

### **Model Documentation: universal LSCs**

Table S1 in Supplementary Material lists all universal LSCs (uLSCs) organized by “use case”, a brief description and the page number reference for this documentation file.

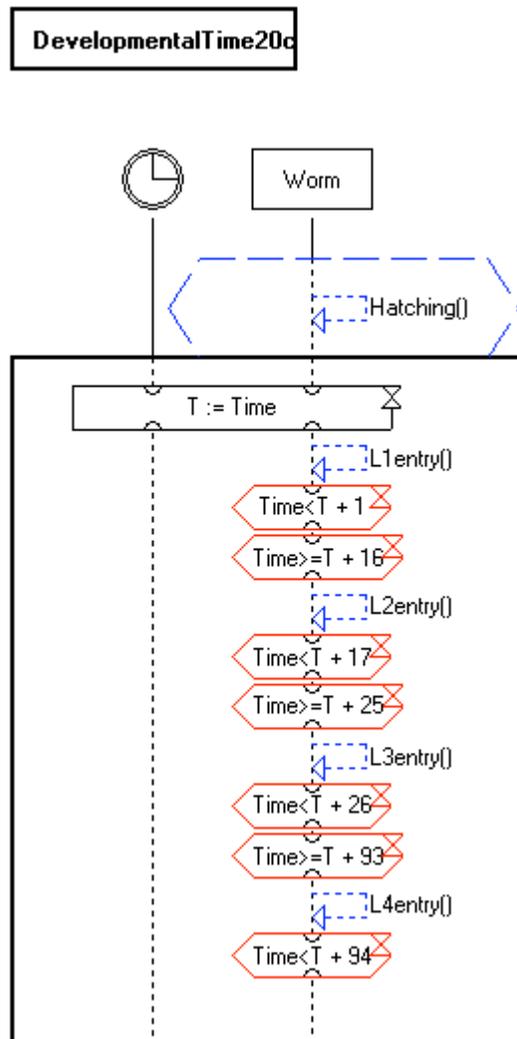
This file contains headings corresponding to each use case followed by each uLSC within the use case together with an explanation of the behaviors it specifies. Within the explanations, we use the convention “LSC name (use case name)” when referring to LSCs.

The existential LSCs (eLSCs) are provided in a separate Supplementary Material document (Model Documentation: existential LSCs). These are organized in use cases in the model according to the data table they represent.



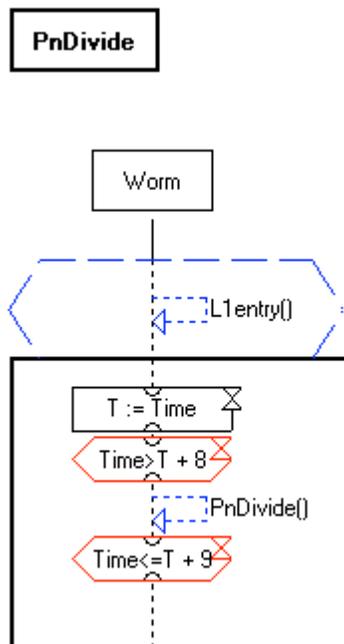
## Core Behaviors

This use case contains LSCs that specify broadly-applicable behaviors. The “DevelopmentalTime20c” LSC controls important high-level events that drive the behavior of the whole model (the timing of the larval stages). The LSC “PnDivide” sets the ordering and timing of the division of the Pn cells, whose daughters the Pn.p cells are at the focus of our model. LSC “VPCborn” specifies birth of the Pn.p cells. LSC “PnpAssumeFate” governs the timing of VPC fate assumption.

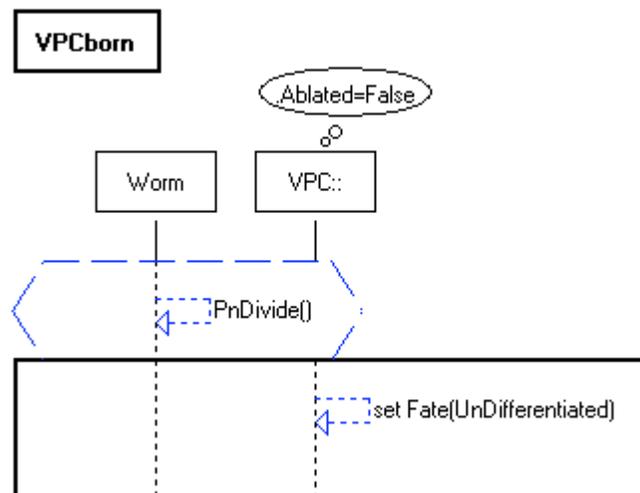


The LSC “DevelopmentalTime20c” specifies the ordering and timing of the relevant developmental stages. It serves as a “skeleton” that drives the behavior of the model. The Hatching method appearing in the prechart triggers the activation of the chart, and the main chart specifies that entering the larval stages - L1entry, L2entry, L3entry and L4entry must occur according to this order and satisfying timed constraints. The time lengths of the larval stages L1, L2 and L3 are 16, 9 and 9 hours respectively (Sulston & Horvitz, 1977; Data based on timescale). This is specified in the LSC by storing the time in variable T immediately after

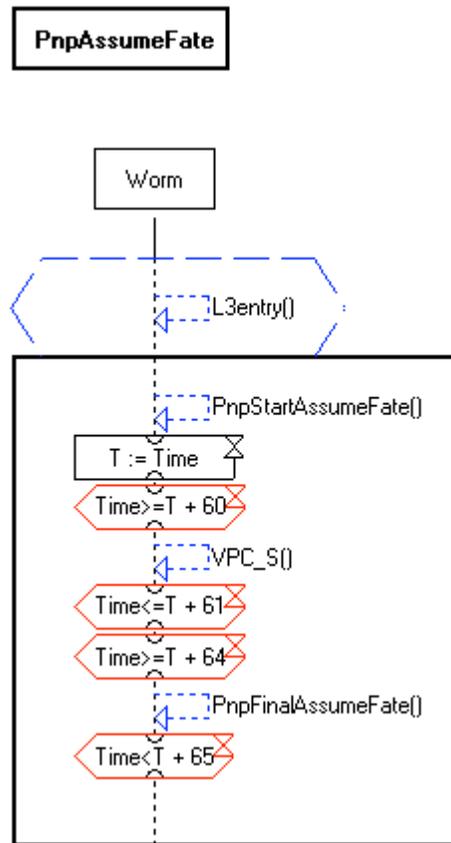
Hatching, and using timed constraints in conditions. Thus the condition  $\text{Time} < T + 1$  ensures L1entry occurs after Hatching (no time progress allowed between Hatching and L1entry). The conditions  $\text{Time} \geq T + 16$  and  $\text{Time} < T + 17$  specify L2entry is 16 hours after hatching, and similarly the conditions  $\text{Time} \geq T + 25$  and  $\text{Time} < T + 26$  specify L3entry is 25 hours after hatching. In the L3 stage we measure the first hour in the scale of minute units, and after that again in the scale of hour units, thus the conditions  $\text{Time} \geq T + 93$  and  $\text{Time} < T + 94$  specify that L4entry is 34 hours (25 hrs + 60 min + 8 hrs = 93 clock ticks) after hatching.



The LSC “PnDivide” specifies that the Pn cells divide 9 hours after hatching. This LSC will be activated by the execution of L1entry method (in the main chart of “DevelopmentalTime20c (Core Behaviors)”).



The LSC “VPCborn” is activated by the execution of the PnDivide method (in the main chart of “PnDivide (Core Behaviors)”). Initially the fate of all VPCs is None specifying that these cells are not born yet. In “VPCborn” every VPC that is not ablated is set to fate UnDifferentiated. This occurs 9 hours after hatching (as specified by “PnDivide (Core Behaviors)”).

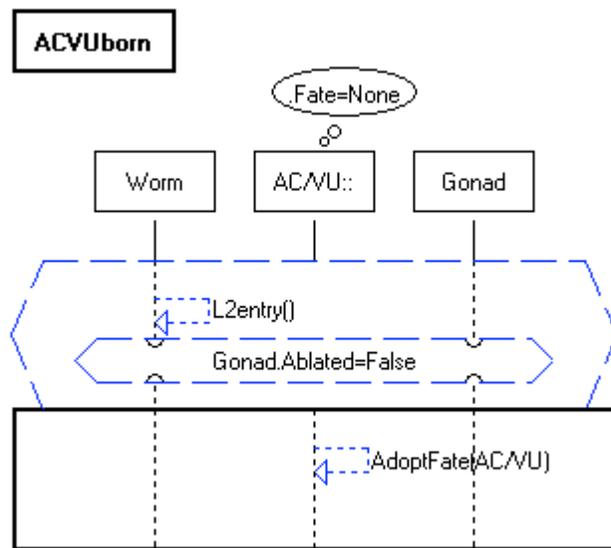


The LSC “PnpAssumeFate” specifies that the earliest Pn.p fate assumption starts after L3entry, as specified by the method PnpStartAssumeFate, S phase occurs 1 hour (in 60 “minute” clock ticks) after L3 entry (Ambros, 1999) and is specified using the VPC\_S method, while the final Pn.p fate assumption happens 5 hours after L3 entry, the conditions  $\text{Time} \geq T + 64$  and  $\text{Time} < T + 65$  reflect the fact that the first hour in L3 is counted in minutes scale.

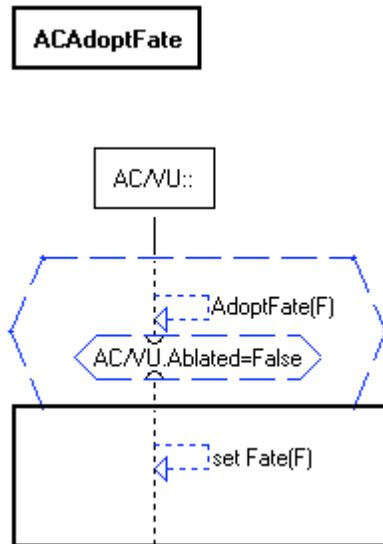
## AC/VU

The anchor cell plays an important role in the fate specification for VPCs (Kimble, 1981). A major gene in fate specification, *lin-12*, is also critical for AC formation (Greenwald et al., 1983). This use case thus describes some relevant aspects of the AC/VU decision. We focus on two cells, Z1.ppp and Z4.aaa that have a potential to become anchor cells.

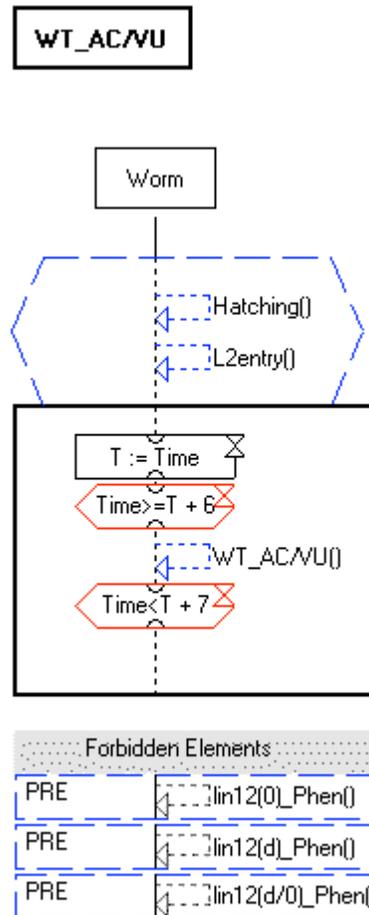
In our model we have an AC/VU class, where both objects Z1.ppp and Z4.aaa are instances of the AC/VU class. The use of the class definition allows us to specify behavioral properties which hold for both Z1.ppp and for Z4.aaa in a more concise fashion. Our model does not represent the other VU cells that can become ACs in a *lin-12(0)* background resulting in a phenotype with more than 2 ACs.



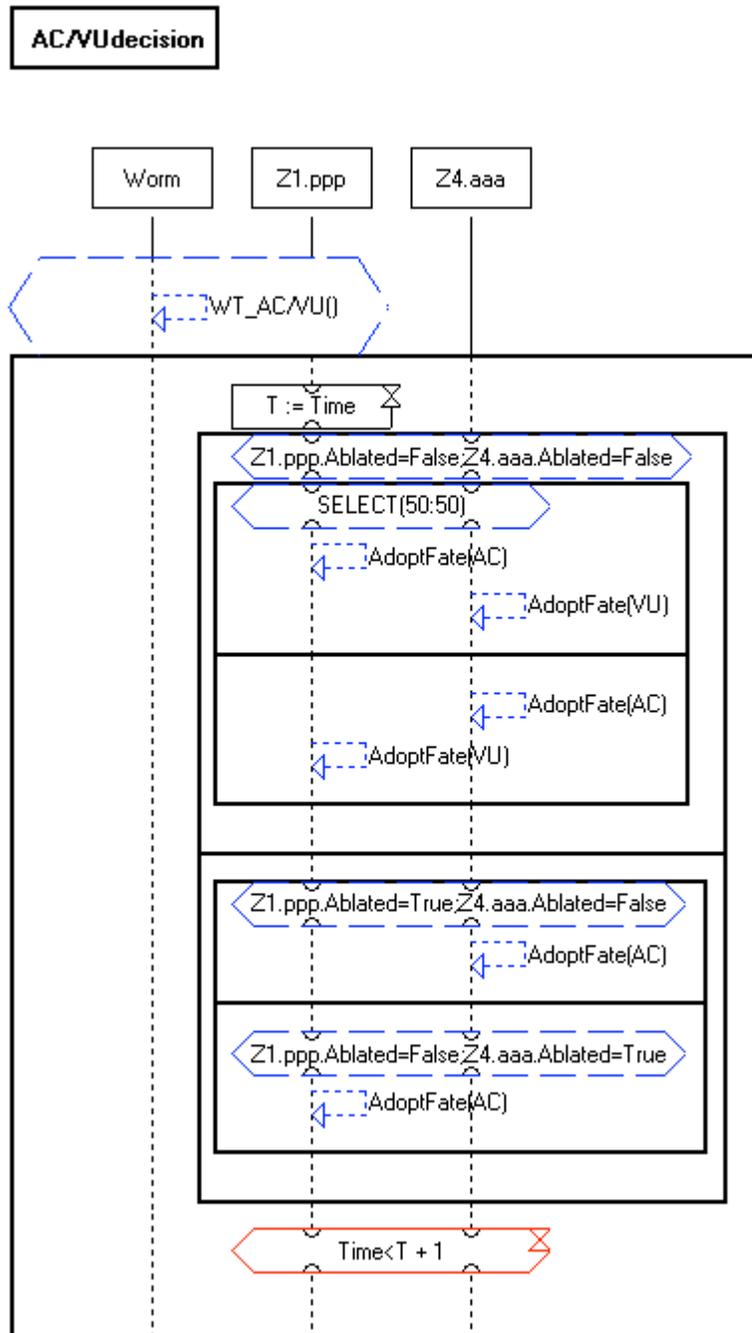
The LSC “ACVUborn” specifies the setting of an AC/VU fate triggered by the execution of method L2entry appearing in the prechart. This LSC will be activated by the execution of L2entry method in the main chart of “DevelopmentalTime20c (Core Behaviors)”. For the main chart to be triggered the condition “Gonad.Ablated = False” must hold; this condition allows fates to be assigned only to cells that have not been ablated. The main chart specifies that each AC/VU object whose fate is currently set as None, will eventually change its Fate to AC/VU by calling the AdoptFate(AC/VU) method. The actual change of the property Fate from value None to value AC/VU happens as a result of the activation of LSC “ACAdoptFate”.



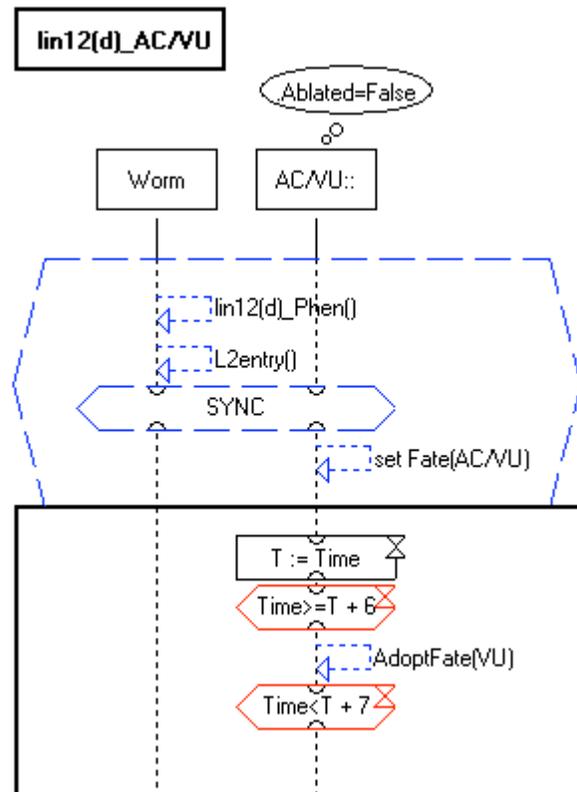
The LSC “ACAdoptFate” specifies that for every AC/VU instance that executes the AdoptFate(F) method, the Fate property will be set to the value of F by executing the setFate(F) message. This LSC makes use of both symbolic instances (the fact that the participating instance is the symbolic instance AC/VU), and of symbolic messages (the fact that we use the parameter F in the messages AdoptFate(F) and setFate(F)). This alleviates having to specify an equivalent behavior by creating a concrete LSC for each instance (Z1.ppp, Z4.aaa) and for each possible fate value (None, AC/VU, AC, VU). This two step fate assignment does not reflect here a biological phenomenon, but rather a common programming pattern, which is also used in use case VPC Fate Assumption, and ensures that an ablated cell does not acquire a fate.



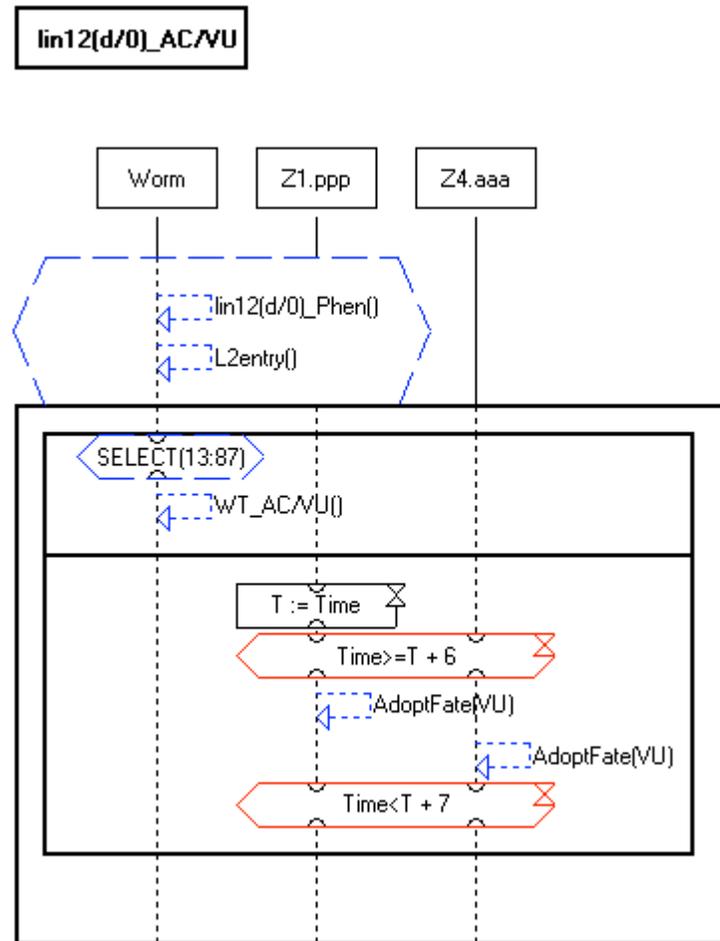
The timing and appropriate genetic background of the WT mechanism that establishes the AC/VU decision is controlled by LSC “WT\_AC/VU”. A method is established (“WT\_AC/VU”) to trigger that mechanism (contained in LSC “AC/VUdecision”) under the appropriate conditions. LSC “WT\_AC/VU” specifies the triggering of the wild-type AC/VU decision. This is done by executing the WT\_AC/VU method 6 hours after L2entry. For certain *lin-12* mutations this AC/VU decision is different from the wild-type case. The model therefore should not execute the method WT\_AC/VU in the case of *lin-12(0)*, *lin-12(d)* or *lin-12(d/0)* mutations. This is achieved by designating methods that are executed for these *lin-12* mutations in the Forbidden Elements section. The methods *lin-12(0)\_Phen*, *lin-12(d)\_Phen* and *lin-12(d/0)\_Phen* thus appear as cold forbidden messages, with the prechart as their scope, meaning that if any of them occur while the prechart is active (after hatching but before L2entry in this LSC) the prechart is exited without the main chart being activated, and thus the method WT\_AC/VU is not executed.



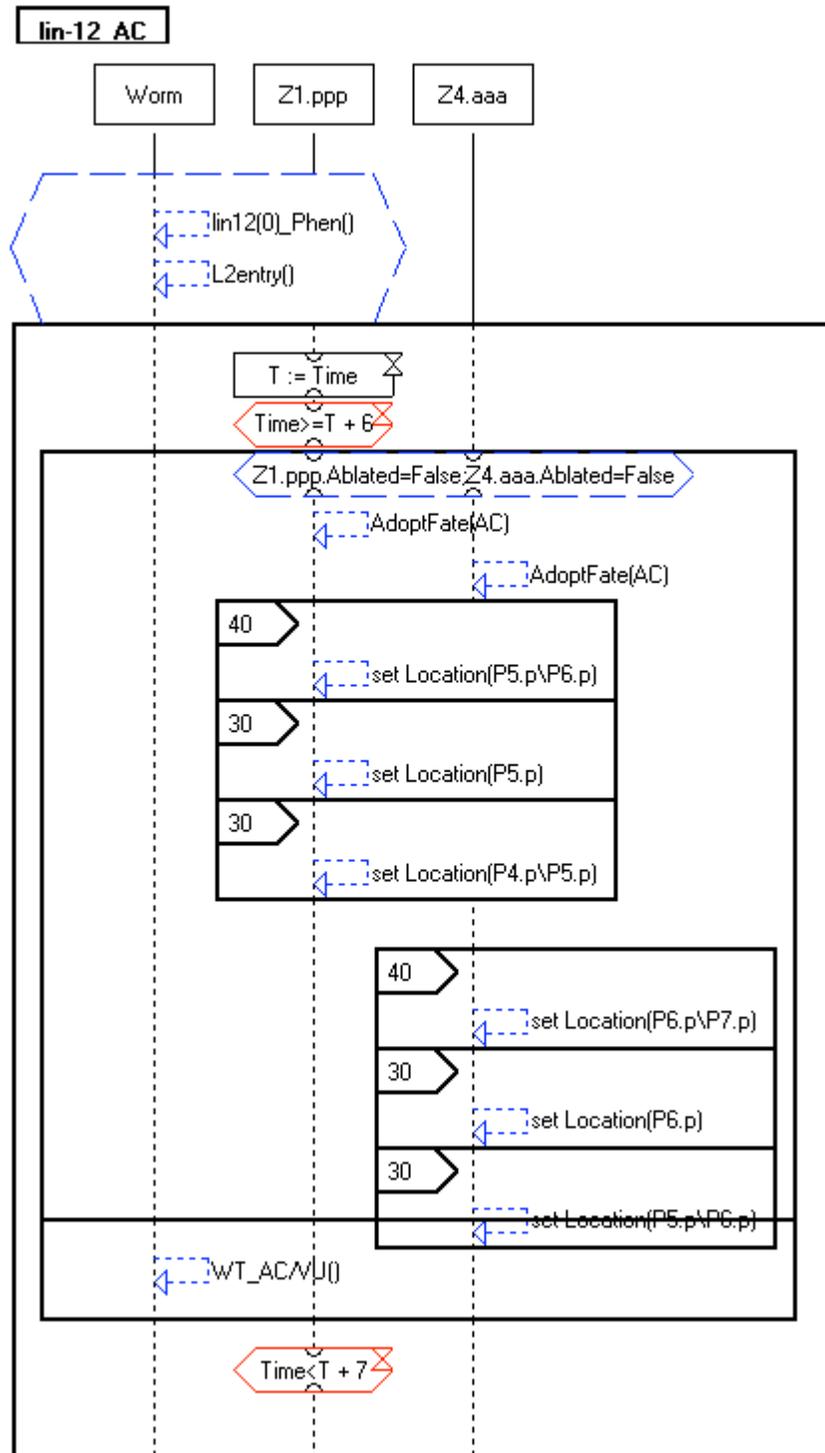
LSC “AC/VUdecision” specifies the wild-type AC/VU decision. The LSC is activated by the method WT\_AC/VU executed in the main chart of LSC “WT\_AC/VU”. Within a population of wild-type animals, in half of the individuals, Z1.ppp adopts the AC fate while Z4.aaa adopts a VU fate. In the other half these fates are reversed (Z1.ppp adopts a VU fate and Z4.aaa adopts the AC fate). This behavior is modeled as a non-deterministic decision. The other aspect of the behavior that is represented here is that if one of the cells Z1.ppp or Z4.aaa is ablated the remaining one will become an anchor cell (Kimble, 1981).



LSC “`lin12(d)_AC/VU`” specifies that in a *lin-12(d)* background both Z1.ppp and Z4.aaa will adopt the fate VU 6 hours after they are born, if they are not ablated (Greenwald et al., 1983). Thus 100% of animals homozygous for the *lin-12(d)* mutation in the current model (*lin-12(n137)*) have no anchor cell.

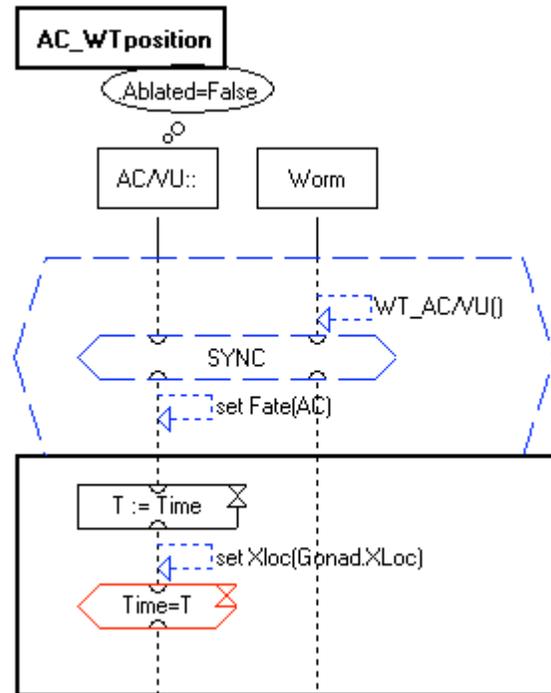


LSC “lin12(d/0)\_AC/VU” (Greenwald et al., 1983, Table 3) specifies that in the *lin-12(d/0)* genotype in 13% of the cases the behavior is like that of the wild-type and is activated by the method “WT\_AC/VU” while in the remaining 87% of the cases the behavior is like that of *lin-12(d)* background thus both Z1.ppp and Z4.aaa will adopt the fate VU six hours after L2entry. Checking that a cell is not ablated before assuming VU fate is performed in LSC “ACAdoptFate”.

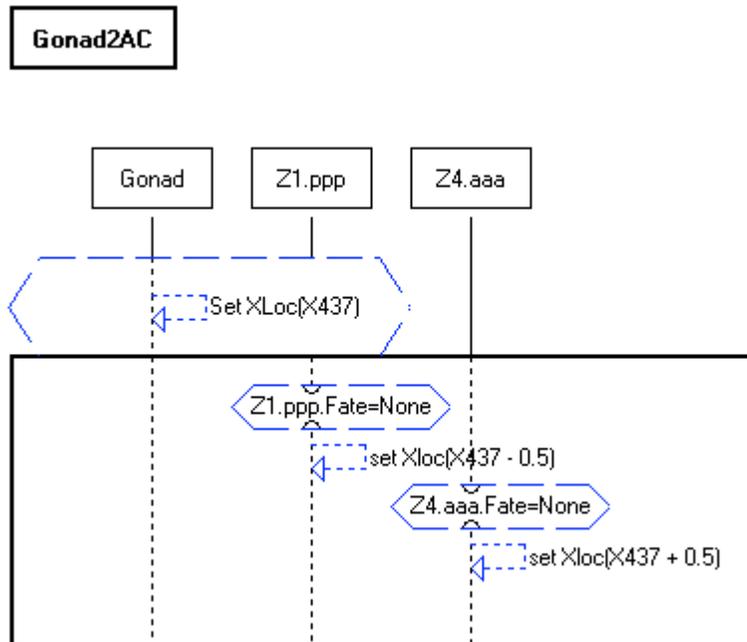


LSC "lin-12\_AC" specifies that in a *lin-12(0)* background both Z1.ppp and Z4.aaa become anchor cells if they are not ablated. If one of them is ablated, the remaining cell becomes an anchor cell, as in the wild-type. Thus the execution of the message "WT\_AC/VU" is to activate this behavior specified in LSC "AC/VUdecision". To be able to reproduce more

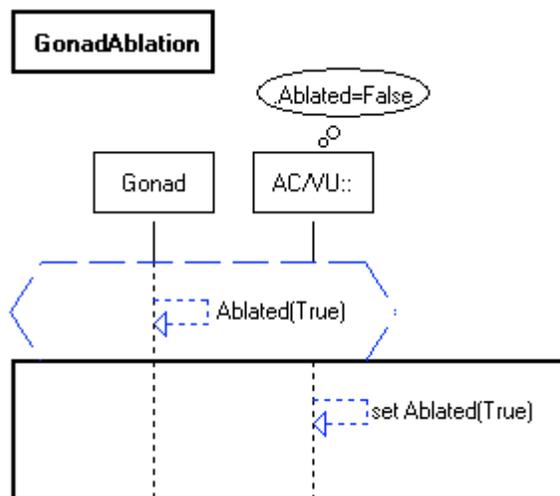
accurately results of *lin-12(0)* background, the LSC “lin-12\_AC” allows non-deterministic positioning of the two anchor cells, which affects the level of LIN-3 to which the VPCs respond and, as a result, the fates they adopt.



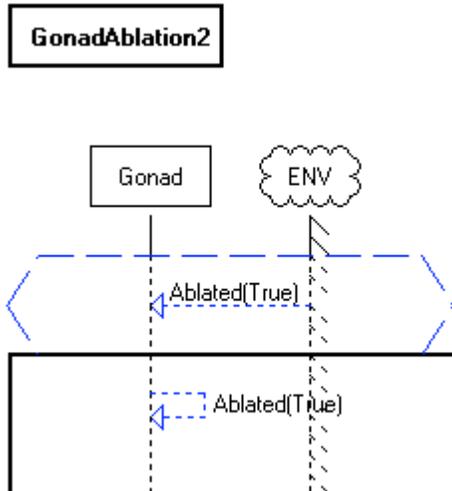
LSC “AC\_WTposition” specifies that in wild-type development the anchor cell moves towards the center of the gonad (Kimble and Hirsch 1979). This will occur 6 hours after L2entry, as specified in “WT\_AC/VU” LSC. Since either Z1.ppp or Z4.aaa can become an anchor cell and move to the center of the gonad, we use a symbolic instance with class AC/VU in “AC\_WTposition” to handle both possibilities. Thus an AC/VU cell that becomes an anchor cell moves to the center without any time delays, setting its X coordinate location to that of the gonad. In a *lin-12(0)* background typically both Z1.ppp and Z4.aaa become anchor cells, however in this case LSC “AC\_WTposition” is not activated since this is not a wild-type behavior and the WT\_AC/VU method is not executed, thus avoiding a situation in which both cells move to the center position.



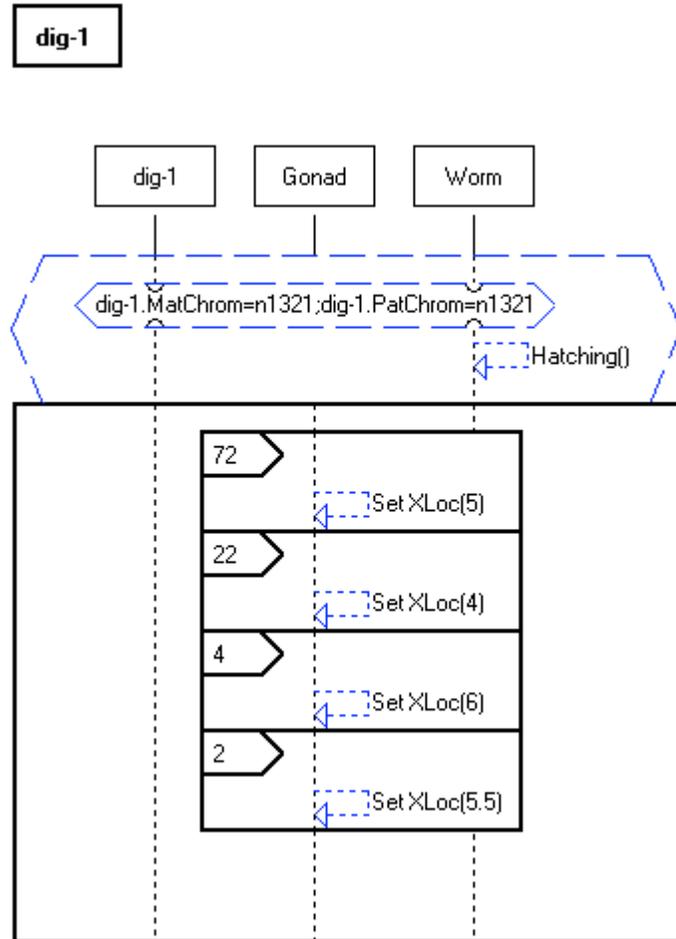
The LSC “Gonad2AC” specifies that if the gonad moves from its normal position (e.g., a *dig-1* mutant), the cells Z1.ppp and Z4.aaa move with the gonad and are positioned half a unit (measured in units of distance between VPCs) to the left and to the right of the X location of the center of the gonad.



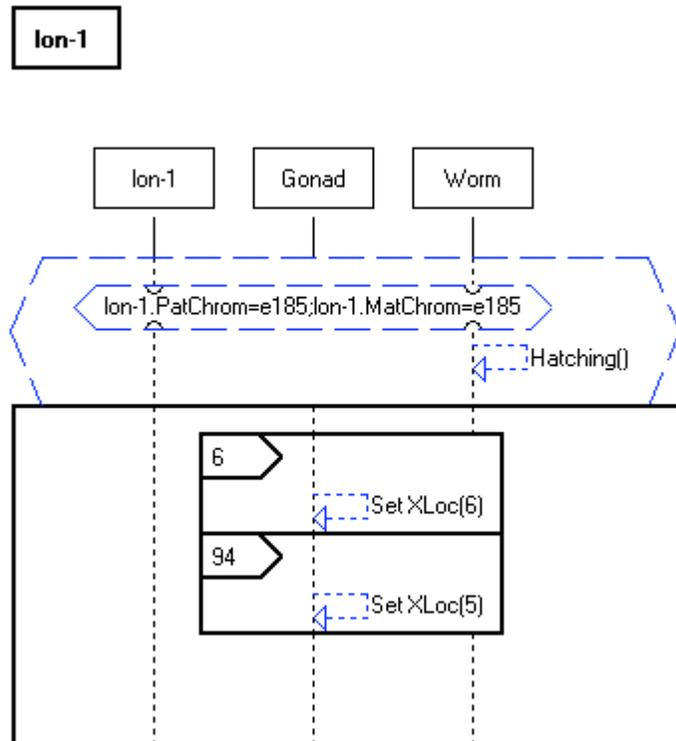
The LSC “GonadAblation” specifies that the cells Z1.ppp and Z4.aaa are part of the gonad. Thus ablating the gonad results in these cells being ablated.



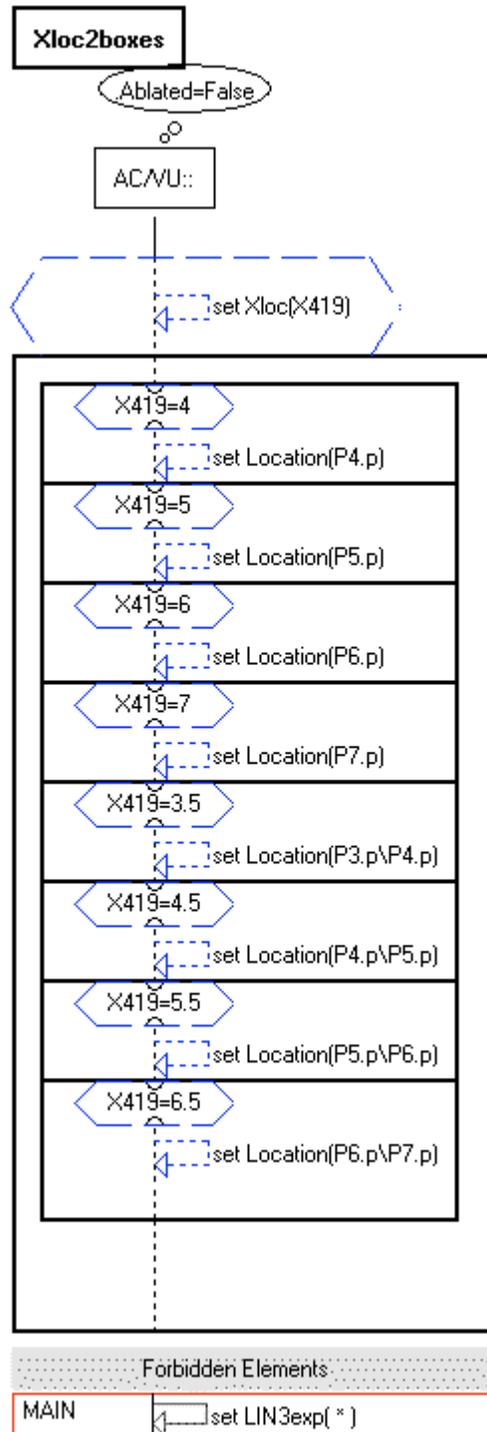
The LSC “GonadAblation2” specifies that when the Gonad is ablated as a result of an external action (designated by the message “Ablated(True)” sent from the object ENV that represents an environment to the gonad) the gonad becomes ablated as specified by the self message Ablated(True), resulting in an activation of LSC “GonadAblation”. LSC “GonadAblation2” is used in interactive play-out, where the user actively performs a gonad ablation, whereas LSC “GonadAblation” can also be directly activated in “batch” mode.



The LSC “*dig-1*” defines a probabilistic distribution for the X location of the gonad that is displaced in the *dig-1* background (Thomas et. al., 1990). In 72% of the cases the gonad sets its X location to 5 (above P5.p), in 22% of the cases the X location is set to 4 (above P4.p), in 4% of the cases the X location is set to 6 (the normal wild-type position above P6.p) while the remaining 2% of the cases the X location is set to 5.5 (above the midway position between P5.p and P6.p).

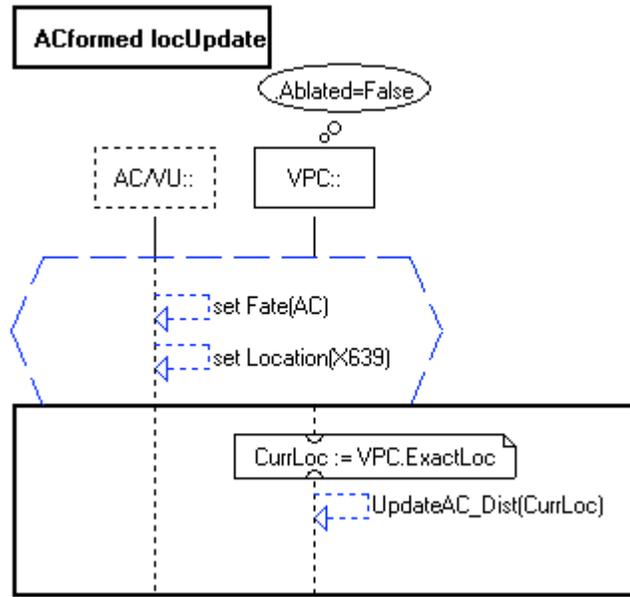


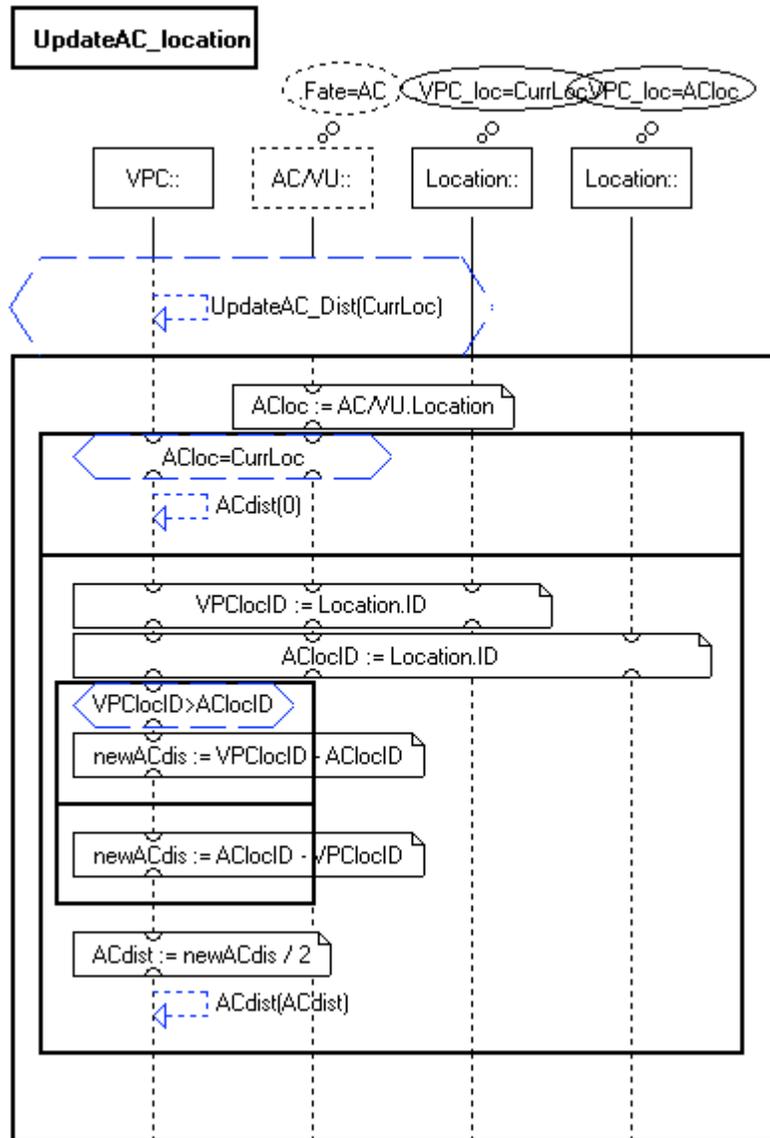
The LSC "lon-1" defines a probabilistic distribution for the X location of the gonad in a similar way to that of LSC "dig-1". In 6 % of the cases the X location is set to 6 (the normal wild-type position above P6.p) while the remaining 94% of the cases the X location is set to 5 (above P5.p).



The AC/VU class has two properties that are used to maintain the X location, the Xloc property which is of type number (range -32000 to 32000 with delta 0.001) and the Location property which is of type VPCLocation (an enumeration with the values PreP3.p, P3.p, P3.p\P4.p, P4.p, P4.p\P5.p, P5.p, P5.p\P6.p, P6.p, P6.p\P7.p, P7.p, P7.p\P8.p, P8.p, PostP8.p). The property VPCLocation is calculated from the property Xloc each time the Xloc property is

modified as shown in LSC “Xloc2boxes”. The property Xloc can be directly used in a numeric computation, e.g., in LSC “Gonad2AC” or in Visual Basic functions that update the GUI), while the Location property is used for a more direct comparison with the VPCs and their Locations, as specified in use cases LIN-3 and Pn.p Movements. The property VPCLocation is more abstract than property Xloc, Xloc provides a handle for more detailed modeling of anchor cell positions and movements in the future.

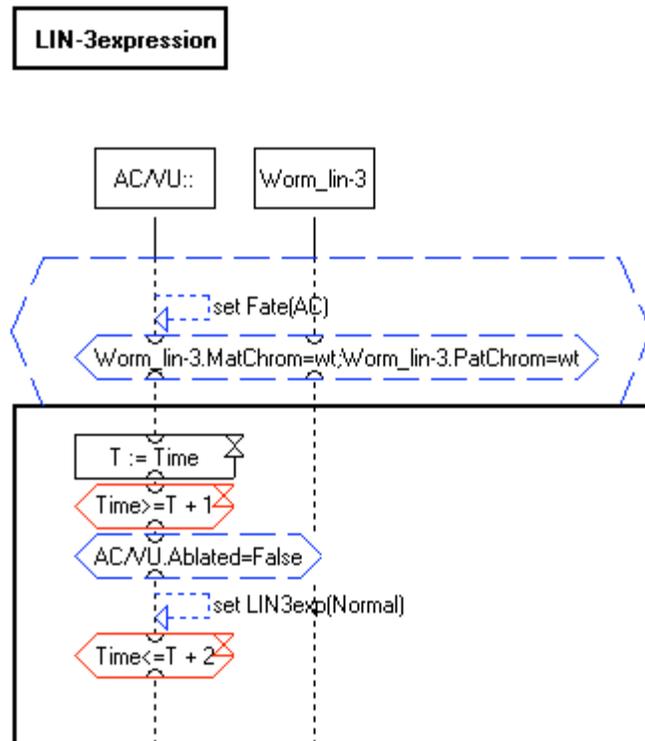




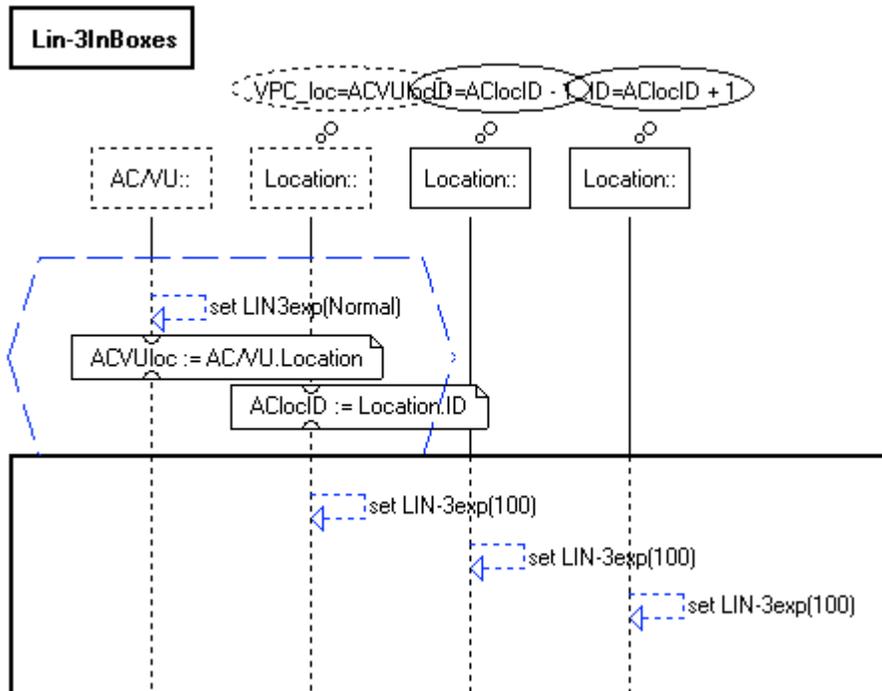
LSCs “ACFormedlocUpdate” and “UpdateAC\_location” calculate the distance of a VPC from the anchor cell. This distance is stored in property ACdist. This property is also updated upon VPC movement.

## LIN-3

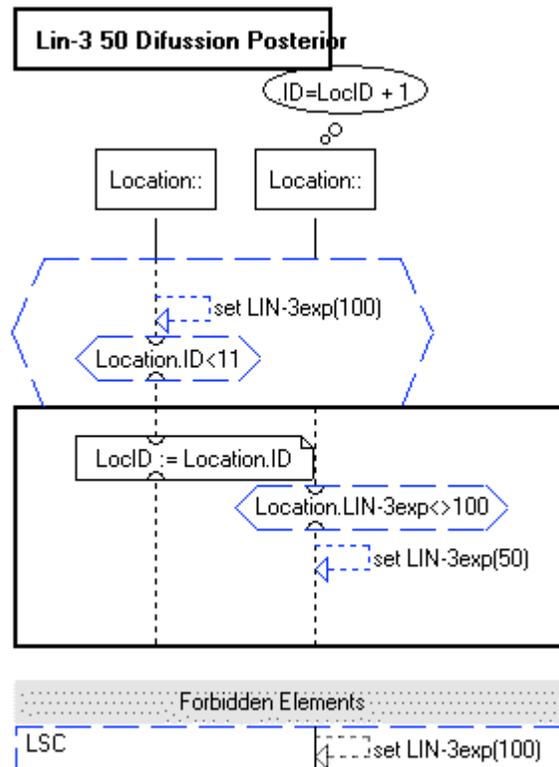
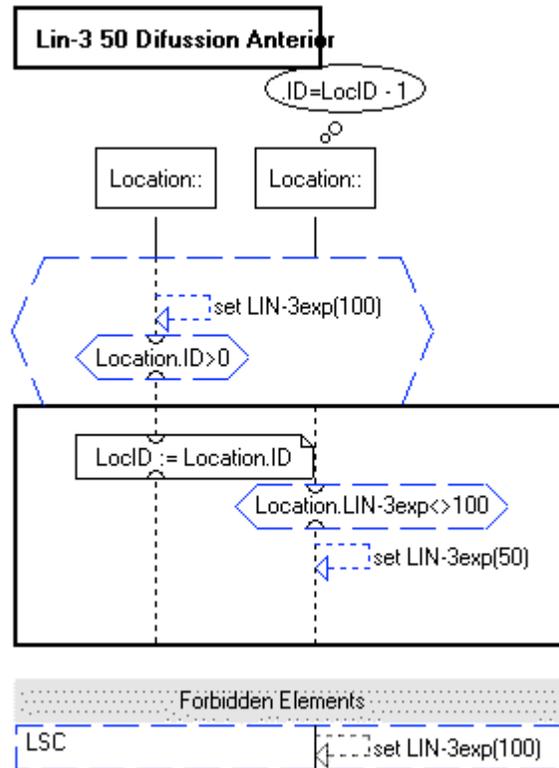
VPCs are induced to adopt vulval fates by a LIN-3 signal (Hill and Sternberg, 1992). This use case describes the LIN-3 gradient formation which is centered around the anchor cell. We assume 3 possible LIN-3 concentration levels: High ( $LIN-3_{exp} = 100$ ), Medium ( $LIN-3_{exp} = 50$ ) and Low ( $LIN-3_{exp} = 20$ ). In our modeling the anchor cell will set a concentration level for each of the Locations, and a VPC will respond to the level according to the Location which it occupies (that is, the central Location, see LSC “ClearNeighbors (Pn.p Movements)” for details). It should be noted that the current modeling of the gradient is quite simplistic (in terms of possible values of LIN-3 expression and the effect of genetic mutations on LIN-3 expression) but suffices for our modeling purposes.



The LSC “LIN-3expression” sets the LIN-3 expression of an anchor cell to Normal if the *lin-3* gene is wild-type between 1-2 hour after the cell adopts an AC fate, which occurs in the wild-type 6 hours after L2entry as specified in LSC “WT\_AC/VU (AC/VU)”.

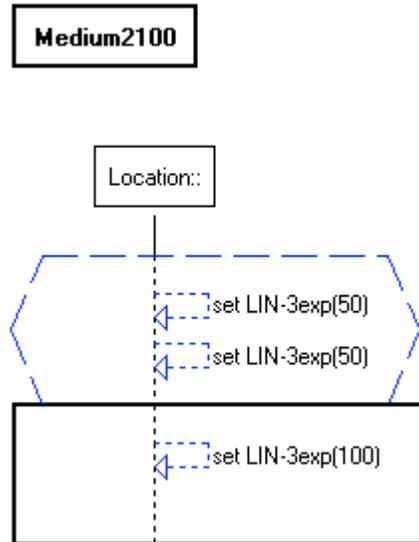


The LSC “LIN-3InBoxes” sets a high expression level of LIN-3 (LIN-3exp = 100) to the three Locations that are closest to the anchor cell, the Location immediately below the anchor cell and one Location to the left and to the right of this Location. The LSC uses symbolic instances both for the AC/VU cell and for the Locations, since each of the cells Z1.ppp and Z4.aaa has the potential to become an anchor cell, and also the Locations that will receive a high LIN-3 level which are Locations with VPC\_loc property values of P5.p/P6.p, P6.p and P6.p/P7.p and IDs 6,7 and 8 in the wild-type, could include different Locations for example in a *dig-1*, a *lon-1* or a *lin-12(0)* mutation.

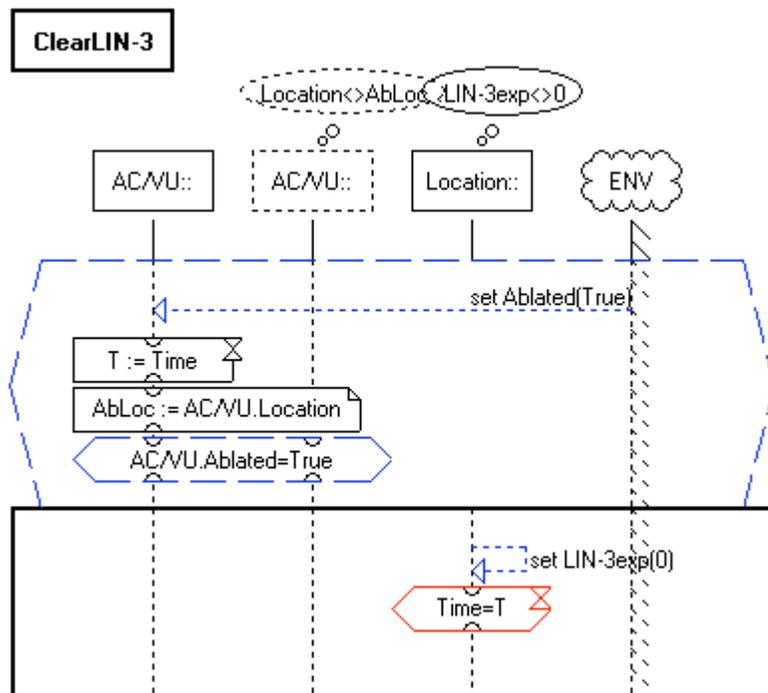


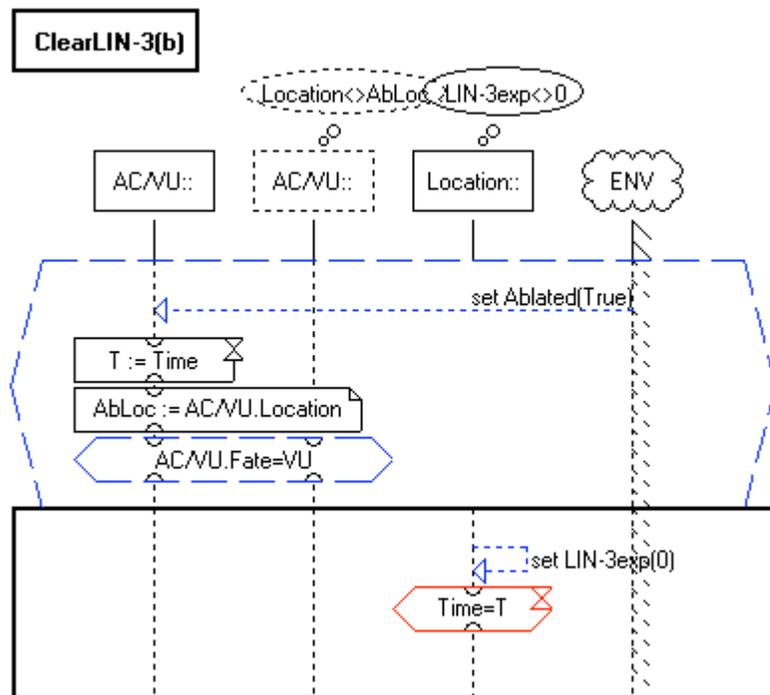


which the new LIN-3 expression will be set to low level (LIN-3exp = 20). The condition  $LIN-3exp \neq 100; LIN-3exp \neq 50$  ensures that the Location that is set to a low level is not a Location in which LIN-3 is already expressed at a high or medium level.

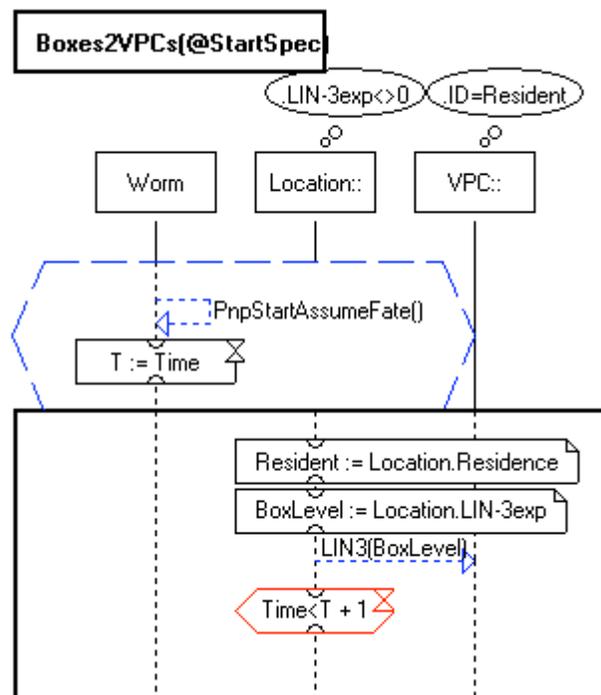


In LSC "Medium2100" a Location whose LIN-3 expression is updated twice to a medium level (LIN-3exp = 50) is, as a result, set to a high expression level (LIN-3exp = 100). Although it is not specified directly in this LSC, this LSC will be activated in a *lin-12(0)* background where there are two ACs.





LSCs “ClearLIN-3” and “ClearLIN-3(b)” sets all non-zero LIN-3 expression in all Locations to 0 immediately after ablation of all anchor cells.

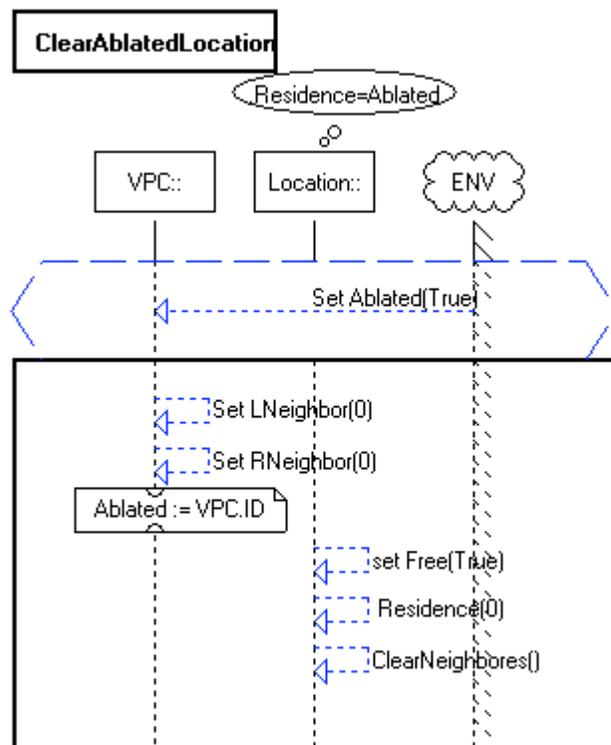


The LSC “Boxes2VPCs(@StartSpec)” is triggered by the method PnPStartAssumeFate, and each Location whose LIN-3 expression is not 0, sends the method LIN3(BoxLevel) to the VPC that resides in this Location, where the parameter BoxLevel stores the level of LIN-3 expression in the Location. The VPCs “hear” the level of LIN-3 expression according to the Location in which they reside.

## Pn.p Movements

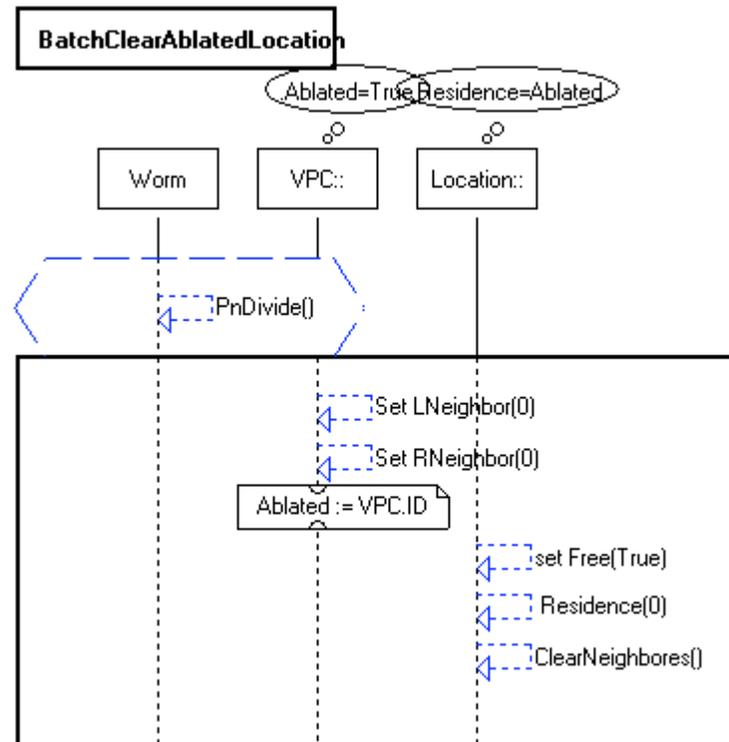
This use case describes our representation of movements of VPCs as a result of ablating other VPCs (Sulston and White, 1980; Sternberg and Horvitz, 1986). The movement description is important for our model since the fate a VPC adopts depends on its position relative to the anchor cell and the neighboring VPCs. Our modeling of VPC movements is somewhat complex and detailed, but makes simplifying assumptions about the biology; more experimental data is needed to build and validate a more elaborate and realistic representation.

The direct physical consequences of an ablation – that a gap appears where the ablated cell was, for instance – are specified, as well as events like another cell moving to the vacated space. The cells are modeled as keeping track of their neighbors, so that when a neighbor is ablated, a remaining cell is cued that there is a space it may move into. As part of the book-keeping required for this model, Locations store information about their own contents in the Residence and Free properties.

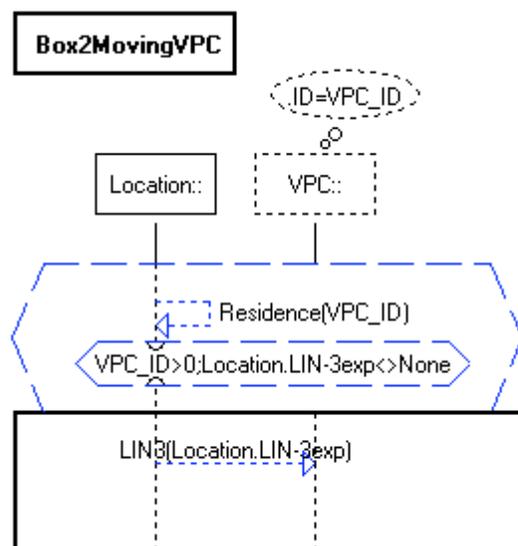


Properties of an object can be altered by other objects or by elements outside of the system, referred to collectively as the environment, or ENV. ENV represents a user clicking on objects during manual play-out. This LSC is triggered when a VPC's Ablated property is set to True by the ENV. LSC "ClearAblatedLocation" specifies that after a cell is ablated, it dies, and clearing of its debris results in an open space where the cell used to reside. The ablated cell's left and right neighbors, specified by the properties Lneighbor and Rneighbor are set to zero to ensure that the ablated cell does not continue to react to newly-vacated neighboring spaces. The

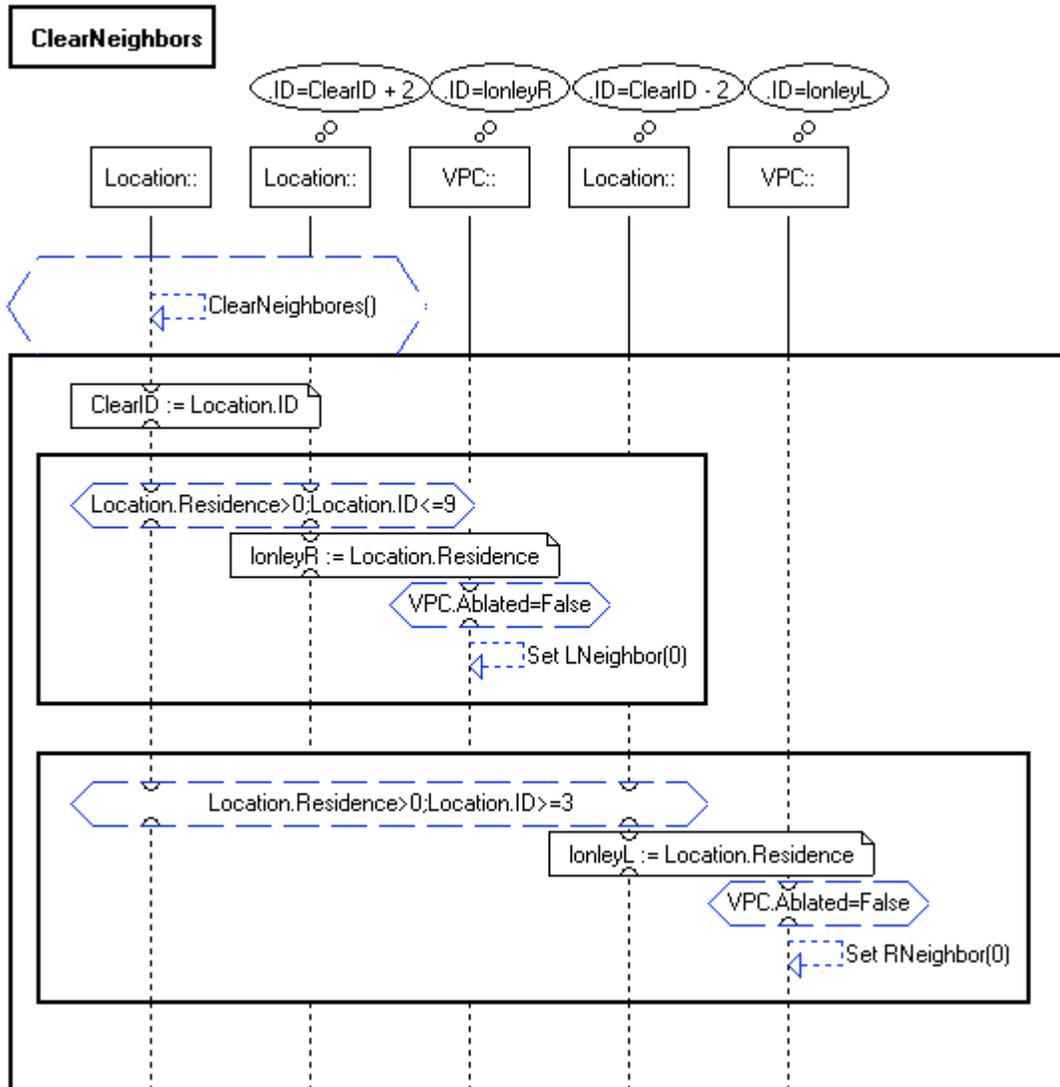
Locations previously holding the recently-ablated cell makes note that it no longer contains a cell.



During batch mode it is currently impossible for a user to click and to interrupt the simulation. For cell ablation in batch run, a VPC's Ablated property may be set to True automatically by a Jump Start at the beginning of a simulation. The method PnDivide (executed in the main chart of LSC "PnDivide (Core Behaviors)") was chosen as an event that occurs prior to any VPC movements; in fact, it occurs by definition before the Pn.p cells are born. This way, the response to cell ablation that begins in this LSC's main chart is ready to go upon Pn.p birth.

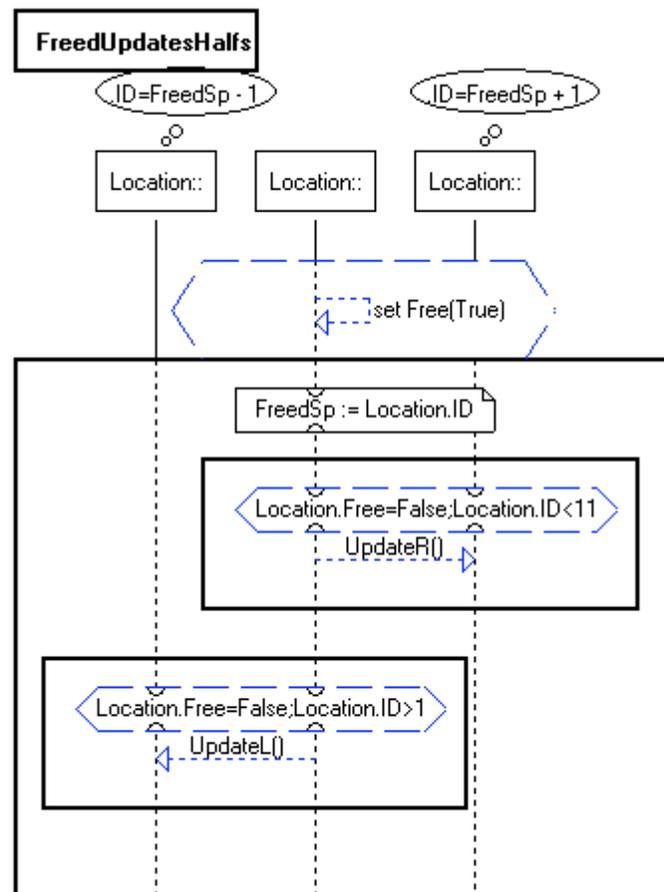


A Location's Residence keeps track of the identity of the VPC residing in that Location. After an ablation or cell movement, the Residence property is updated, triggering LSC "Box2MovingVPC". "Location" stores a particular value of LIN-3, based on the level of LIN-3 determined by the number of ACs and the position of the VPCs with respect to the AC(s); see LIN-3 use case for details. In the Box2MovingVPC LSC, a non-zero value in a Location is transferred to a VPC that has just moved to that Location, by setting of the VPC's LIN3 property value.



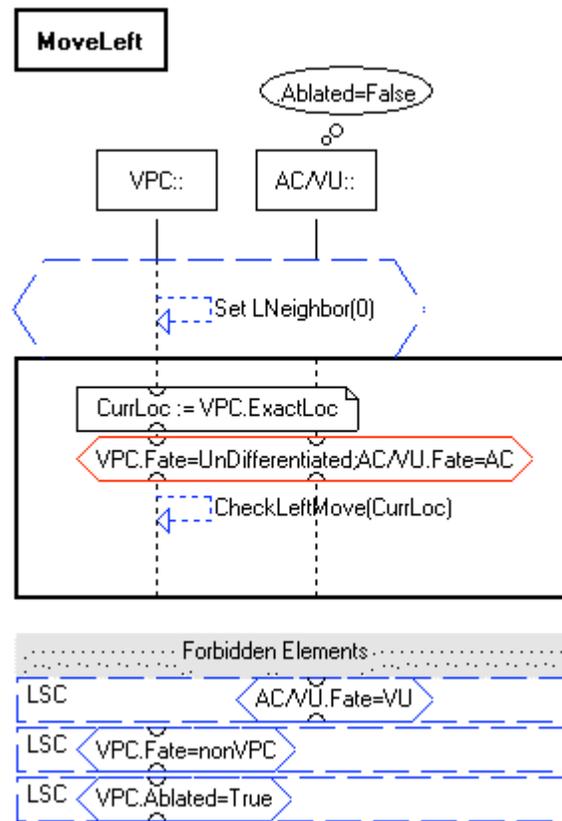
The LSC "ClearNeighbors" is activated by the execution of method `ClearNeighbors`, appearing in the main chart of LSCs "ClearAblatedLocation" and "BatchClearAblatedLocation". The LSC "ClearNeighbors" informs the cells neighboring a recently ablated cell that there is now a neighboring space. The Residence property of Location is used to ensure that a cell is informed properly that its right or left neighbor is now a space, rather than a cell. At the start of a simulation, the center of a cell occupies one Location and the

cell's left and right edges occupy half of each of the Locations to the left and right, respectively. Thus, the center of a neighboring cell is positioned two Locations away from the Location containing the original cell's center. These are the Locations bound by the symbolic instances in this LSC. The condition "Location.Residence>0" ensures that updating a neighboring cell's properties regarding the recently ablated cell only occurs when there is a neighboring cell, i.e., where the Location two units away from the ablated cell's center's Location has Residence other than 0. The subchart containing the condition "Location.ID<=9" deals with updating of properties of cells that are to the right of an ablated cell. Since there are only 11 Locations, it is sufficient to consider this subchart for ablated cells whose center is in Location 9 or smaller, since an ablated cell with Location larger than 9 has no right neighbors. Variable "lonelyR" refers to such cells that are right neighbors of an ablated cell. The logic is similar for the left neighbors of an ablated cell in the other subchart in this LSC. The condition "VPC.Ablated=False" ensures that a neighboring cell's properties are updated only when such a neighboring cell has not been ablated.

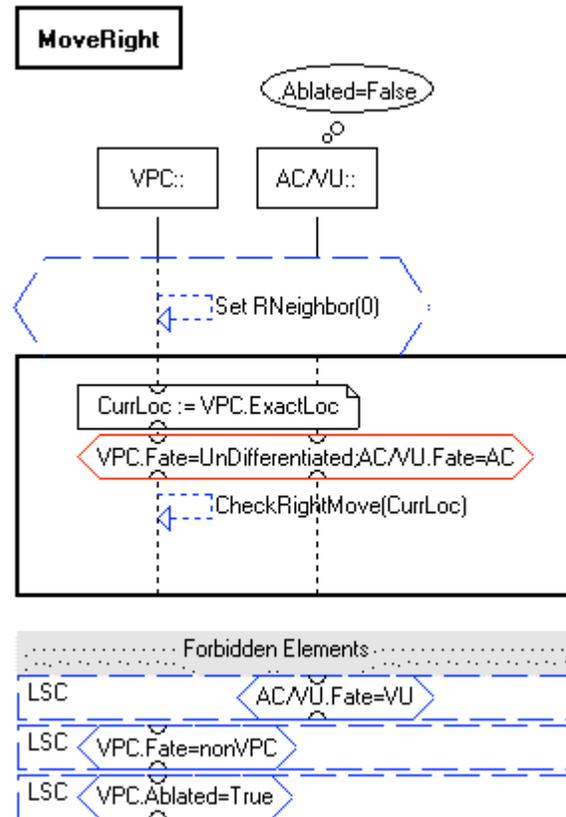


The LSCs "ClearAblatedLocation" and "BatchClearAblatedLocation" empty the Location that was the center of a recently ablated cell. Parts of the ablated cell were in other Locations, and those Location properties are updated accordingly. In the first subchart, as long as the center of the ablated cell was at least one Location unit away from the last Location unit on the right, then there is a Location to the right of the ablated cell whose properties should be

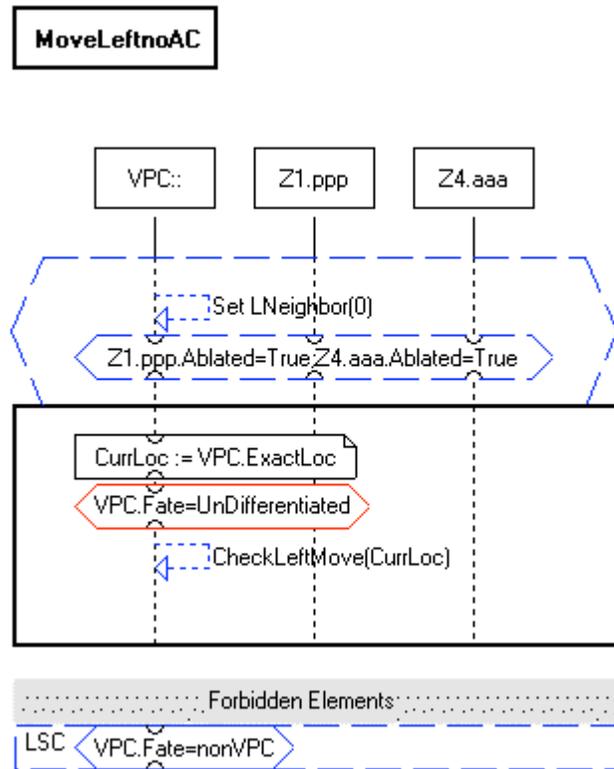
updated. The second subchart similarly specifies that if an ablated cell's center was positioned such that there was at least one Location to the left, then the Location to the left of the ablated cell's center's Location should be updated.



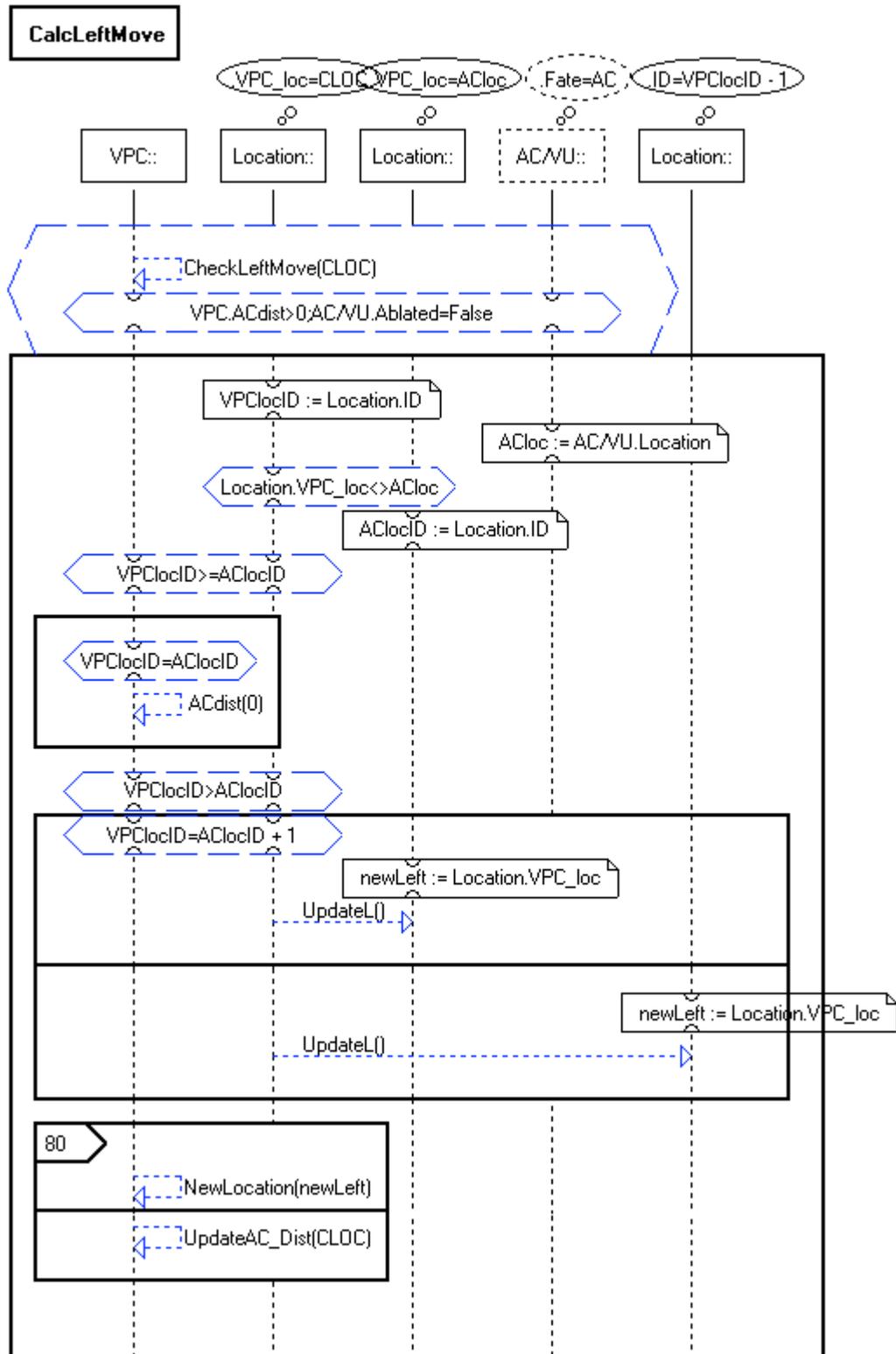
In LSC “MoveLeft”, if a cell is notified of a space to its left, the cell will consider moving one Location to the left (but may not actually move, depending on non-deterministic conditions set in the LSC “CalcLeftMove (Pn.p Movements)”). The cell that considers moving must be a VPC – part of the vulval equivalence group (excluding P3.p if it fused and became a nonVPC), and must not itself be already ablated. The movement potentially triggered in this LSC is only considered if the AC has not been ablated.



LSC “MoveRight” is symmetric to the LSC “MoveLeft”.



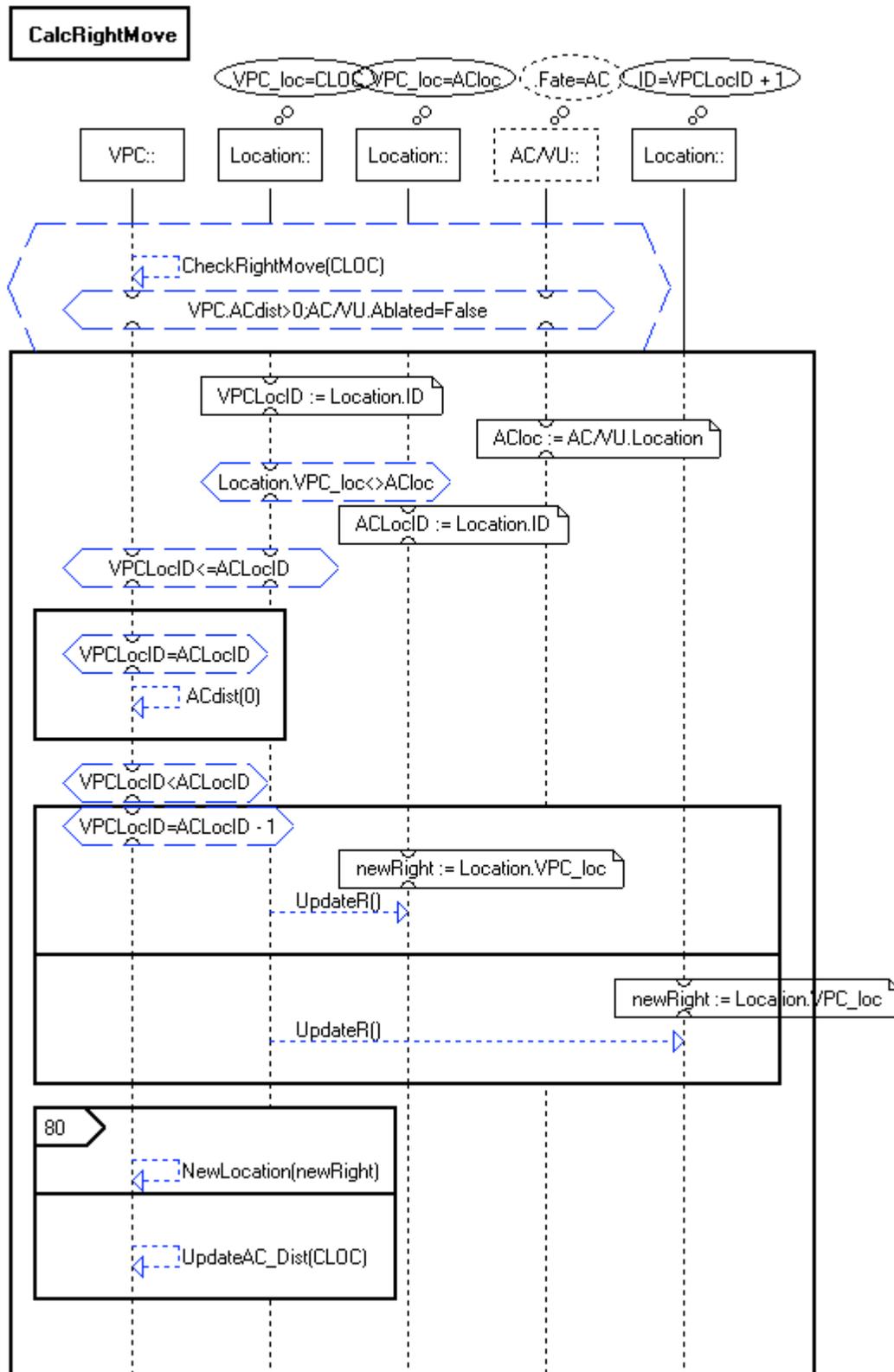
If the gonad has been ablated in addition to a VPC, VPCs may still move to fill the space left by the ablated VPC. However, the movement rules under such circumstances may differ from the movement rules for the case where the gonad is intact. The rules governing these movements have not been experimentally investigated systematically. The current model implementation, however, treats both cases in a similar way, where the movements in the absence of an anchor cell are towards the position of one of the AC/VU cells before it was ablated.



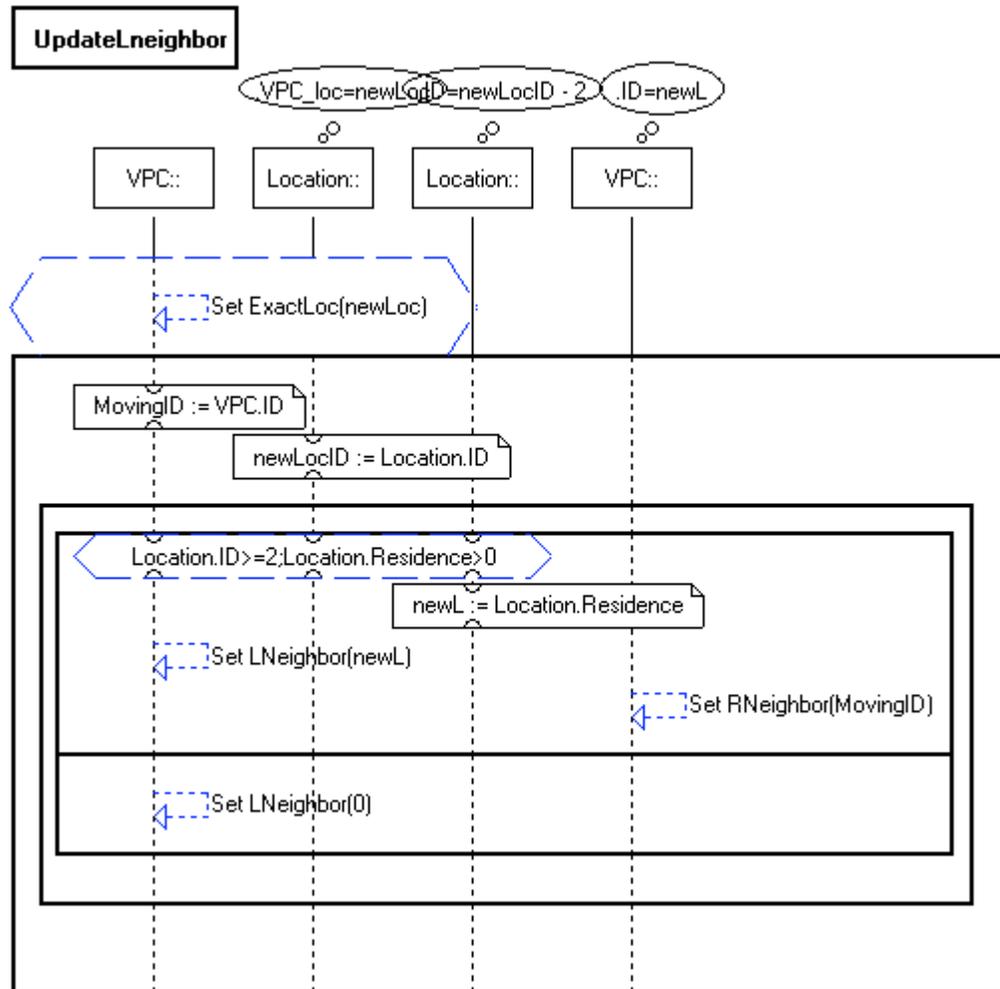
Whether or not a VPC moves to fill a space to its left depends upon a number of factors. This LSC is triggered only for VPCs that are not already located directly beneath the AC and

only if the gonad is not ablated. The VPC considering moving left then checks to see if it is to the right of the AC. The first subchart deals with the case in which a VPC's locID property is discordant with its ACdist property and updates the ACdist property appropriately.

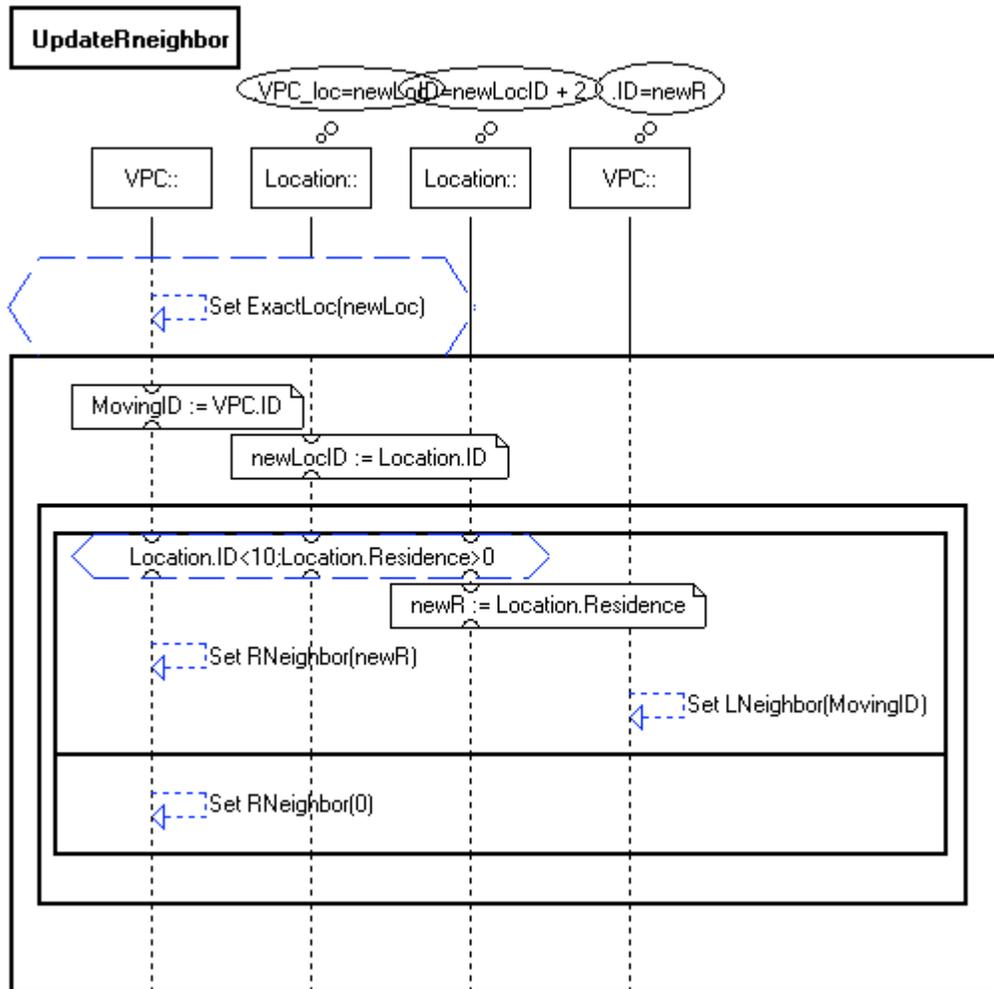
The rest of the chart holds only if the VPC considering moving is actually located to the right of the AC. "UpdateL()" triggers the updating of the neighbor property of the Location closest to the AC. If the VPC considering moving is located more than one Location unit to the right of the AC, then whatever Location is one unit to the left of the VPC considering moving should update its neighbor property to include a piece of the moving VPC. Either way, the Location one unit to the left of the VPC that is considering moving is stored. In 80% of the time, the VPC considering moving will move one unit to the left, by setting its NewLocation to match the stored Location. In the other 20% of the time, the VPC does not move and only updates its ACdist property. CLOC refers to the current Location of the VPC considering moving.



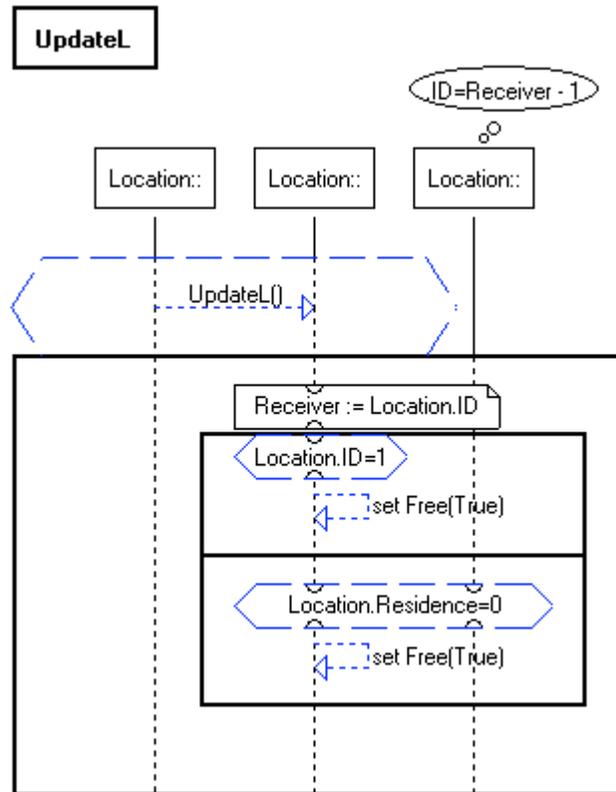
LSC “CalcRightMove” is symmetric to LSC “CalcLeftMove”.



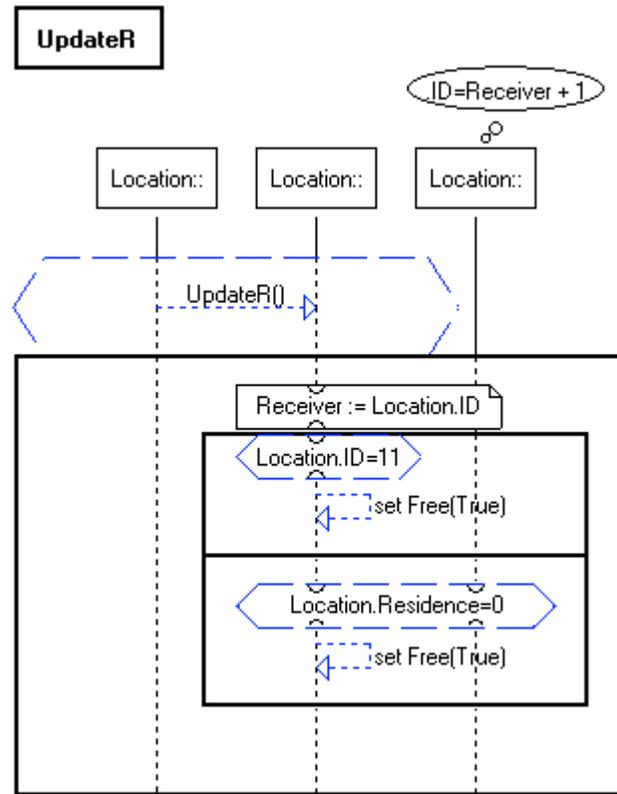
The LSC “UpdateLneighbor” specifies that once a cell moves by setting its ExactLoc property to a newLoc, if the new Location is at least two units away from the leftmost Location, then the moving VPC may, once it has moved, have a left neighbor. The “Residence>0” condition guarantees that there is indeed a cell two Locations away from the moving VPC’s destination and thus that the moving VPC will have a left neighbor. The moving VPC takes note of its left neighbor and the left neighbor takes note of the moving VPC. If the moving VPC will not have a left neighbor, it takes note of that.



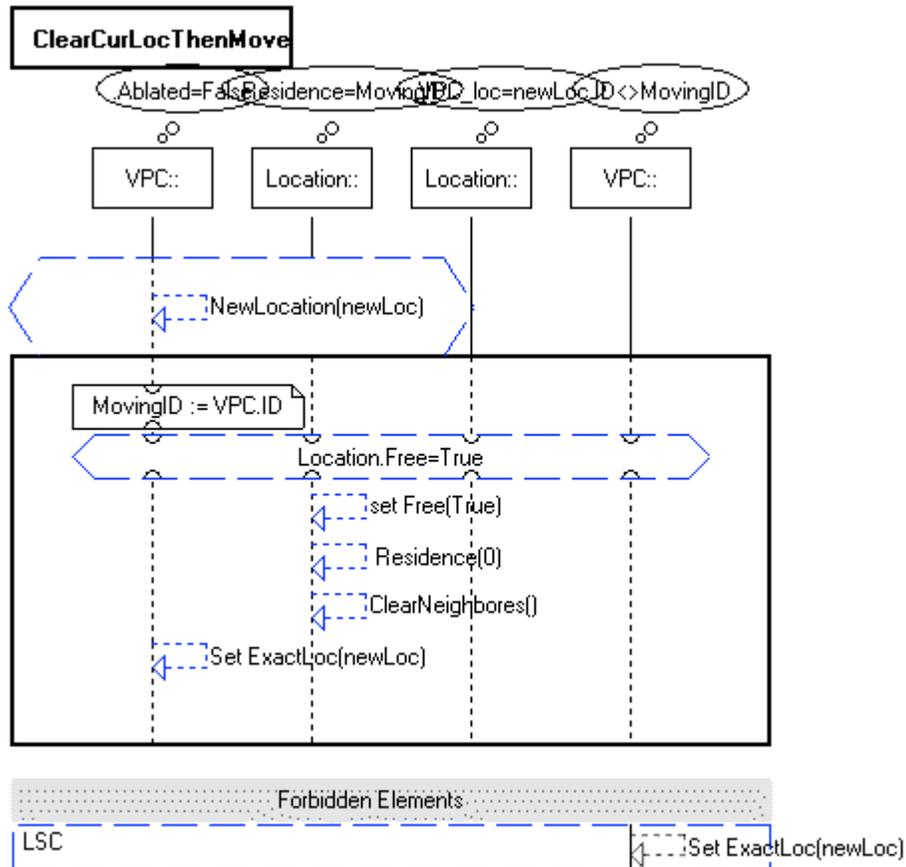
LSC “UpdateRneighbor” is symmetric to LSC “UpdateLneighbor”.



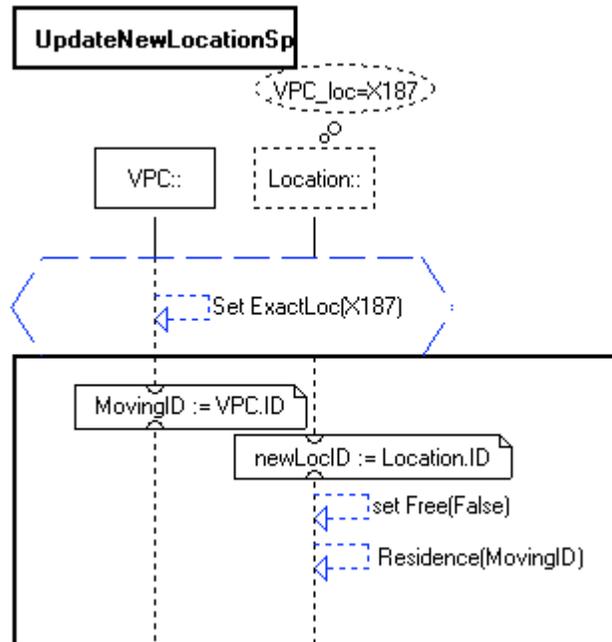
LSC “UpdateL” allows continued leftward movement of a VPC that has just moved left. A VPC can move left again if it is in P3.p/P4.pLocation or if it is in another location and there is no adjacent VPC to its left. The first part of the subchart accounts for the former situation. P3.pLocation’s ID is 1 and if a cell’s center is in P3.p/P4.p then P3.pLocation, must have no resident and thus its Free property should temporarily be set to True to allow potential movement. The second part of the subchart accounts for the latter situation in which a cell is in any location other than P3.pLocation and P3.p/P4.pLocation and is considering moving left. The Location one unit to the left of the VPC’s current Location is temporarily considered Free to allow for potential movement if the Location one unit further to the left has no resident (or cell center).



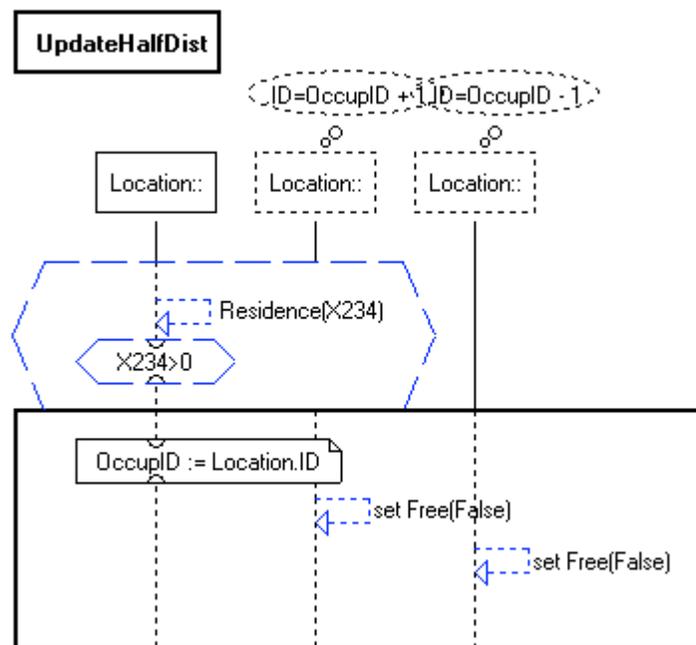
LSC “UpdateR” is symmetric to LSC “UpdateL”.



When a VPC moves, this LSC first checks that the Location the VPC is moving to is indeed Free. This LSC updates the properties of the Location formerly containing that VPC's center and triggers the updating of the properties of the neighboring Locations. Finally, this LSC completes the VPC's move by setting the VPC's ExactLoc property to the destination value. The Forbidden Element ensures that no two cells simultaneously move, or reset their ExactLoc properties. This prevents two cells from occupying the same Location at the same time, a physical impossibility.



This LSC informs the newly occupied Location of its new contents. If a VPC moves to a Location, that Location is no longer free and now contains that VPC.

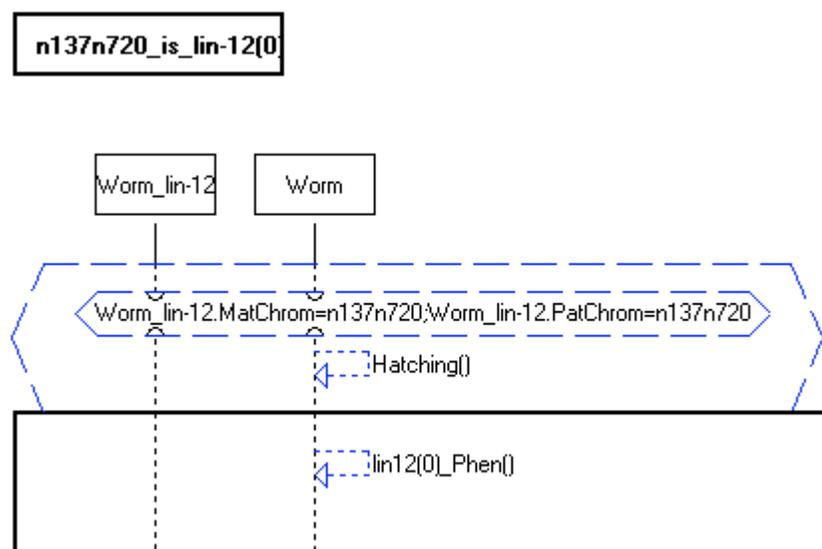


A Location is Free only when it has no contents whatsoever; if a Location contains cell edges, it is not Free. This LSC tells the Locations one unit to the left and right of the Location containing a new cell center that they are not free.

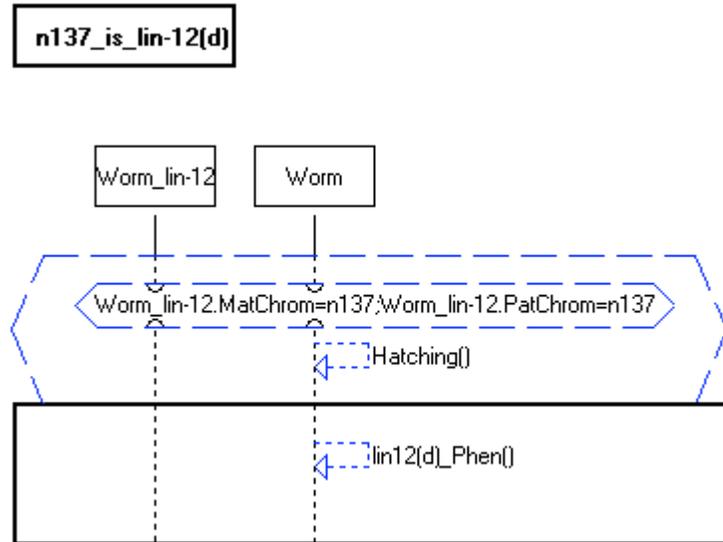
## Lateral Signaling

This use case describes the lateral signaling between neighboring VPCs (Greenwald 1983; Sternberg 1988). This signaling induces the receiving VPC not to become primary and to later become secondary. The actual response to the signal is described in use case VPC Fate Assumption.

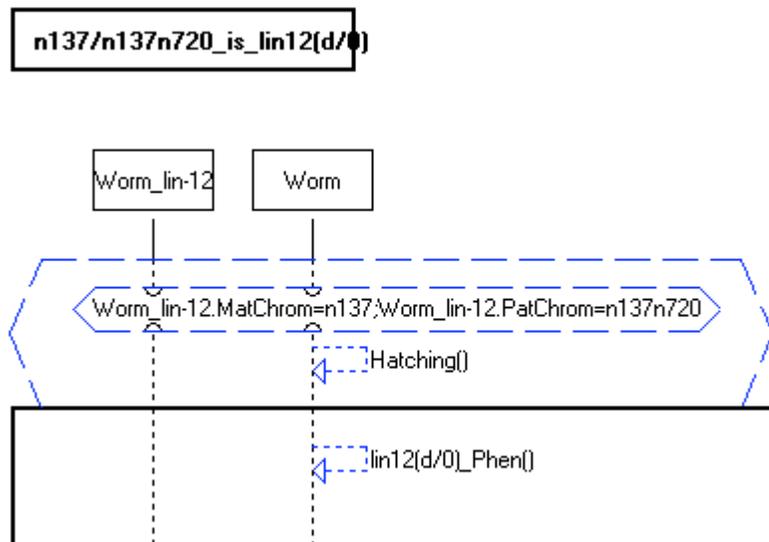
The first 4 LSCs described in this use case translate from specific genotypes to general phenotype methods. The general phenotype methods can then be used for a more concise behavioral description, e.g., in use case AC/VU.

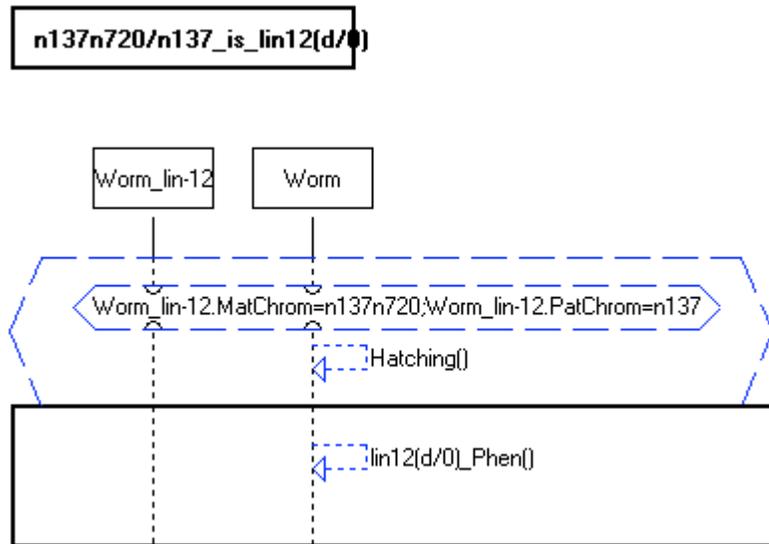


LSC "n137n720\_is\_lin-12(0)" specifies that a `lin12(0)_Phen` method is executed if both maternal and paternal chromosomal copies of *lin-12* have allele value n137n720 before hatching.

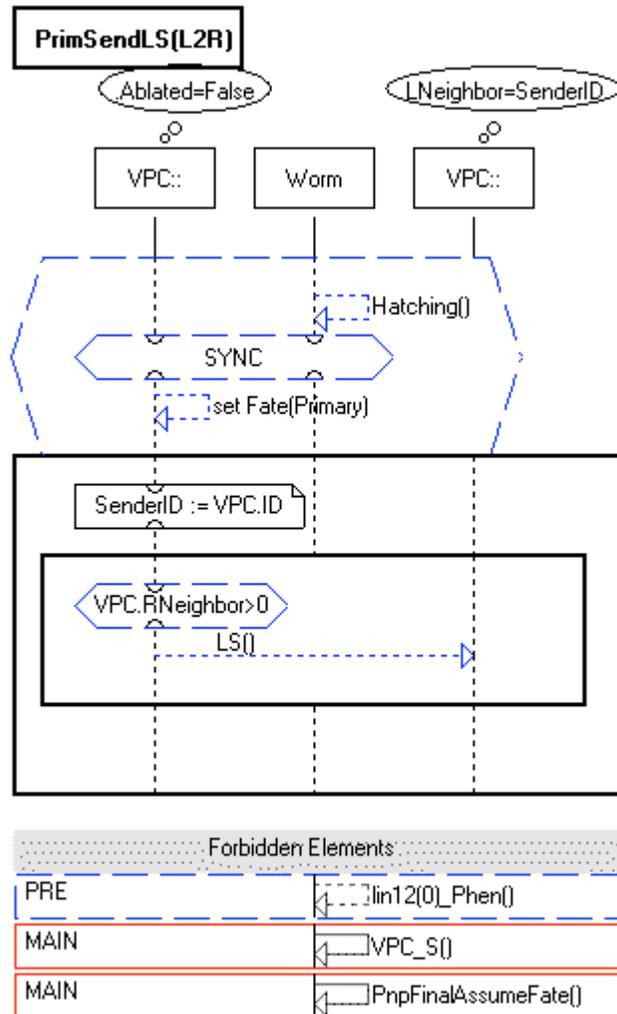


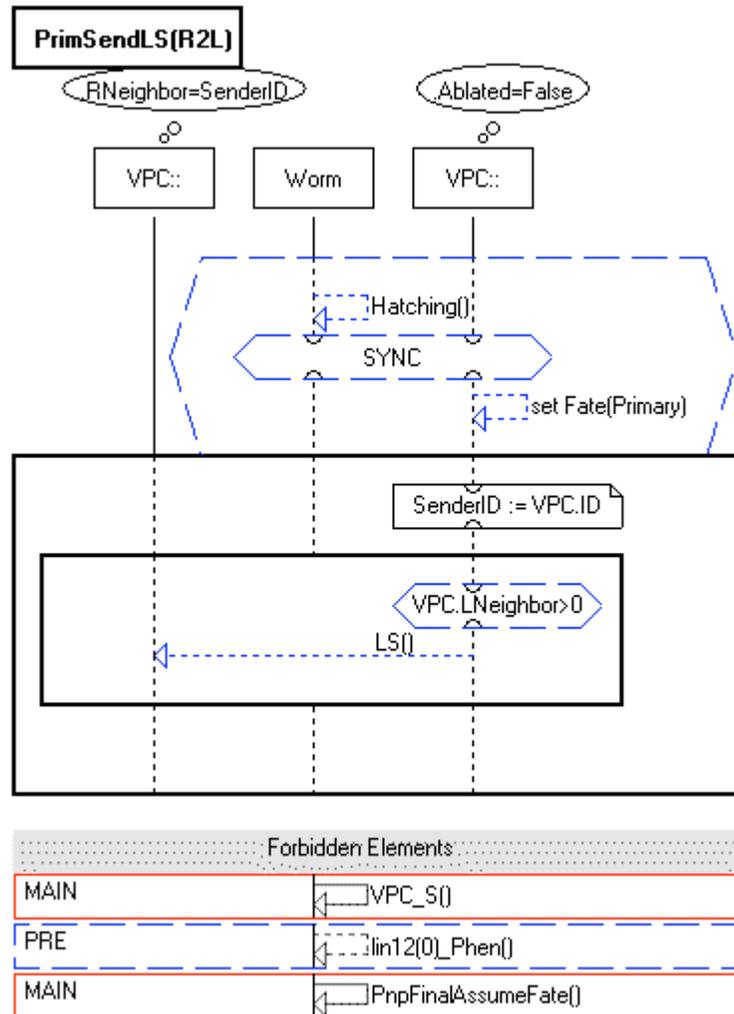
LSC “n137\_is\_lin-12(d)” specifies that a lin12(d)\_Phen method is executed if both maternal and paternal chromosomal copies of *lin-12* have allele value n137 before hatching.





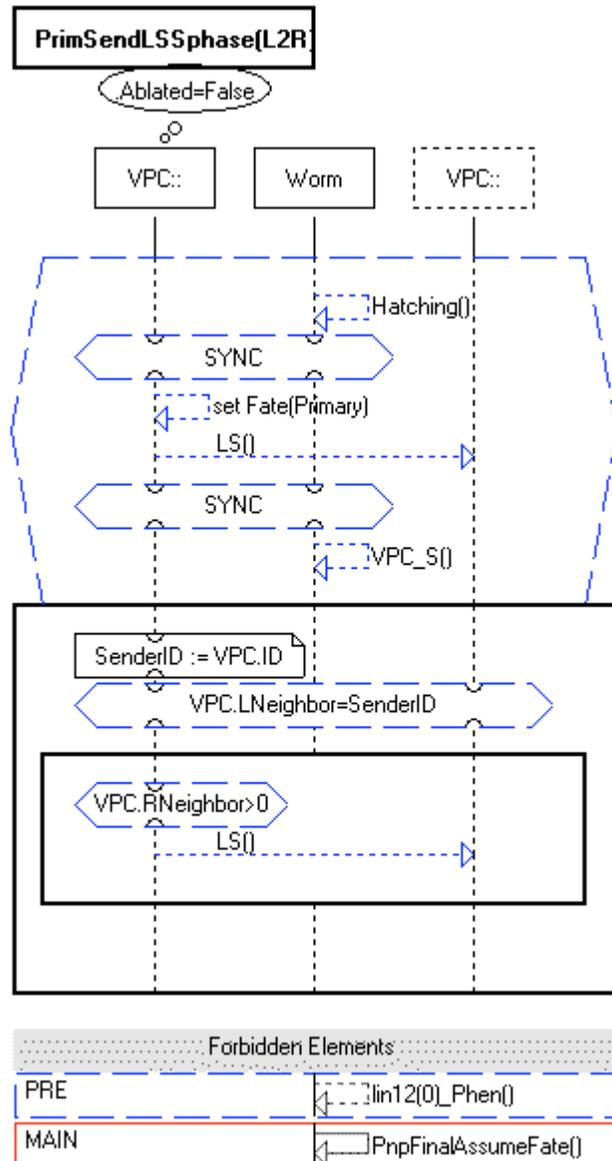
LSCs “n137/n137n720\_is\_lin-12(d/0)”, “n137n720/n137\_is\_lin-12(d/0)” specify that a `lin12(d/0)_Phen` method is executed if one of the chromosomal copies of *lin-12* has allele value n137 while the other has allele value n137n720.

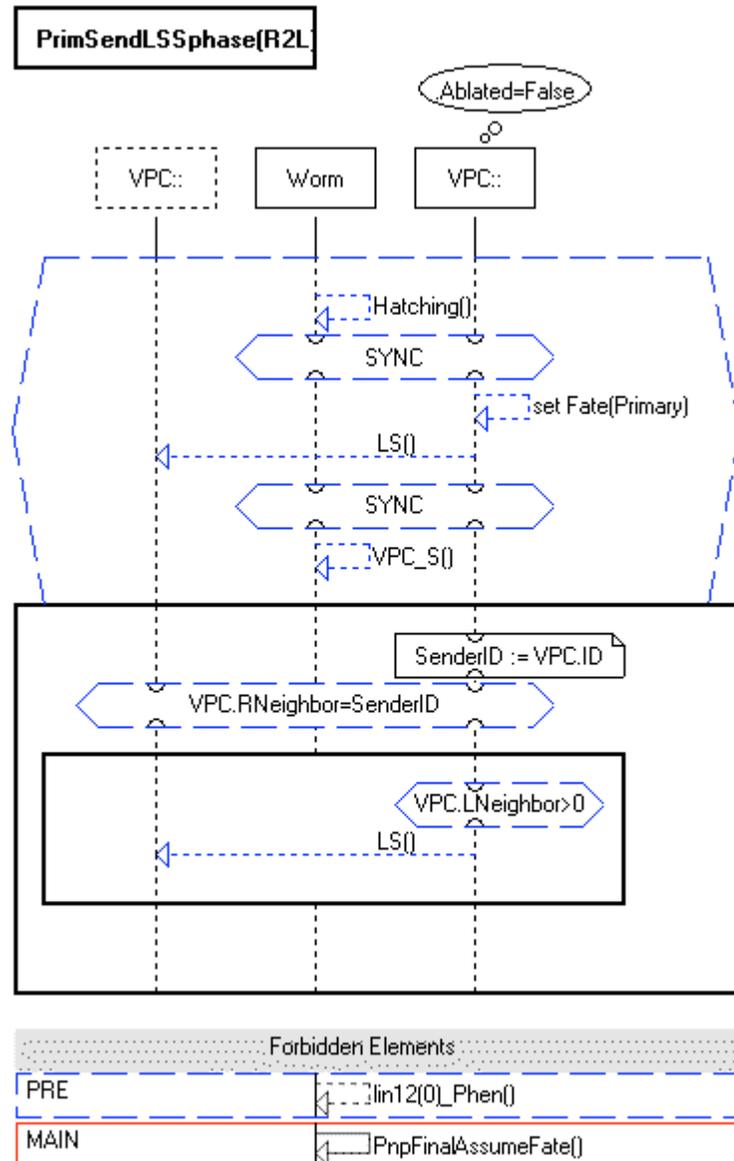




Lateral signaling specifies VPC fate in two stages. Lateral signal “heard” by a VPC before the cell enters into S phase (DNA replication) of the cell cycle causes the cell to adopt a non-primary fate. Lateral signal received after S phase promotes adoption of a secondary fate (Ambros, 1999).

LSC “PrimSendLS(L2R)” specifies that a VPC that becomes primary sends a lateral signal corresponding to the method “LS” to its right neighbor (Simske and Kim, 1995). In a similar way LSC “PrimSendLS(R2L)” send a lateral signal to the left neighbor. The “LS” method will be executed in these LSCs before S phase, this ordering is achieved by the method “VPC\_S” appearing as a hot forbidden message with the main chart as its scope, thus once the chart is active it must complete successfully including the execution of “LS” before “VPC\_S” occurs as specified in LSC “PnpAssumeFate(VPC Fate Assumption)”.

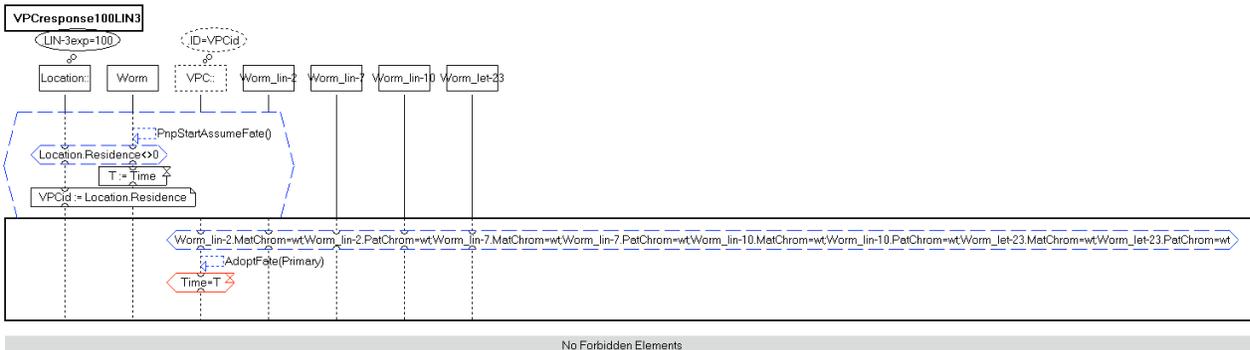




The LSC “PrimSendLSSphase(L2R)” specifies that a VPC that sent a lateral signal to its right neighbor before S phase will send a second round of lateral signal after S phase. The response to the lateral signal in each of the rounds is different, as specified in use case VPC Fate Assumption. A similar behavior for sending the lateral signal to the left neighbor is specified in LSC “PrimSendLSSphase(R2L)”.

## VPC Fate Assumption

The use case VPC Fate Assumption describes the specification and determination of VPC fates. To facilitate concise modeling, we define a class VPC, and 6 instances of this class, P3.p, P4.p, P5.p, P6.p, P7.p and P8.p. The notion of a class and its instances in the object-oriented sense, matches the biological intuition of viewing the VPCs as an equivalence group.



In “VPCresponse100LIN3” a VPC that senses a high LIN-3 signal adopts a primary fate. In the wild type this will be P6.p which is closest to the anchor cell and thus induced to become primary. The modeling of LSC “VPCresponse100LIN3” is general enough to handle many other cases besides the wild type, including *dig-1* or *lon-1* in which P6.p may not necessarily be the closest VPC to the anchor cell, or VPC ablations and movements which effect the positioning and LIN-3 signal level for the VPCs, and also *lin-12(0)* which results in 2 anchor cells and potentially higher LIN-3 signal levels also to P5.p and P7.p.

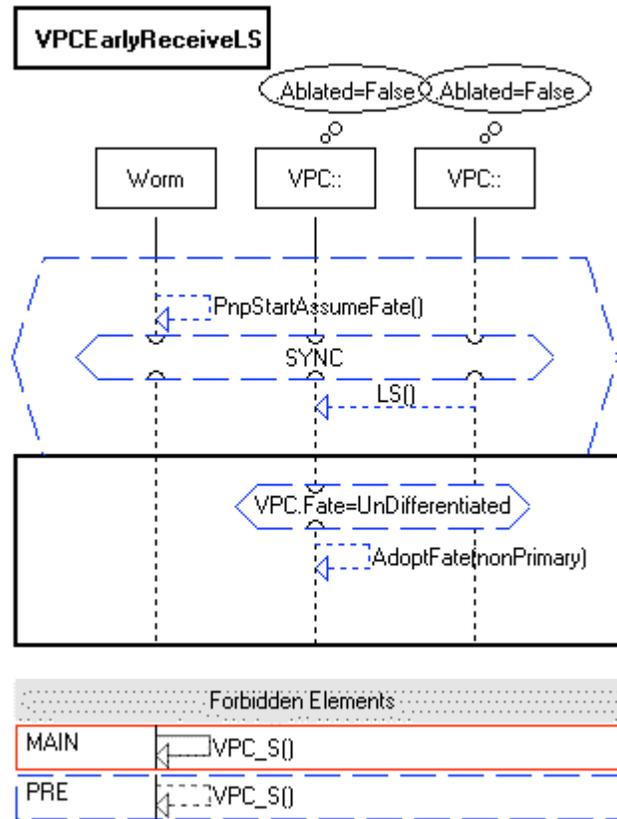
In the prechart, after PnpStartAssumeFate is executed, Locations for which the LIN-3exp is 100 (100 representing a high level) become bound, and store in variable VPCid the ID of the resident VPC in this Location, which then adopts a primary fate without any time delay. The cold condition in the main chart specifies that *let-23*, which encodes the receptor of the inductive signal (Aroian et al., 1990), is wild-type. That condition also specifies *lin-2*, *lin-7*, and *lin-10* as WT for the primary fate assumption to occur as a result of the LIN-3 signal. This represents the hypothesis that *lin-2*, *lin-7*, and *lin-10* promote membrane localization of LET-23 (Kaech et al., 1998).



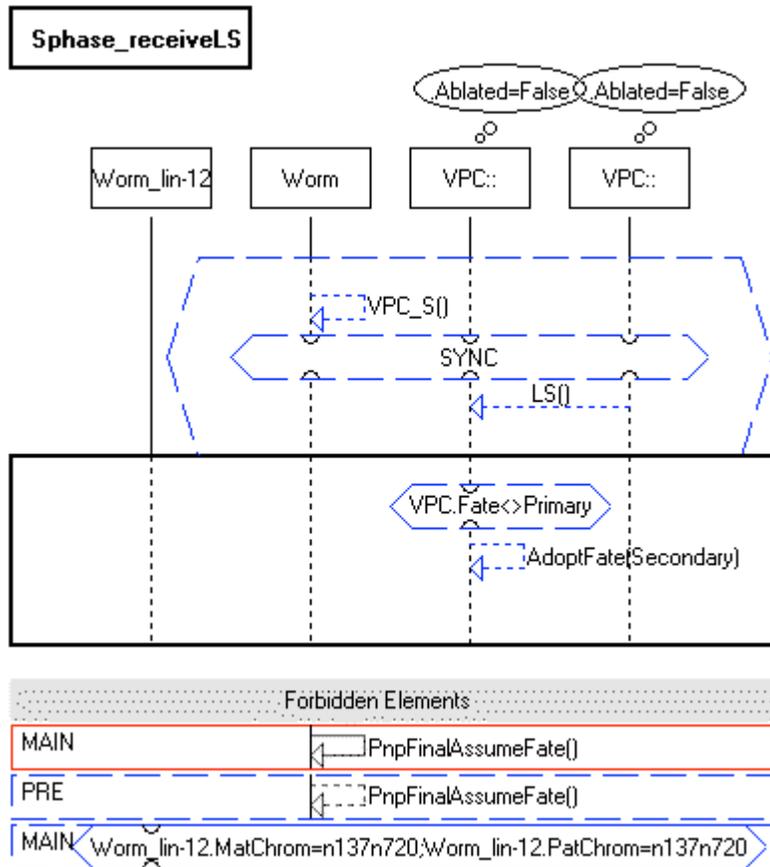
We have incorporated in our model two mechanisms for acquiring a secondary fate, one mechanism by a VPC receiving a medium LIN-3 signal, as specified in LSCs “VPCresponse50LIN3”, “VPCresponse50LIN3B”, and the second mechanism by a VPC receiving a lateral signal from a primary neighbor as specified by LSCs “VPCEarlyReceiveLS”, “Sphase\_receiveLS”. These two mechanisms correspond to the *Graded Signaling* vs. *Sequential Signaling* hypotheses, which he have examined by testing three models: the Core Model (contains all 4 LSCs mentioned above), the Graded Signaling model which is obtained by removing the two LSCs “VPCEarlyReceiveLS”, “Sphase\_receiveLS”, and the Sequential Signaling Model obtained by removing the two LSCs “VPCresponse50LIN3”, “VPCresponse50LIN3B”.

In “VPCresponse50LIN3” a VPC that senses a medium LIN-3 signal adopts a secondary fate. In the wild type this will apply to P5.p and P7.p which are at distance one unit from the anchor cell and thus receive a medium LIN-3 signal as described in use case LIN-3. This will occur after receiving a LIN3(50) signal (50 representing a medium signal) and only after S phase. Another condition is that the VPC is still Undifferentiated. The secondary fate is assumed before the execution of PnpFinalAssumeFate, which appears as a hot forbidden message. Due to this ordering a VPC receiving a medium LIN-3 signal will execute a secondary fate and avoid assuming a default tertiary fate as specified in LSC “GroundStateNonVulval”. The LSC “VPCresponse50LIN3B” is identical except for the condition that the VPC is nonPrimary instead of Undifferentiated. This LSC is needed since lateral signaling will induce P5.p and P7.p in the wild type to first become non-Primary and only later it can become secondary. In the Core model, for the wild-type case, the method AdoptFate(Secondary) will be executed once for each

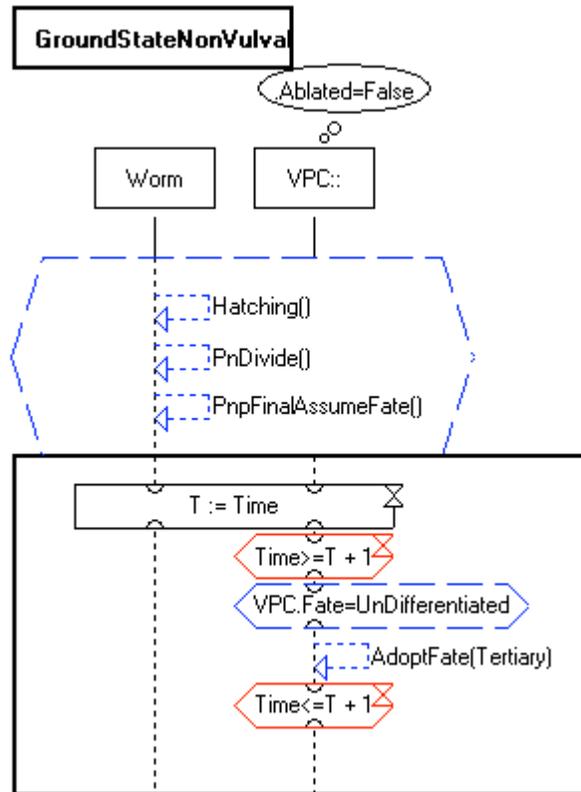
of the VPCs P5.p and P7.p as a result of the main charts of both “VPCresponse50LIN3B” and “Sphase\_receiveLS”.



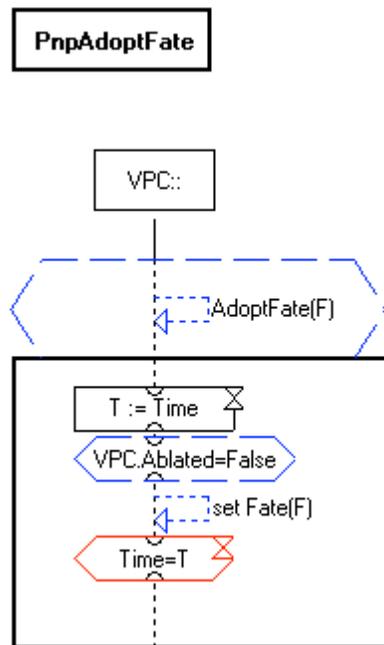
In “VPCEarlyReceiveLS” a VPC that receives a lateral signal (“LS”) after the PnpStartAssumeFate stage (in LSC “PnpAssumeFate”) and is still undifferentiated adopts a nonPrimary fate. Thus the lateral signaling first prevents a VPC from becoming primary, and only later (after S phase) induces secondary fates as specified in LSC “Sphase\_receiveLS”.



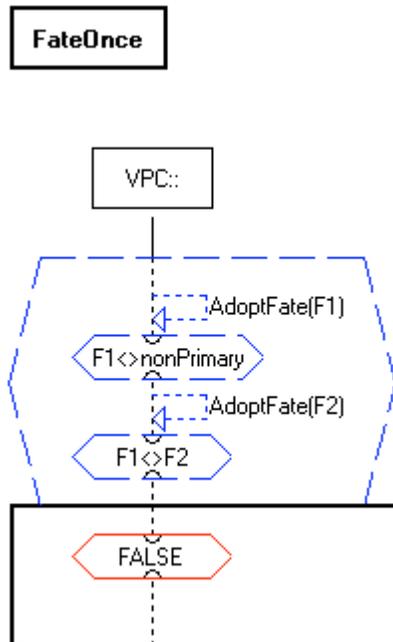
In “Sphase\_receiveLS” a VPC that receives a lateral signal (“LS”) after S phase (in LSC “PnpAssumeFate”) and is not primary adopts a secondary fate.



The LSC “GroundStateNonVulval” specifies that a VPC that is not ablated and is undifferentiated by the PnPFinalAssumeFate stage, adopts a tertiary fate. Vulval fates (primary or secondary) are adopted prior to PnPFinalAssumeFate, and if by then a VPC is still undifferentiated it will by default adopt a tertiary fate.



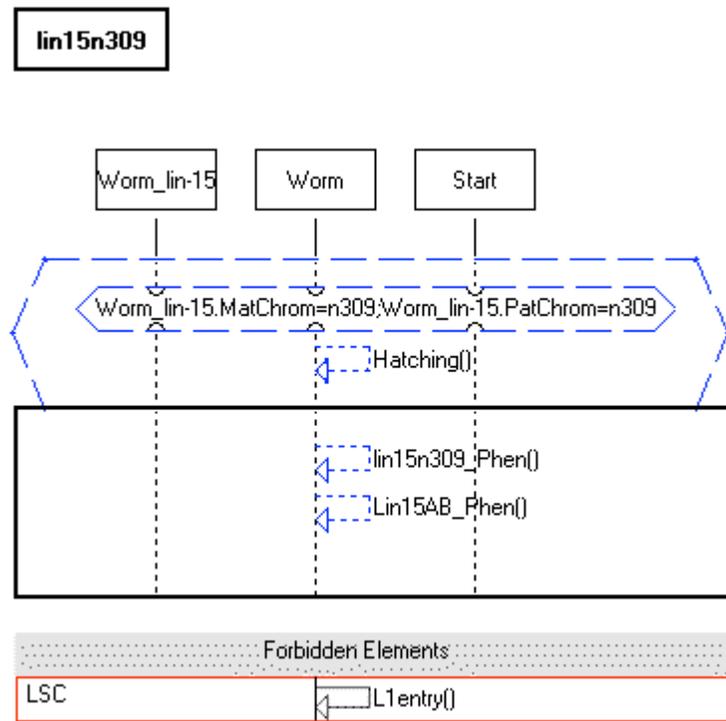
The LSC “PnpAdoptFate” specifies that for every VPC instance that executes the AdoptFate(F) method, the Fate property will be set to the value of F by executing the setFate(F) message. Fate is assigned in two steps, first executing the method AdoptFate (e.g., AdoptFate(Primary) in “VPCresponse100LIN3”) will activate “PnpAdoptFate”. Next, the execution of the setFate message (in the “VPCresponse100LIN3” example setFate(Primary)) will be executed. It is the second step that causes the actual change in the value of the property. This two step fate assignment does not reflect a biological phenomenon, but rather a common programming pattern, whereby a method is called that, in turn, invokes another message that is responsible to perform the actual work. The two steps are required in this case to permit the “PnpAdoptFate” to evaluate a condition that ensures that the VPC is not ablated before executing the setFate message. Thus the two phase fate assignment prevents an ablated VPC from acquiring a fate (even if another LSC executes the adoptFate method without having checked whether the VPC was ablated). A similar approach is used for fate acquisition of AC/VU cells in “ACAdoptFate (AC/VU)”.



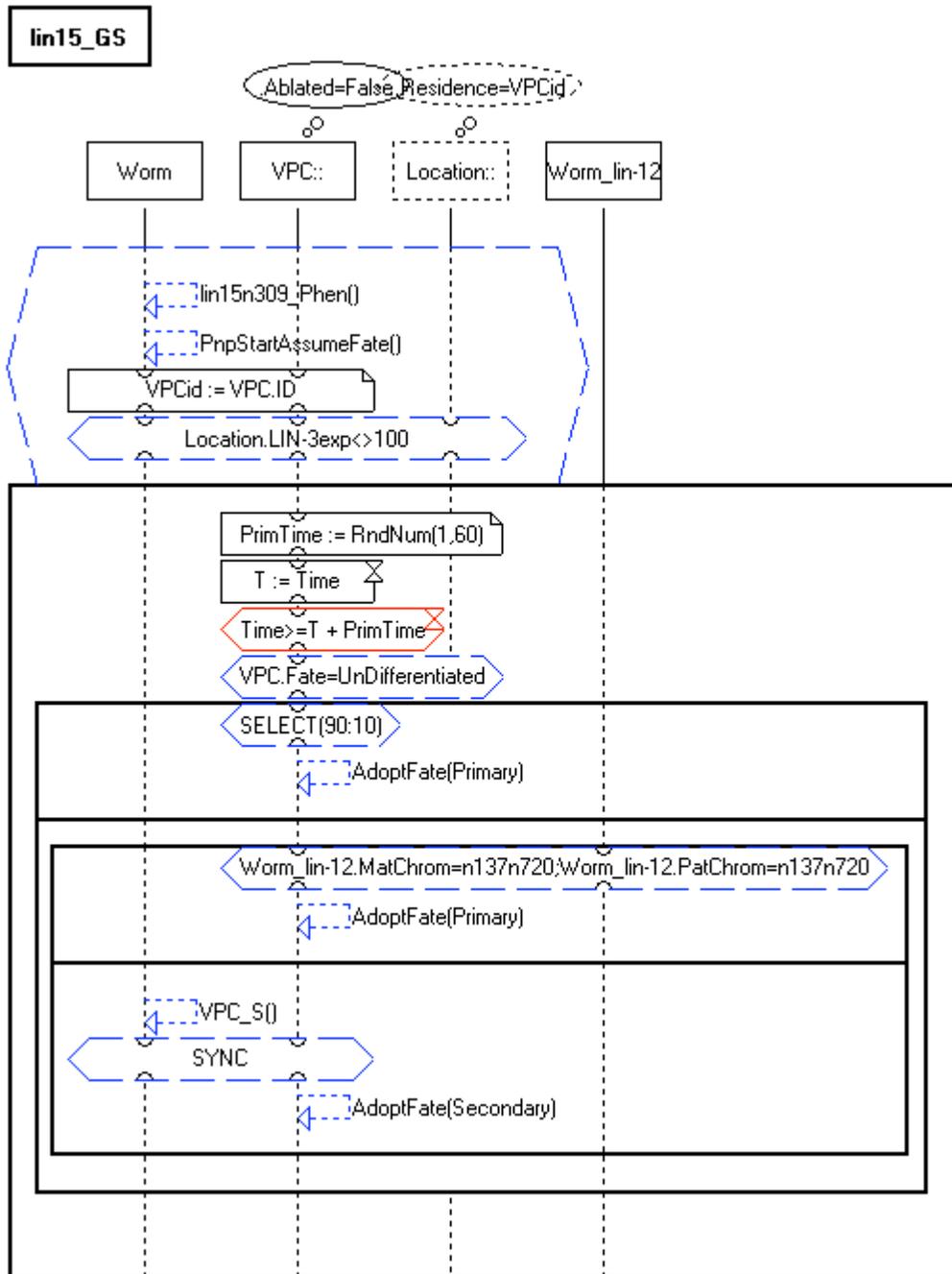
The LSC “FateOnce” specifies that the same VPC should not acquire two different fates. This is an example of an anti-scenario, where the forbidden scenario is described in the prechart, while the main chart contains the hot condition FALSE, which can never be satisfied, thus the scenario described in the prechart should never happen. The current play-out implementation cannot ensure that the specified anti-scenarios do not occur, but instead during execution any such anti-scenario violation is flagged and the user is alerted (if running with charts open, the chart will appear with a red X and the chart will remain open until the user allows the run to proceed; if running in batch mode or in manual mode, the violation will be listed in the log file). As part of the model testing, we focused on increasing the confidence that anti-scenarios do not happen. In LSC “FateOnce” there is an exception to the rule that different fates should not be adopted, for the case where AdoptFate(nonPrimary) is the first fate adopted. This is due to the fact that in the wild-type the lateral signaling induces P5.p and P7.p to first become nonPrimary (before S phase) and later Secondary (after S phase).

## Hyp7 Inhibitory Signal

This use case specifies model behavior in a *lin-15* background (Sternberg, 1988).



LSC "lin15n309" translates from a specific genotype to general phenotype methods. The general phenotype methods can then be used for a more concise behavioral description, e.g., in LSC "lin15\_GS". Similar LSCs specify general phenotypic classes for *lin-12* mutations. The advantage of this generalization is that other alleles that behave similarly can be readily mapped to this behavior. LSC "lin15n309" specifies that methods `lin15n309_Phen`, `Lin15AB_Phen` are executed if both maternal and paternal chromosomal copies of *lin-15* have allele value `n309` before hatching.

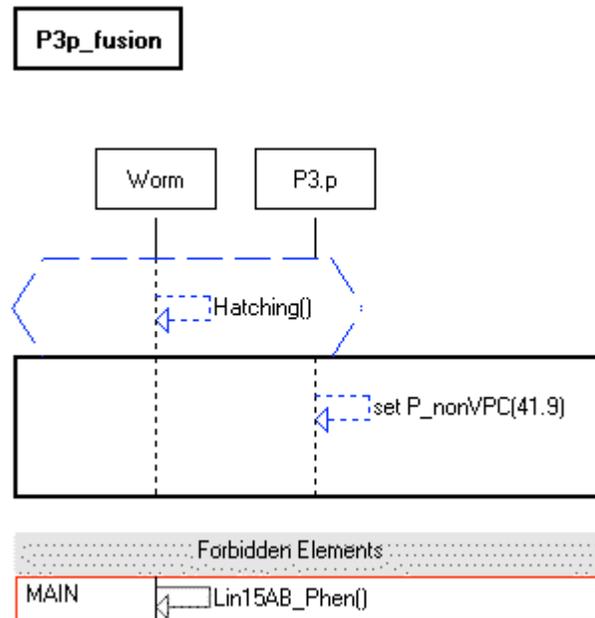


The LSC “lin15\_GS” is activated in a *lin-15* background as specified by the message “lin15n309\_Phen” appearing in the prechart. Any VPC that does not get a high level of LIN-3 signal will have a chance to compete in a “race” to become primary. Each such VPC chooses a random time to acquire a fate within a one hour interval after message “PnPStartAssumeFate” is executed. If at this random time point, which is selected as a number between 1 and 60 (representing minutes), the VPC is still undifferentiated, in 90% of the cases the VPC will set its fate to primary, while in the remaining 10% of the cases the VPC will set its fate to secondary after S phase, unless this is in a *lin-12(0)* background in which secondary fates are usually not observed (except for one animal in SH89 T1C 3). The choice to become secondary allows the

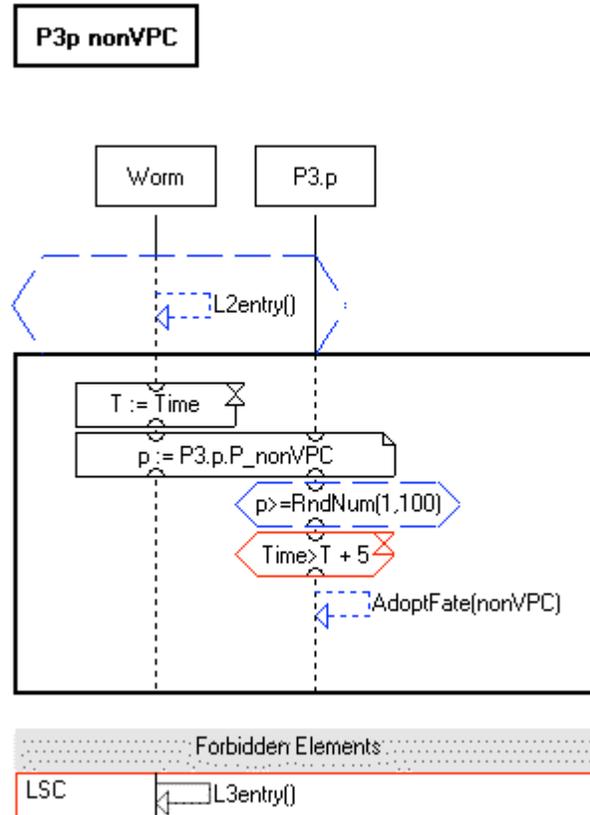
model to reproduce results in which an isolated VPC becomes secondary without receiving a medium LIN-3 signal (see text for additional discussion of this implementation).

## Pn.p Fusion

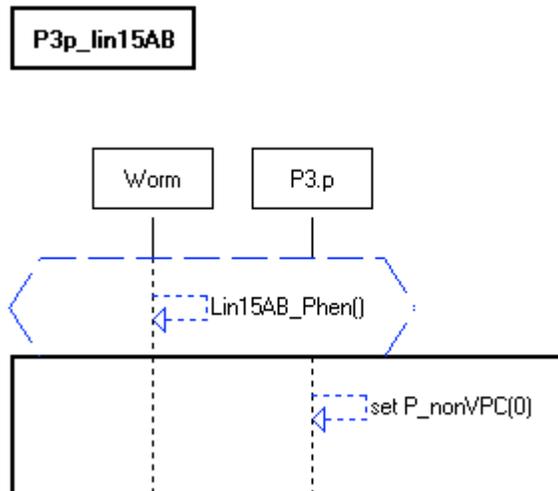
This use case deals with the potential of P3.p to fuse with the surrounding hypodermal syncytium (hyp7) (Chen and Han, 2001; Clandinin et al., 1997). This is a non-deterministic decision: in the wild type P3.p fuses in 41.9% of the animals.



LSC "P3p\_fusion" sets the property P\_nonVPC representing the probability of fusing and thus adopting a nonVPC fate for P3.p.



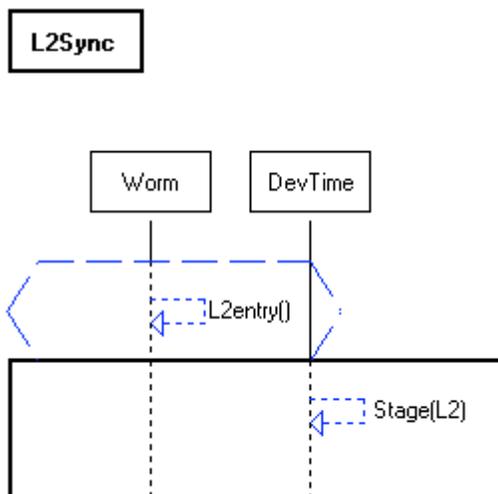
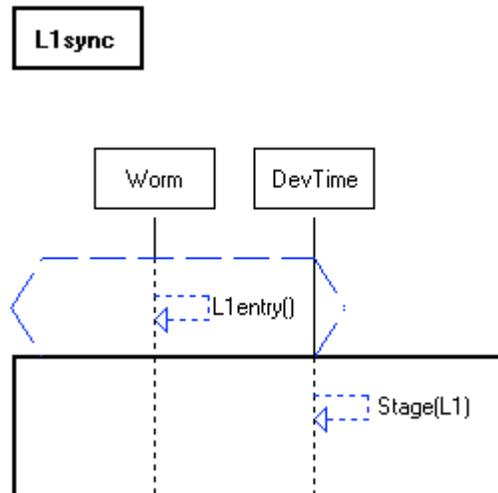
The LSC “P3p nonVPC” specifies the non-deterministic decision of P3.p to fuse and adopt a nonVPC fate. This decision is made after 5 hours in L2 stage. The function `RndNum(1,100)` generates a random number between 1 and 100, if this number is smaller or equal to the property `P_nonVPC` then P3.p executes the `AdoptFate(nonVPC)` method which will activate LSC “PnpAdoptFate” to actually set the fate of P3.p to nonVPC. Otherwise, the chart is exited, and P3.p remains Undifferentiated and will later assume a fate according to the LSCs in use case VPC Fate Assumption.

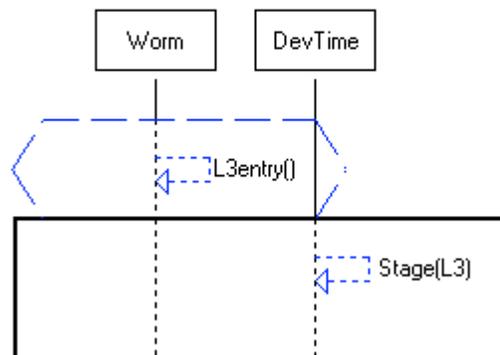
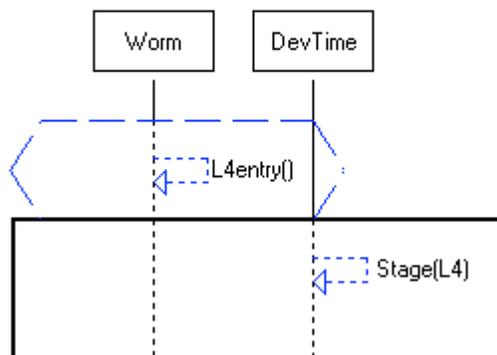
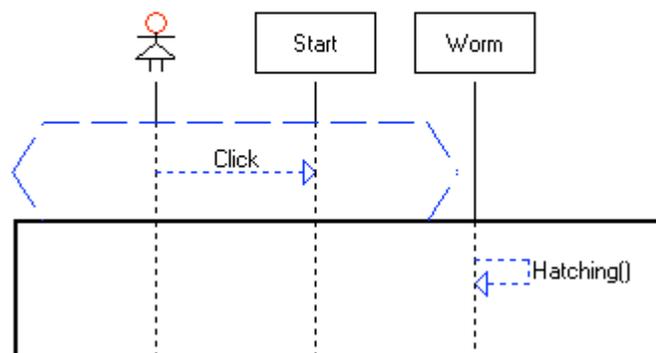


According to the experimental data in a *lin-15* background P3.p adopts either a primary or secondary fate, but does not fuse with Hyp7. To model this fact, LSC “P3p\_lin15AB” sets the probability of P3.p fusing to 0. The effect of other genetic mutations on the fusion decision is not considered in the current version of the model.

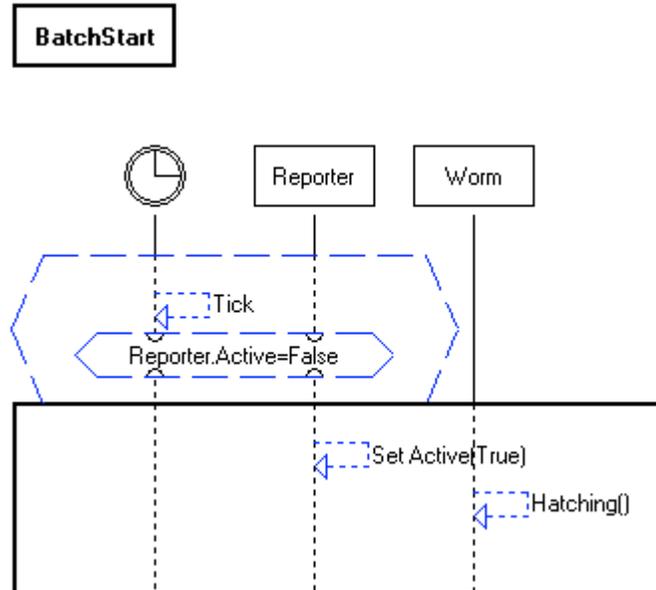
## Mechanics

This use case deals with some technical aspects that allow visualization and user interaction with the model, rather than the actual biological modeling. The DevTime object is used to show the time progress and larval stage during development, in LSCs “L1sync”, “L2sync”, “L3sync” and “L4sync” the stage property of DevTime is updated according to the entry of the corresponding larval stage in the worm.

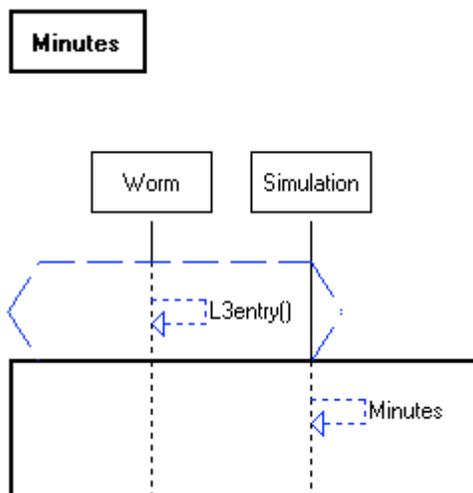
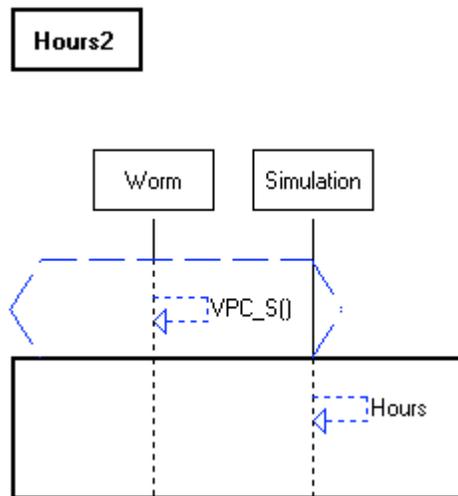
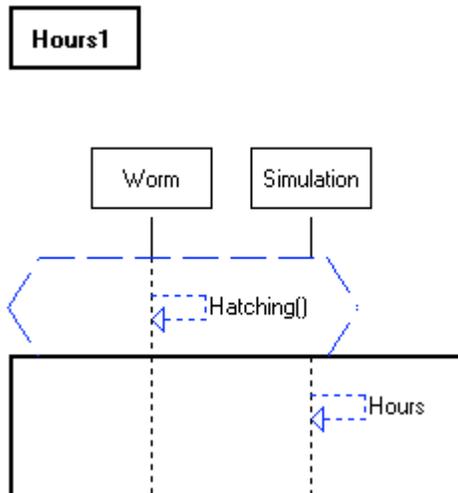


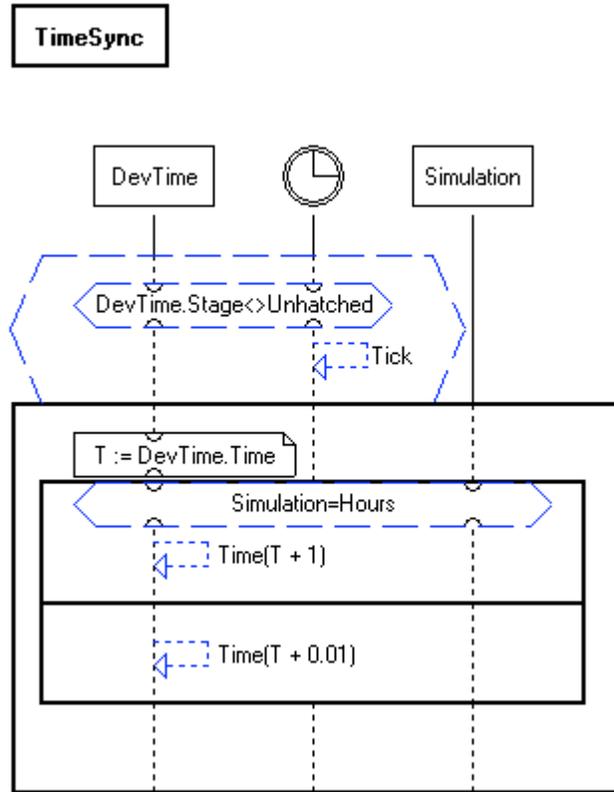
**L3sync****L4sync****Start**

Our model supports two main execution modes, interactive mode in which the user actively interacts with the GUI to run the simulation, and batch mode in which a set of predefined experiments is performed. In the first mode, the user clicks the start button to begin a run, which results in the Hatching method being executed as specified in LSC “Start”, which triggers the worm development as specified in “DevelopmentalTime20c (Core Behaviors)”.



In batch mode the user does not interact with the GUI and thus LSC “Start” is not activated, instead LSC “BatchStart” is triggered by a time tick, since time is set to progress automatically by the Play-Engine in batch mode. A special object, the Reporter is used in batch mode, it has a Boolean property Active which is initially set to FALSE, thus the condition in the prechart holds after the first time tick. The main chart is then executed, setting the active property to TRUE (so this chart will not be reactivated in each of the following time ticks), and executing the Hatching method which triggers the worm’s development.

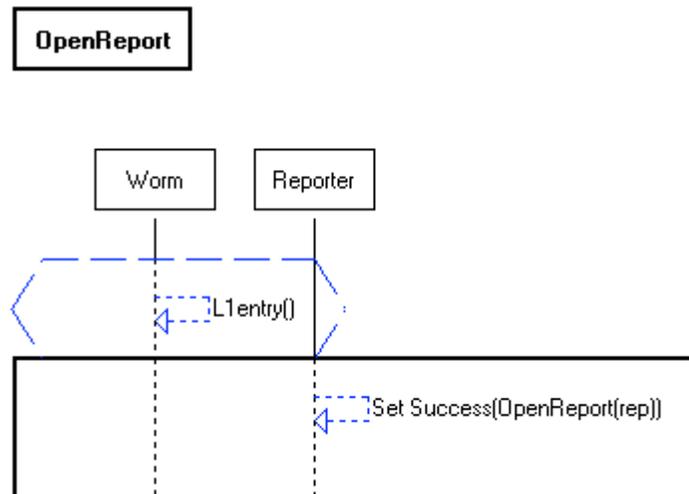




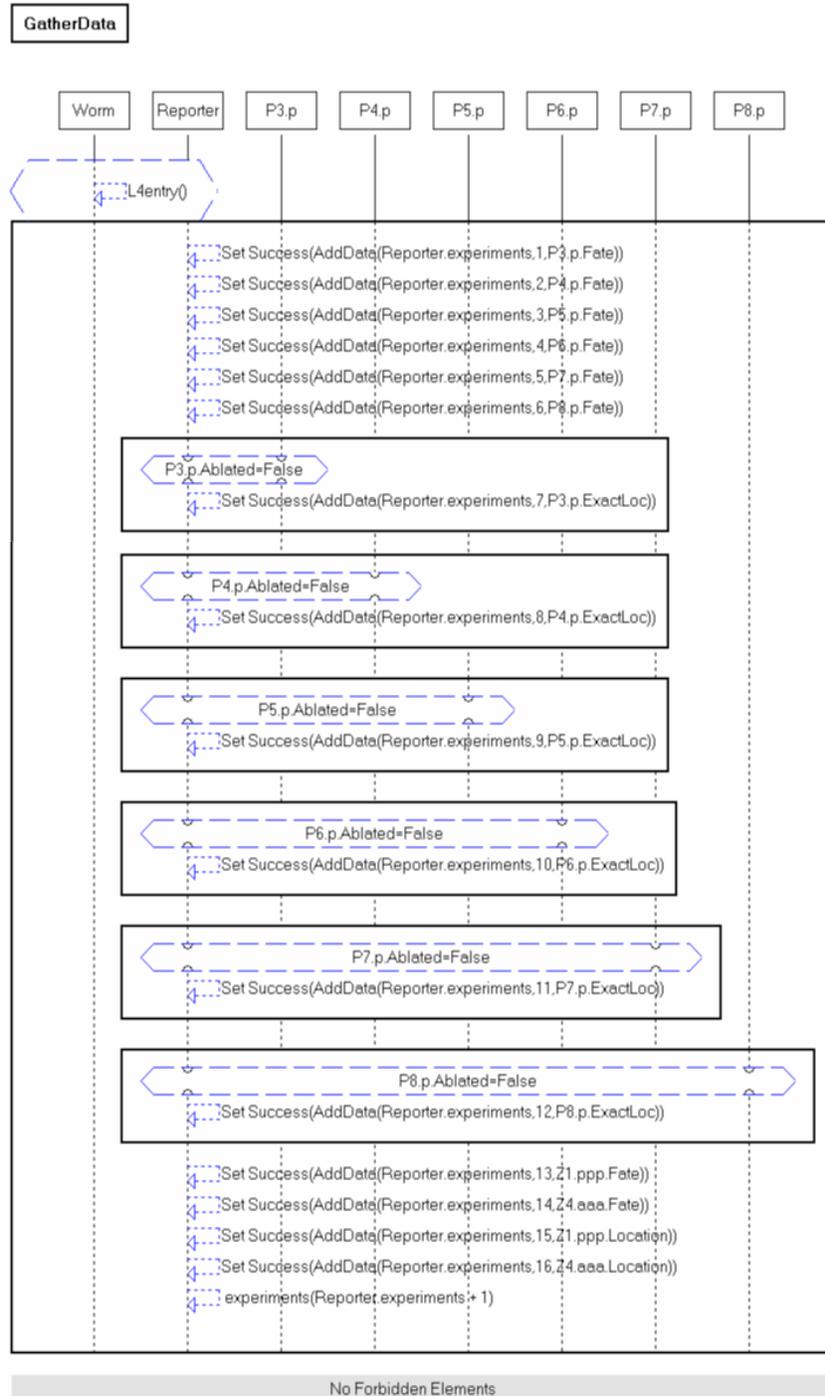
As explained in use case Core Behaviors, time progress in our model is set in units of hours during all development except for the first hour in L3 that is measured in units of minutes. To visualize this in the developmental time control, we add a Simulation object with property TimeScale that can be assigned the values Minutes or Hours. LSC "Hours1" sets the Simulation scale to Hours after hatching, after L3entry it is set to Minutes in LSC "Minutes", and back to Hours after S phase as specified in LSC "Hours2". S phase occurs one hour after L3entry as specified in "PnpAssumeFate (VPC Fate Assumption)". In LSC "TimeSync" the time progress of the DevTime control is set according to the scale, with increments of 1 for hours and of 0.01 for minutes (these values were selected for the graphical representation and their only effect is on displaying the time progress in the DevTime control).

## Reporter

This use case is used for storing important information on a run in an Excel sheet during batch mode execution.



For the purpose of manipulating the Excel file, we introduce a special object, the Reporter. In LSC “OpenReport” a report is opened after L2entry by calling the `OpenReport(rep)` function, that opens an Excel file whose name appears in parameter `rep`. The reporter sets the property `Success` to the value `true` or `false` according to the status returned by the function.



The LSC “GatherData” records information to the Excel sheet, it is activated by the execution of L4entry appearing in the prechart, and stores the fates of the six VPCs in columns 1- 6 and their Locations in columns 7 – 12, the Location information is especially useful for experiments with ablations in which the VPCs move and end up in different locations than their original wild-type locations. Additionally the fates and Locations of the AC/VU cells Z1.ppp and Z4.aaa are recorded in columns 13,14 and 15,16 respectively. Finally the experiments property of the Recorder is incremented by 1, this property is used to determine the row in the excel sheet

where the information is recorded, thus the Reporter is ready to write the results of the next run in a new row.

## References

- Ambros, V., Cell cycle-dependent sequencing of cell fate decisions in *Caenorhabditis elegans* vulva precursor cells. *Development* 122(8), 2507-15.
- Aroian, R. V., Koga, M., Mendel, J. E., Ohshima, Y., and Sternberg, P. W. (1990). The let-23 gene necessary for *Caenorhabditis elegans* vulval induction encodes a tyrosine kinase of the EGF receptor subfamily. *Nature* 348, 693-9.
- Chen, Z., and Han, M. (2001). *C. elegans* Rb, NuRD, and Ras regulate lin-39-mediated cell fusion during vulval fate specification. *Curr Biol* 11, 1874-9.
- Clandinin, T. R., Katz, W. S., and Sternberg, P. W. (1997). *Caenorhabditis elegans* HOM-C genes regulate the response of vulval precursor cells to inductive signal. *Dev Biol* 182, 150-61.
- Greenwald, I., Sternberg, P., and Horvitz, H. (1983). The lin-12 locus specifies cell fates in *Caenorhabditis elegans*. *Cell* 34, 435-44.
- Hill, R.J. and P.W. Sternberg. (1992). The gene lin-3 encodes an inductive signal for vulval development in *C. elegans*. *Nature*. 358, 470-6.
- Kaech, S. M., Whitfield, C. W., and Kim, S. K. (1998). The LIN-2/LIN-7/LIN-10 complex mediates basolateral membrane localization of the *C. elegans* EGF receptor LET-23 in vulval epithelial cells. *Cell* 94, 761-71.
- Kimble, J. and D. Hirsh. (1979). The postembryonic cell lineages of the hermaphrodite and male gonads in *Caenorhabditis elegans*. *Dev Biol* 70(2), 396-417.
- Kimble, J. (1981). Alterations in cell lineage following laser ablation of cells in the somatic gonad of *Caenorhabditis elegans*. *Dev Biol* 87, 286-300.
- Simske, J. S., and Kim, S. K. (1995). Sequential signalling during *Caenorhabditis elegans* vulval induction. *Nature* 375, 142-6.
- Sternberg, P. W. (1988). Lateral inhibition during vulval induction in *Caenorhabditis elegans*. *Nature* 335, 551-4.
- Sternberg, P. W., and Horvitz, H. R. (1986). Pattern formation during vulval development in *C. elegans*. *Cell* 44, 761-72.
- Sternberg, P. W., and Horvitz, H. R. (1989). The combined action of two intercellular signaling pathways specifies three cell fates during vulval induction in *C. elegans*. *Cell* 58, 679-93.
- Sulston, J. E., and Horvitz, H. R. (1977). Post-embryonic cell lineages of the nematode, *Caenorhabditis elegans*. *Dev Biol* 56, 110-56.
- Sulston, J. E. and White, J. G. (1980). Regulation and cell autonomy during postembryonic development of *Caenorhabditis elegans*. *Dev Biol* 78, 577-97.