics of PAL. These rules are presented in Ciocchetta et al. [13] and are repeated here for convenience and completeness.

The rules of PAL specify Population behaviour and its relation to Object behaviour. The semantics of an Object are as in Bio-PEPA, and repeated here in Figure 3 for convenience. Figures 4 and 5 describe how Objects and Populations influence each other and how Populations evolve, respectively. These rules collectively allow a CTMC to be defined from a PAL model.

Some Object actions are hidden from the Population level as defined in the Action Hidden Rule in Figure 4. The modeller defines a set of hidden actions A when describing a model in PAL. These could include actions such as synthesis and degradation of intracellular proteins within a cell. Actions such as these do not change the Population layer composition of the system, therefore do not need to be mirrored by the Population. For example, transitions of cell proteins indirectly affect mass but are hidden from the Population layer.

Object actions that are mirrored by the Population are defined by the Action Mirror Rule in Figure 4. For example, in the cell model, the changing mass

Action Mirror/Hidden Rules

Internal Actions that are mirrored by Populations

$$\frac{O_i \xrightarrow{(\alpha,w)} O_i'}{P\{\{O\}\}_A \xrightarrow{(\alpha,w)} P\{\{O'\}\}_A} \; where\ \exists\ O_i \in O \ \wedge \ \alpha \notin A$$

$$where\ O' = O \ \oplus \ O_i' \wedge P\{\{O\}\} \xrightarrow{(\alpha,w)} P\{\{O'\}\}$$

Internal Actions that are hidden from Populations

$$\frac{O_i \xrightarrow{(\alpha,w)} O_i'}{P\{\{O\}\}_A \xrightarrow{\tau} P\{\{O'\}\}_A} \; where\ \exists\ O_i \in O \ \wedge \ \alpha \in A$$

$$where\ O' = O \ \oplus \ O_i'$$

where $\oplus$ overwrites $O_i\ in\ O$ with new O_i' state, leaving the rest of O unchanged.

Fig. 4. Semantics of PAL: Action Mirror/Hidden Rules.

of the cell (an internal species) will trigger the transition of the cell from the growing state to the dividing state. The changing internal action in this case has an impact on the Population view of the system. These internal actions are mirrored by Population actions which are defined by the Prefix Population Transition Rules shown at the top of Figure 5.

There are essentially three Prefix Population Transition Rules: adding, deleting and activator. The deletion rule has two variants depending on whether the deletion is initiated from the Object or the Population level. These rules are asymmetric because when deleting an Object from a Population the rule needs to identify the specific Object that is to be deleted. For example, the deletion rule can be used for cell phase transitions and deaths of specific cells, therefore the cell must be known to the rule so that the correct cell is deleted. In the case of addition and the activator rule a specific Object does not need to be known. For example, in the addition rule a new initialisation of an Object is added to a Population. The Object in the activator rule does not need to be known by the rule as the rule does not change the Object. In the case study here this would be a dividing cell becoming two growing cells.

Lastly, potential interactions at the Population level are dealt with by the Population Transition Rules (see lower section of Figure 5). Populations can perform actions autonomously and this allows actions such as death from a Population action to be defined in a model. This feature is not used in the case study, but could be added if, for example, chemical messengers from a dying cell influenced death in another cell. Populations synchronise/communicate on specific external actions as defined in the Population Transition Rules shown at

Prefix Population Transition Rules

Adding an Object to a Population

$((\alpha, 1) \uparrow P\{\{O\}\}_A) \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A \ \mathit{where}\ \alpha \notin A$

$O' = O \cup \llbracket \mathit{initialO} \rrbracket$

Deleting a specific Object from a Population

$((\alpha, 1) \downarrow P\{\{O\}\}_A) \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A \ \mathit{where}\ \alpha \notin A \ \wedge \ \alpha \in \text{SpeciesActions}(O)$

$\exists i.\ O_i \in O \wedge O_i \xrightarrow{(\alpha, w)} O_i' \wedge O = \llbracket O_1, ..., O_n \rrbracket \wedge |O| \geq 1 \wedge O' = O \backslash \llbracket O_i \rrbracket$

Activator does not increase or decrease a Population

$((\alpha, 1)((+))P\{\{O\}\}_A) \xrightarrow{(\alpha, w)} P\{\{O\}\}_A \ \mathit{where}\ \alpha \notin A$

$O = \llbracket O_1, ..., O_n \rrbracket \wedge |O| \geq 1$

Deleting a random Object from a Population

$((\alpha, 1) \downarrow P\{\{O\}\}_A) \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A \ \mathit{where}\ \alpha \notin A \ \wedge \ \alpha \notin \text{SpeciesActions}(O)$

$\exists i.\ O_i \in O \wedge O = \llbracket O_1, ..., O_n \rrbracket \wedge |O| \geq 1 \wedge O' = O \backslash \llbracket O_i \rrbracket$

where (α, w) comes from synchronising with another Population
or from the Object layer.

Population Transition Rules

Constant

$$\frac{P\{\{O\}\}_A \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A}{D \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A} \ \mathit{where}\ D = P\{\{O\}\}_A$$

Asynchronous Left

$$\frac{P\{\{O\}\}_A \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A}{P\{\{O\}\}_A \underset{L_s}{\Diamond} M \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A \underset{L_s}{\Diamond} M} \ \mathit{where}\ \alpha \notin L_s$$

Asynchronous Right

$$\frac{P\{\{O\}\}_A \xrightarrow{(\alpha, w)} P\{\{O'\}\}_A}{M \underset{L_s}{\Diamond} P\{\{O\}\}_A \xrightarrow{(\alpha, w)} M \underset{L_s}{\Diamond} P\{\{O'\}\}_A} \ \mathit{where}\ \alpha \notin L_s$$

Population Synchronisation

$$\frac{P_1\{\{O_1\}\}_A \xrightarrow{(\alpha, w)} P_1\{\{O_1'\}\}_A \quad P_2\{\{O_2\}\}_A \xrightarrow{(\alpha, w)} P_2\{\{O_2'\}\}_A}{P_1\{\{O_1\}\}_A \underset{L_s}{\Diamond} P_2\{\{O_2\}\}_A \xrightarrow{(\alpha, w)} P_1\{\{O_1'\}\}_A \underset{L_s}{\Diamond} P_2\{\{O_2'\}\}_A}$$

$\mathit{where}\ \alpha \in L_s$

where L_s is the multi-scale synchronisation action set

Fig. 5. Semantics of PAL: Prefix Population Transition Rules and Population Transition Rules.

the base of Figure 5. This allows the definition of cell phase transitions and cell division as these actions involve two Populations changing in number. For more detail on PAL see Chapter 4, pp.74-80 of Scott [12].

2.4 Case Study

PAL is applied to a mammalian cell cycle and DNA damage case study to illustrate its capabilities in systems biology. A cell cycle is the series of events that take place in a cell leading to its division. The motivation of this case study is to analyse the effects of damage from radiation treatments to the length of a cancer cell cycle and cell survival. A PAL model has been created by linking together an established cell cycle model from Tyson et al. [11] with a repair model with an external force applying damage by Zhang et al. [10]. Other models such as Powathil et al. [4, 14] and Guerrero et al. [15] use the Tyson et al. [11] model as a basis for cell cycle transitions and regulation. Zhang et al. [10] presents a number of potential models for the transcription factor p53 activity observed experimentally in response to DNA damage. p53 is at the centre of a number of DNA damage responses which interact downstream with the regulation of the cell cycle. The linking of the Zhang et al. [10] model with the Tyson et al. [11] model allows the creation of a novel model investigating the effects of DNA damage from radiation treatments on the species affecting progression of the cell cycle and consequent effect on cell colonies. This has not been previously considered in the literature. The novel PAL model allows multi-scale analysis, including Object layer experimentation (average length of a single cell cycle, Section 3.1) and Population layer experimentation (cell population growth, Section 3.2).

2.5 PAL Model

Although PAL makes describing the model simpler than other multi-scale techniques, the model is too long to be shown here. See Chapter 6, pp.109-112 of Scott [12]. The description and results of experimentation are given here. The case study has two distinct layers: cell population and intracellular. The cell Population layer is described in the PAL model by defining two PAL Populations based on the two steady states of a cell: Growing and Dividing. These Populations contain *G cell* and *D cell* Object components, illustrated in Figure 6 by a Growing cell becoming a Dividing cell, and that in turn becoming two Growing cells.

In the intracellular layer, G and D cells contain internal species which are the cell mass and proteins translated from Tyson et al. [11] and Zhang et al. [10]. These proteins include the Cdk-cyclin B complex (CycB), the APC-Cdh1 complex (Cdh1), the active form of Cdc20 (Cdc20A), the total Cdc20 (Cdc20T) and the intermediary enzyme (IEP). These are shown as the internal species in Figure 6. The graphic shows how the proteins rise and fall in response to each other, creating the conditions of the cell cycle. Note the black label in the graphic indicating Growing phase or Dividing phase.

Transitions between the two Populations are controlled by changes in cell mass and threshold values of the CycB, indicated on the arrows in Figure 6. To make the cell cycle relevant to mammalian cells the parameter values of this model are taken from Powathil et al. [4], therefore, time in the model is in hours.

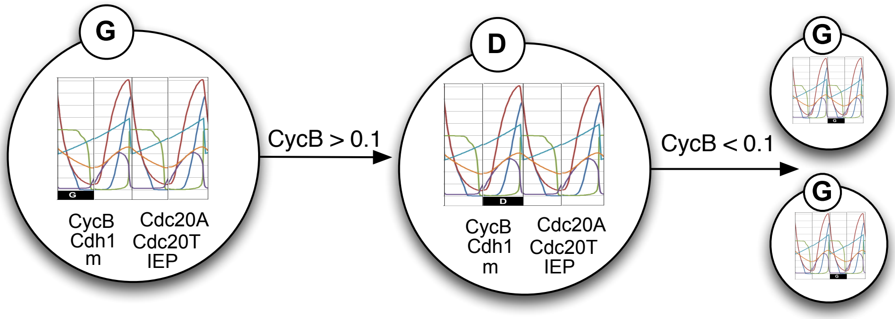

Fig. 6. Example of a single G cell evolving, through its internal species, to a D cell, and then to two G cells.

To model how the cell cycle proteins are affected by DNA damage a ODE model originally developed by Zhang et al. [10] is translated into internal species and parameters of the PAL model. The model consists of DNA damage caused by radiation treatment, the p53 and Mdm2 (nucleus and cytoplasmic) that promote the degradation of p53. p53 inhibits the activity of CycB preventing the progression of the cell cycle. In the PAL model the Tyson et al. [11] and Zhang et al. [10] models are innovatively linked together by changing the CycB degradation rate to be influenced by changes in the p53 levels. Levels of p53 are at equilibrium when there is no damage in the system. When there is damage it causes p53 levels to pulse according to Zhang et al. [10]. Damage is a parameter of the model and Zhang et al. [10] states the simple assumption is made that damage is repaired at a constant rate.

Damage is not uniform: although the whole plate of cells has the same treatment, radiation at lower levels will hit some cells but not others. To model this, different damage levels are assigned to each cell in the colony depending on the highest level of damage at the start of the simulation. For example, if the highest damage in a simulation is five the cells are assigned damages in the range of zero to five on a random distribution. Damage occurs immediately in all simulation experiments and ranges from 0-12 (integer values). We assume damage greater than four causes cell death through the Population actions. Cell cycle length is impacted with damage of four or less by the increase of the CycB degradation rate.

The investigation of the effects of damage in the PAL model from the average length of a cell cycle (Object layer) to population growth (Population layer) was carried out. The damage in the model is abstract, therefore matching the real notion of damage from radiation was achieved by the creation of a relationship function between the abstract damage and Gy, $damage/2$ = Gy dose, chosen here to fit with experimental results, see Section 3.3. This is why damage ranges from 0-12, to fit with 0 to 6 Gy.

3 Results

A PAL model parser has been implemented and translates a PAL model into a Bio-PEPA model to allow analysis of the model in the Bio-PEPA Eclipse plug-in [13, 16]. This is a complete development environment for Bio-PEPA models, with editing, simulation, experimentation, model checking, and export to SBML [17]. The parser source code can be found in the following repository [3]. A range of experiments were carried out, allowing analysis at the Object level (cell components, Section 3.1) and Population level (groups of cells driven by their internal mechanisms, Section 3.2).

3.1 Object Layer Experiments: Analysis of Average Length of a Cell Cycle

Simulation distribution analysis was undertaken to analyse the average length of the cell cycle and the impact of increasing the amount of damage. This analysis takes place at the individual cell scale (Object layer). Simulation distributions obtain the percentage of a user-defined number of stochastic simulations for which some property is true at or before a given time t. The Bio-PEPA plug-in plots the Cumulative Distribution Function (CDF) and Probability Distribution Function (PDF) of any agents in the model, with respect to the target value.

Table 1. Average length of cell cycle and 95% confidence interval in hours of each simulation distribution experiment.

Experiment	Average cycle	Confidence Interval (95%)
Control	23.96	(23.51, 24.41)
Damage 1	24.18	(23.70, 24.66)
Damage 2	24.73	(24.16, 25.30)
Damage 3	24.74	(24.09, 25.39)
Damage 4	25.88	(25.19, 26.57)

Five experiments were carried out (damage 0-4, i.e. no cell death), see Table 1 for results. As this analysis is observing one cell cycle, the PAL model starts with one G cell with the specified level of damage. The chosen component in this analysis is an agent which tracks a cell's completion of one cell cycle. The number of stochastic simulation replications is 200 and the stop time is 48 hours. The computation time for each experiment was approximately 3 hours (MacOS X Yosemite version 10.10.5, 2.2 GHz Intel core i7, 16GB 1600MHz DDR3).

All simulations completed a cell cycle before the stop time of 48 hours. The results show that damage from one to three does not significantly affect the cell cycle average length. The intracellular proteins can cope with these damage levels. The average cell cycle length increases when damage of four is applied.

[3] PAL Parser source code: https://github.com/MissErinScott/PAL-Parser

3.2 Population Layer Experiments: Analysis of Cell Population Growth

Discrete stochastic simulation time-series analysis was carried out to analyse cell growth over a longer period and the effects of damage on a colony of cells. This analysis takes place at the colony scale (Population layer). The initial population was eight G cells (due to population limitations of Bio-PEPA plug-in [13]). Different damage levels are assigned to each cell randomly depending on the highest level of damage at the start of the simulation. Cells are simulated from 0 - 64 hours. Throughout the time period of the simulation, new cells will be assigned different damage levels based on a constant repair rate. Experiments were carried out with damage ranging from 0 (control) to 12. Results from one replication are presented in Figure 7 which shows the total population growth of G and D cells at different damage levels. Each experiment is one stochastic simulation which had a computation time of 3 to 20 minutes dependent on damage and a further 15 minutes for manual processing. Four replications of these stochastic simulations were carried out for damage values 0 to 6, 9 and 12.

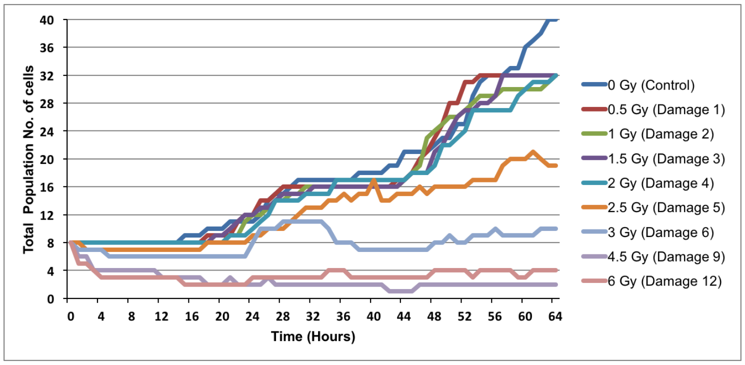

Fig. 7. Single replication of total cell Population Growth at different damage levels.

The results show that population growth is not affected by damage from one to four, as expected. These results reinforce the simulation distribution results which show the cell cycle is not affected by this lower damage range. The damage in each experiment is repaired at a constant rate, therefore the population of cells starts to recover, dependent on the damage assigned. Death occurred in the experiments where damage is above four. This feature had the effect of reducing the population substantially. The populations assigned the damage levels of nine and twelve had a greater reduction as they had the greater proportion of cells assigned a damage above four.

3.3 Comparison with Wet Laboratory Survival Fraction Results

The population results of the PAL model are compared to wet laboratory survival fraction results of cells given different Gy doses of radiation treatment [1]. This data is shown in Figure 8 (red squares) and was generated using Clonogenic survival assays following treatment with 0-6 Gray (Gy) 160 kVp x-rays as previously described by Butterworth et al. [18]. The survival fractions were calculated as the plating efficiency of the treated group divided by the plating efficiency of the untreated control cells, with error bars representing the standard deviation (SD) (n = 7). The wet laboratory experiments had duration of twelve days and initial populations of 200 to 600 cells. Previously, Butterworth et al. [18] showed that all damage to cells would take place within 48 hours (2 days); however, twelve days are required for observable colonies to form. The advantage of computational modelling is that the results can be observed at 48 hours and assumptions made that if cells have survived to 48 hours then they will form colonies by 12 days. We add a margin of 16 hours to be sure all damage is accounted for. The complexity of the model (each cell has 21 internal species and 31 actions) mean that the evolution of an initial population of 8 G cells can be computed in reasonable time (3-20 minutes for each simulation, as in Section 3.2). In 64 hours, these will grow to at most 35 G and 35 D cells; a potential total of 1470 species.

For each of the four replications, the end population value data point is taken from each population experiment at different damage levels as in Figure 7 and a survival fraction is calculated based on the control experiment. The mean survival fraction of the four replications is shown in Figure 8 with error bars representing the standard deviation (n = 4).

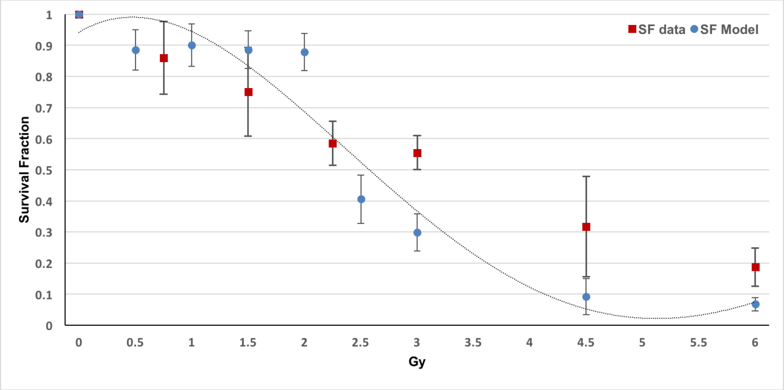

Fig. 8. Comparison results between PAL model (blue circles) and radiation treatment (red squares) survival fractions. Error bars give ± SD and fitted line to model (dotted line).

The survival fraction results are compared with the damage levels based on the simple relationship $damage/2$ = Gy dose, based on the approximate alignment of 3 Gy with damage 6. Damage levels 0, 3, 6, 9 and 12 are compared with radiation doses of 0, 1.5, 3, 4.5 and 6 respectively.

Based on these simple assumptions, the results show that the model gives a closer fit to lower Gy doses (high cell survival) but an overestimation of death at higher dose levels (low cell survival). Clearly, more needs to be done to refine the model, but the point here is to illustrate the utility of PAL.

4 Conclusion

In this paper we have discussed the definition of Process Algebra with Layers (PAL), a multi-scale process algebra designed to model multi-scale systems. PAL's strength is that it allows the convenient representation of a multi-scale system in one model, in contrast to the current hybrid frameworks. PAL removes the need for the modeller to focus on the integration of the separate modelling languages that define the separate scales in a hybrid model. The novel features of PAL are the layers of the language: Object and Population. These layers allow the user to elegantly describe the the components of each scale and the interactions between scales in one PAL model. This can allow mechanistic models to be developed showing how one layer affects another.

The addition and deletion of objects is a feature of many multi-scale systems. For example, in this study cell division requires addition and cell death requires deletion. The syntax and semantics of PAL allows this feature to be easily defined by a single action integrating the scales. In comparison, existing multi-scale Process Algebra languages psPAH [8] and PEPA nets [9] would need multiple lines of code to define this feature, making their models difficult to construct and read.

The Objects in PAL currently do not have the ability to interact explicitly with one another, which may be a limitation. This would involve explicitly modelling space. Objects would, for example, need location attributes to react to their surrounding Objects. It would be necessary to ensure this addition would not compromise the integrative nature of PAL. This spatial definition may overcomplicate the definition of a PAL model which may lead to the loss of some multi-scale features PAL already encapsulates.

PAL has been applied successfully to a cell cycle and DNA damage multi-scale system here. The PAL model links together the established models of Zhang et al. [10] and Tyson et al. [11]. This allowed the creation of a novel model investigating the effects of DNA damage from radiation on a cell colony by linking mechanistically to the progression of the cell cycle as determined by cell proteins and mass. PAL easily captures the internal species and the colony activity, thus supporting investigation across scales in one model. The Object and Population results of the model showed that low radiation doses do not significantly affect the cell cycle average length, nor do they significantly affect colony growth of cancer cells.

The comparison of results to wet laboratory data shows how PAL can be used to aid in investigations of cancer treatment in systems biology. For example, the profile of Figure 8 suggests the model can be further analysed by varying the threshold for cell death, and by modifying the simple assumption of $damage/2 =$ Gy. The model could also be refined by including more varied notions of repair, which would need targetted wet lab experiments to measure repair rates. Exploration of hypotheses in the PAL model develop understanding of the system and direct attention to the most sensitive areas for parameters or compounds. This, in turn, allows researchers to develop a more focussed programme of future biological experiments in DNA damage, cell cycle and population growth rates, reducing the number of expensive and time-consuming biological wet laboratory work. The strength of modelling approaches can be directly correlated with how they can affect broader science questions in a multi-disciplinary approach.

Future work on this PAL model could include testing a variety of different degradation rates for CycB affected by p53. The damage repair could be more specific to the levels of p53 and Mdm2. Further work could be undertaken to compare the PAL model results to other cancer treatments such as Temozolomide (TMZ) and combination of these treatments (radiation + TMZ). This comparison could be achieved easily as the damage is abstract in the model therefore the focus can be made on the damage relationship function to the specific treatment. Future work on PAL itself will include a direct implementation of a PAL tool (thereby avoiding the limitations of translation into Bio-PEPA), exploration of other analysis techniques such as model checking, and translation to/from common languages such as SBML (already available for Bio-PEPA).

Acknowledgements

Erin Scott is grateful to the Scottish Informatics and Computer Science Alliance (SICSA), a research initiative of the Scottish Funding Council, for financial support of her PhD studies. We wish to thank the EPSRC project EP/K039342/1 for supplying us with the idea of the case study and the wet laboratory data. We also thank the anonymous referees for their helpful comments in improving this document.

References

1. Nicol, J.: Radiation Dose Enhancement: The development and application of radio consisting gold nanoparticles. PhD thesis, Queen's University Belfast (2016)
2. Dada, J.O., Mendes, P.: Multi-scale modelling and simulation in systems biology. Integrative biology : quantitative biosciences from nano to macro **3**(2) (2011) 86–96
3. Walker, D.C., Southgate, J.: The virtual cell–a candidate co-ordinator for 'middle-out' modelling of biological systems. Briefings in bioinformatics **10**(4) (July 2009) 450–61
4. Powathil, G.G., Gordon, K.E., Hill, L.A., Chaplain, M.A.J.: Modelling the effects of cell-cycle heterogeneity on the response of a solid tumour to chemotherapy : Biological insights from a hybrid multiscale cellular automaton model. Journal of Theoretical Biology **308** (2012) 1–19

5. Johns, T.C., Carnell, R.E., Crossley, J.F., Gregory, J.M., Mitchell, J.F.B., Senior, C.A., Tett, S.F.B., Wood, R.A., Low: The second Hadley Centre coupled ocean-atmosphere GCM: Model description, spinup and validation. Climate Dynamics **13**(2) (1997) 103–134
6. Priami, C.: Process calculi and life science. Electronic Notes in Theoretical Computer Science **162** (2006) 301–304
7. Romero-Campero, F.J., Twycross, J., Cao, H., Blakes, J., Krasnogor, N.: A Multiscale Modelling Framework Based On P Systems. Membrane Computing **5391** (2009) 63–77
8. Degasperi, A.: Multi-scale modelling of biological systems in process algebra with multi-way synchronisation. PhD thesis, University of Glasgow (2011)
9. Gilmore, S., Hillston, J., Kloul, L., Ribaudo, M.: PEPA nets: a structured performance modelling formalism. Performance Evaluation (2003) 111–130
10. Zhang, T., Brazhnik, P., Tyson, J.J.: Exploring mechanisms of the DNA-damage response: p53 pulses and their possible relevance to apoptosis. Cell Cycle **6**(1) (2007) 85–94
11. Tyson, J.J., Novak, B.: Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions. Journal of theoretical biology **210**(2) (2001) 249–263
12. Scott, E.: Process Algebra with Layers: a language for Multi-Scale Integration Modelling. PhD thesis, http://hdl.handle.net/1893/23516, University of Stirling (2016)
13. Ciocchetta, F., Hillston, J.: Bio-PEPA: a framework for the modelling and analysis of biological systems. Theoretical Computer Science **410** (2009) 3065–3084
14. Powathil, G.G., Adamson, D.J., Chaplain, M.A.: Towards predicting the response of a solid tumour to chemotherapy and radiotherapy treatments: clinical insights from a computational model. PLOS Computational Biology **9**(7) (2013)
15. Guerrero, P., Alarcón, T.: Stochastic Multiscale Models of Cell Population Dynamics: Asymptotic and Numerical Methods. Mathematical Modelling of Natural Phenomena **10**(1) (2015) 64–93
16. Eclipse Foundation: Eclipse web page: https://eclipse.org/ (2017)
17. SBML.org: SBML web page : http://sbml.org/Main_Page (2012)
18. Butterworth, K. T., Nicol, J. R., Ghita, M., Rosa, S., Chaudhary, P., McGarry, C. K., ... Prise, K.M.: Preclincial evaluation of Gold-DTDTPA Nanoparticles As Theranostic Agents In Prostate Cancer Radiotherapy. Nanomedicine **10.2217/nn** (2016) 2035–2047