Translating Statecharts to behaviourally equivalent Petri Nets (discrete-time semantics)

Mouzzam Hussain[1]

[a]*mouzzam.hussain@student.uantwerpen.be*

Abstract

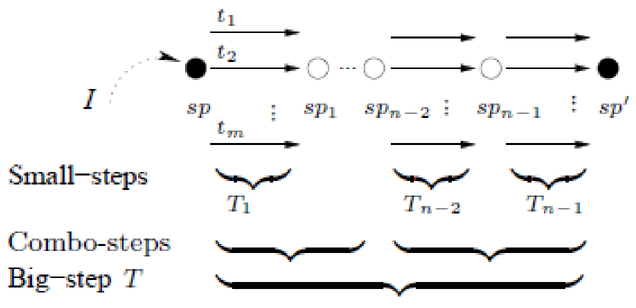
Model driven development is getting popular among researchers resulting in an existence of abundance of modeling languages. The decisions of choosing a suitable language for the model under study rests with the requirements engineer. **?** have addressed this problem from a semantic point of view of Big Step Modeling Languages (BSMLs) which comes in the popular class of behavioral modeling languages such that model can respond to environmental input by executing multiple or concurrent transitions.In this article semantics of BSMLs are deconstructed into a set of orthogonal semantics with advantages and disadvantages making it easier for the modeler to take a informed decision by taking in account the advantages and disadvantages of each option.

Keywords: Big Step Modeling Languages, Model Driven Development MDD, State Charts, Petri Nets

1. Introduction

As Model Driven Development is getting popular there is an ever increasing need to improve how we create well-designed behavioral modeling language. There is a wealth of literature available on this topic. The existing languages are based on either Harels State charts, or those subscribed synchrony hypothesis. These languages are referred by **?** as family of big-step modelling languages (BSMLs). It is because they are used to model systems that respond to environmental by multiple transactions without missing any environmental input. There are two main problems that are being addressed

Email address: `mouzzam.hussain@student.uantwerpen.be` (Mouzzam Hussain) by ?that are, when it is suitable to choose a semantic variation and how can one compare different semantic variations, on basis of what criteria? These questions are addressed by deconstructing the semantic variations into a set of almost orthogonal semantic features described in section [3] and analyzing the advantages and disadvantages of the semantic option for each feature. In section 2 describes background, section 3 provides the semantics and in section 4 conclusion and future work is given.



The figure 1 shows a Big step T where sp is the snapshot, sp1, n-2 are the intermediate snapshots and sp‘ is the final snapshot. The dotted arrow represent an input is received by the environment and solid arrows represent transition executions. The set of transition executions T(1 i ¡ n) are small steps. T has two combo transitions in the example.

2. Background

A BSML can be defined as a modeling language whose execution semantics can be described as a sequence of big steps, each triggered by inputs from the environment. A big step can be described as a sequence of small steps. When a small step is executed the model moves from one snap shot to the other where snap shot is defined as a set containing the variables and their values along with the status of events. More over there is also a notion of combo steps which are a continuous segments of sequence of small steps of a big step such that the values considered for the computation are based on the values of the snap shot at the beginning of the combo step.

2.1. Syntax

Syntax: The syntax for BSMLs described by [? is based on composed hierarchical transition systems (CHTSs) syntax. A CHTS consists of a composition tree with control states as nodes and a set of transition between control states.A BSMLs has states and transitions, states have different types. The states and transitions are explained as follows:

A control state is represented as a rectangle with a label specifying the name. A transition is shown as an arrow among two control states. Control state can be of type: Or, Basic or Concurrent where basic control state has no children, An or state has a minimum of one children and concurrent state has at least two children graphically separated by a dotted line. A default state is the one in which system resides in at the start, it is graphically represented with an arrow with no source. Two states have ancestor relation if one is the parent or grandparent of other. Two states are orthogonal if dont have ancestors relation and their least common ancestor is a concurrent state. Two control states are ancestrally related if one is a (grand) child of another.

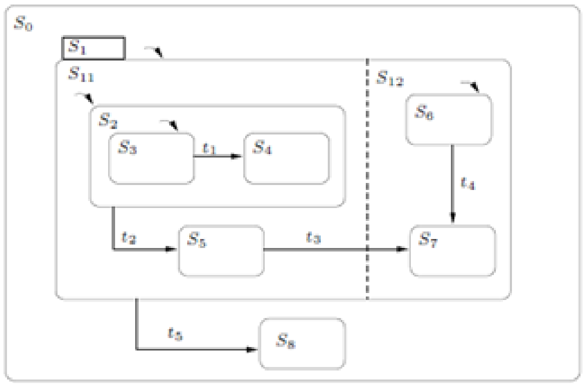


Figure 2: A model graphically represented as CHTS.

In the model in Figure 2, S8 is a Basic-state, and S1 is the only Concurrent-state of the model, whose two children, S11 and S12, are Or-states. The default control state of S11 is S2.Control states S3 and S7 are orthogonal. Or-states S11 and S12 are the two HTSs of the model.

The computation in a model is done by executing the transitions. A transition changes the value of variables and status of events resulting in the model moving from one control state from the other. A transition consists of the four components namely an event trigger, variable condition, variable action, an event action and an event set where event trigger is the conjunction and negation of events, variable condition is a logical expression defined over a set of variables, variable action is a set of assignments and event action is a set of generated events. The figure 3 shows a transition.

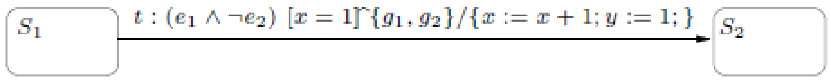

S_1
$t : (e_1 \wedge \neg e_2)\ [x = 1]\hat{}\{g_1, g_2\}/\{x := x + 1; y := 1; \}$
S_2

Figure 3: A Transition with its four components.

where (e1 e2) is event trigger, x = 1 is variable condition, x := x + 1, y = 1 is variable condition, and g1, g2 is event action.

2.2. Common Basic Semantics

The semantics describes how a model will react via a big step to a received input.

Big Step: The execution of big step is defined as a sequence of small steps. A transition can be executed if it is enabled and the following conditions hold: the event trigger and variable conditions are satisfied and the model resides in a set of control states such that source of transition either belong to the set or is a parent or grandparent of one of the states in the set.

Small Step: A small step is a set of transition occurrences such that when it is executed the model moves from one snap shot to the other.

combo step: A combo step is the one in which there are a continuous segments of sequence of small steps of a big step such that the values considered for the computation are based on the values of the snap shot at the beginning of the combo step.

Snap Shot: A snap shot is defined as a set containing the variables and their values along with the status of events. Snapshots consists of three element namely: S a control state in which the model resides in, V a set of pair containing variable name and its value in the snapshot and E a set of events either generated or received from the environment.

The small step has atomic execution which means that two transitions in a small step, variable and event actions of a transition cannot be seen by the other, all the variable, event actions of the small step take effect and during the transition the model is either in source or destination snap shot not in-between.

2.3. Taxonomy of BSML:

A semantic reasoning is needed for why a model should execute multiple small steps in response to an environmental input in order for semantics of BSML to be workable. There are three justification proposed by **?**.

Fast computation: This is an assumption that the system considered for BSML is fast enough so that it doesnt miss any environmental inputs while processing a previous input. This comes under the design family of BSMLs supporting the synchrony hypothesis.

Helpful environment: This is an assumption that the system considered for modeling has an environment that is helpful i.e. it does not overwhelm the system with inputs when it is busy.

Asynchronous communication: This is an assumption in which system has a buffer such that it can store environmental inputs that might arrive while the system is executing a big step. Thus a model have an asynchronous communication with environment ensuring no input is missed.

3. Semantic Aspects:

In **?** semantics of BSMLs are divided into eight semantic aspects: concurrency, transition consistency, maximality, memory protocols, external variable communication, event lifelines, external event communication, and priority. BSMLs vary in consistency in two aspects that are can more than one transition occurrence be taken together or is there an order of execution between small steps.

3.1. Concurrency

3.1.1. Number of Transitions:

There are two ways to model the execution of a system:

Single transition: allows only one transition to be executed The advantage of single option is that model is understandable as the model only considers one transition at a time. The disadvantage is the absent of priority can lead to many non-deterministic execution that are needed to be considered by molder.

Many-transition: in which a maximal set of transitions can be taken in a small step. The advantage of this option is that undesired non determinism of single option can be avoided. The disadvantage is the modeler should consider a more complex system.

Synchronization: combines the Single and Many options. All the involved transitions are executed together. The advantage of this option is when there is no communication among event single concurrency semantic is chosen and if there is an event communication it is facilitated with in small step. A disadvantage is that modeler should be aware that the execution of a transition can lead to synchronization which can be optional or mandatory.

A possible semantic variation point for the synchronization semantics, regardless of the choice of the Mandatory or Optional semantic option, is to consider a priority between taking a small-step that involves a synchronization over a small-step that does not, or vice versa

3.1.2. Order of transition:

This section explains over the semantics of what order of transitions to choose to perform execution. The order of transitions are as follows:

None: There is no order of execution of the transition occurrence in big step. An advantage is that modeler doesn't have to specify the order. The disadvantage is the modeler has too consider non determinism.

Data Flow: The order of transition specified in big step by considering the data flow order among variables. The advantage of this semantic option is an order in the model as well as the definition of data flow allows a variable to be assigned a value exactly once. The disadvantage is that it is less reactive in set of transitions that can be taken in big step.

Explicit: A syntactical order that explicitly specifies in which order the transitions can be taken. The advantage of this option is giving control on how transitions should be executed. The disadvantage is that modeler should consider specification of order of execution.

3.2. Transition Consistency:

This section explains the semantic options that deal with the consistency of the transitions.

3.2.1. Small Step Consistency:

If the control states are arranged in hierarchy, options regarding semantics of transitions exiting control and hierarchical states should be considered. This give rise to the following semantic options:

Arena Orthogonal: In this option, two transition occurrences are included in the same small-step only if their arenas are orthogonal where the arena of a transition is the smallest Or-state that is the (grand) parent of the source and destination control states of the transition. The advantage of this semantic approach is its simplicity where it can also be a disadvantage as non-determinism is introduced to which transition should be taken.

Source/Destination Orthogonal: in order for a small step to be Source/Destination Orthogonal there should be two transitions t and t‘such that their sources and destinations are orthogonal and the destination of t and source of t‘ are orthogonal and vice versa. The advantage is that it avoids non determinism from arena orthogonal semantic. The disadvantage of this is determining destination of small snapshot is not taken.

3.2.2. Interrupt Transitions and Preemption:

The notion of preemption is taken into account when two transitions occur whose sources are orthogonal and either the destination of other transition is not orthogonal with the source and their destinations are ancestry related.

Preemptive and Non-Preemptive: This semantics specifies whether two transitions can be taken together in a small step or not when one is an interrupt to other. If this semantic doesn't allow The advantage of this option is last wish of a concurrent state can be satisfied. A disadvantage of this option is specifying destination of a small step is complicated.

3.3. Maximality

The maximality condition specifies when a sequence of small steps of a big step ends.

Syntactic: When a model or a part of a model ends execution the big step becomes stable as a certain configuration is reached. If concurrency is used than all of the concurrent components should become stable in order for the big step to become stable.

The advantage of syntactic option is that a modeler can identify the static scope. While a disadvantage could be if there exist an execution such that model is never ready to environmental inputs that may lead to non-terminating small steps. Another disadvantage is the possibility of a big step optionally continuing or ending from a control state without considering environmental input.

Implicit: If we dont use syntactic constructs there are two implicit ways of defining maximality namely Take One and Take Many.

Take One: This semantic option assumes a big step to be maximal either if there are no transition to be taken or there is no transition enabled such that its source belongs to a configuration at the beginning of small step.

The advantage of this approach is that modeler can easily follow execution of a big step to next configuration if the source is identified. A disadvantage cant easily keep track of the scope of big steps.

Take Many: This semantic option simply continues the execution of the sequence of small steps until there is non left.

3.3.1. Maximality of Combo-Steps

This semantic specifies that the where the computation is carried out in a segment of big step the values read for variables and events should not be changed during the execution of this big step i.e. from the snap shot at the beginning of this big step.

3.3.2. Non-reactiveness and Environmental Input Assumptions

If a model subscribes to syntactic maximality and it cant reach its stable state as there is no small step to be executed or the model is in a configuration that even with the input from the environment it cannot execute a big step.

3.4. Memory Protocols

In a BSML variables are persistent artifacts or the medium to carry computations. The memory protocol deals with the semantic aspect of how to obtain the values of a variable when it is accessed.

An external variable in a model is a variable that is assigned values by the environment of the model.

Big-Step: When using this memory protocol the values of variables and events returned are always values at the beginning of the big step. An advantage of this semantic is that the enabledness of transitions does not change during the execution of big step.

Small-Step: Similar to the big step memory protocol this semantic also considers the values of variables and events that are computed by the previous small step. An advantage of using this option is a sequence of computations can be described as the sequence of transitions. The disadvantage however is that a transition can enable another transition So the reviewer of a model should check enabledness of all transitions.

Combo-Step: In this memory protocol the values returned for variable and events are from the snapshot at the beginning of the combo step. The advantage is similar to the previous memory protocols as the variable condition does not change during the execution of this combo step. The disadvantage is that it is difficult to determine the combo step by review and to determine variables values used in small step of the combo step.

3.4.1. Syntactic Keywords:

There are four syntactic keywords described by **?** that are explained as follows:

Pre: returns the value of a variable at the beginning of a big-step. **Cur:** returns the value of a variable at the beginning of the current small-step. **Change:** is a Boolean variable whose value is true if the value of a variable is changed during this big step and false otherwise. **New:** returns a value for a variable only if the value of variable is changed during the execution of the big-step. A disadvantage of new keyword is that two transitions depending cyclically on each other might be blocked because their assignment depend on each other via new statement.

New small: It returns the value at the end of current small step but the difference between new and new small is that new small only returns the value if a value has been assigned in the current small step compared to new which returns the most recent value if it has been assigned a value in the current big step. **New big:** It returns the value variable at the end of big step. New big returns the last assignment to a variable compare to new which returns the value depending upon which sequence of small step new has been used.

3.4.2. Race Resolution

It is a condition when more than one transition in a small step assigns a value to a variable. **?** mentioned such small steps as racy small steps and the variables to which the values are written are mentioned as racy small steps. The disadvantage of allowing race condition is that one cannot reason about the assignments of a transition independent to the other.

There are two race resolution mentioned namely Random and Linearization.

Linearization: In this semantic option the values of the variables are according to an arbitrary sequence if executions of the small step at the end of a racy small step when the transition doesnt consider assignments of its preceding transitions. There are two justifications why the number of outcomes could be less than a factorial of transitions of a small step. If a variable is assigned a value by transition in linearization than the other set of assignments can be ignored and if in linearization two consecutive transitions assign values to disjoint variables.

There are two reasons why there could be fewer distinct outcomes than the factorial of the number of the transitions of a small-step: The advantage of this semantic is that it is easier to analyze result of race resolution in a racy small step. The disadvantage is that it doesnt work incrementally in associative way.

Random: In this semantic option all assignments belonging to a transition in racy small step are sequentialized. The advantage is that all the combinations of transitions are considered in an incremental, associative way. The disadvantage is that it is non intuitive and non-predictable.

3.5. External Variable Communication

An External variable is a set containing the environmental input and output variables.

3.5.1. Inter-Component Communication

In BSMLs a model can be structured as a composition of a set of components where these component can behave as the environment or a part of it. These components are meant to represent different parts of the model and they communicate through an Inter Component Communication module. The variables involved in such a modules are described as interface variables in [shahram].

Synchronous: In this type of Inter Component Communication once a values is assigned to an interface variable by a sending component, this value is available for the receiving component in the big step. There are two sub types in this semantic.

Strong Synchronous: The sending component assigns a value to interface variable which cannot be accessed by the receiving component.

Weak Synchronous: In this option the strong synchronous communication is relaxed as the read to the interface variable is allowed for the receiving component even if the variable will be assigned a value in the big step.

The advantage of both the options is that zero time computation principle of the synchrony hypothesis is followed such that the value of the interface variable is exchanged in zero time. However there is an extra advantage as this option treats the environment input variables and interface variables with similarity. The disadvantage of the strong synchronous option is the read from interface variable is blocked as it is not clear whether the interface variable has the same or new value.

Asynchronous: This semantic states that the assignment to a variable is visible in the next big step. The advantage of this semantic option is that a transition never blocks the read of the interface variable. Also the interface events are treated in a modular way ensuring their availability in the beginning of the big step. The disadvantage is that it is not compatible with synchrony hypothesis as the communication between components takes one big step to complete.

3.6. Internal Events:

The internal has a status and can be either present or absent, the event triggers of the models can sense these events. The internal event is a transient medium for communication as opposed to variables.

3.6.1. Life Line

This semantic specify the intermediate snapshots of the big step. There are different types of life lines presented by **?** namely whole, remainder, next combo step, next small step and same.

Whole: In this type of life line semantic the event generated is considered present in the big step regardless of which small step it is generated in and it persists till the end of big step. The advantage of this semantic option is it is modular [32] with respect to events this means that the event generated by the model should be the same as if received from the environment. The disadvantage of this option is that non-causal big steps are allowed.

An advantage of this semantic option is that it is modular [32] with respect to events. For a semantics to be modular, an event generated by the model should be treated the same as if it was received from the environment.

Remainder: In this type of life line semantic the generated event is considered to be present in intermediate snapshot after being generated in in small step and it persists during the big steps. The advantage is that there is a causality relationship between transitions. The disadvantage of this semantic option is that effects of generated events in the big step are not sequenced and it doesnt also support synchrony hypothesis.

Next Combo-Step: This semantic option states that the event generated is present only during combo step and after this combo step. The advantage is that this semantic option provides a more rigorous causal ordering [38] between the triggered and generated events in big step. A disadvantage is that events have multiple instances making it incompatible with synchrony hypothesis where the event should either be present or absent.

Next Small-step: In this semantic option the event generated is only present in the next small step. An advantage of this semantic option is that order between generated and triggered event of the big steps is causal.

Same: In this option for lifeline semantics the event generated is present in the current small step. The advantage of this semantic is that algebraic properties can be derived. The disadvantage is that non causal big steps are possible.

3.6.2. Negation of Events and Global Inconsistency

The absence of the internal event to trigger transition negation operator can be applied to trigger the event. The negation of event is denoted as e by **?** means that a transition is enabled only if an event e is missing. To use the absence of an internal event to trigger a transition, a negation operator can be applied to the event. For an event e, its negation, denoted as e, means the transition is enabled, only if e is absent.

Global Consistency: If a semantic subscribed to the remainder semantic option can avoid the problem of global inconsistency then it is globally consistent semantics. Global consistency can be alternatively described as big step, combo step and future consistency described as follows:

Per Big-Step Consistency: An event generated in a combo step of a big step shouldnt be considered as absent in the triggering condition of current or previous combo steps.

Per Combo-Step Consistency: if an event is absent in the combo step than it should not be generated in the same combo step.

Future Consistency: An event generated in combo step than it should not be considered as absent in current or any future combo steps.

3.7. External Event Communication

BSMLs distinguish syntactically among events that are used by model for communication internal and external. Most often whole semantics are used for environmental input events and remainder semantics are used for environmental output events. This means that environmental input once received in the beginning of big step persists throughout the remainder of big step.

3.7.1. Inter-component Event Communication

An inter-component event communication mechanism uses the interface events of a model to provide the means for communication between the components.

Synchronous Event: In this inter-component communication mechanism, once a sending component generates an interface event during a big-step, the generated event becomes available for the receiving components within the same big-step.

Strong Synchronous Event: In this option, an interface event in a receiving component is either present throughout a big-step from the beginning, or is absent throughout the big-step. This semantics is the Whole lifeline semantics for interface events.

Weak Synchronous Event: In this semantic option, an interface event need not be present from the beginning of a big-step. However, if an interface event is generated by a sending component, the receiving components would receive the event during the current big-step. This semantics is the Remainder lifeline semantics for interface events.

Asynchronous Event: In this semantic option, a generated interface event will be sensed in the big-step after the big-step in which it is generated. The advantage of this option is that the semantics of interface events can be described in non-forward referencing way. The disadvantage is that it doesnt comply with synchrony hypothesis.

3.7.2. Non-distinguishing BSMLs

In non-distinguishing BSML, there is a notion of virtual input that is an event received from the environment at the beginning of a big step.

Assume Input as Environmental: In this semantic option a virtual input is treated as an environmental input and it will persist throughout a big-step. The advantage of this option is environmental input persists throughout the big step. The disadvantage is that it Is difficult to distinguish between different roles of event in a big step.

Assume Input as Internal: In this semantic option virtual input is considered the same as other. The advantage is that it semantics simplicity exists treating all events similarly. The disadvantage is that it doesnt comply with BSMLs notation and it is also difficult to distinguish between different roles of a system with in a big step.

Hybrid: This semantic option has the assumption to take input as internal event but treats as a genuine input event of model. The advantage of this option is once a genuine input is identified it is treated as environment input. The disadvantage is the same as general non distinguishing BSMLs.

3.8. Priority

There could be multiple transitions available that can be chosen to be executed as a small step and are described as set of potential small-steps in [shahram09]. The priority semantics specify which set of transitions should be chosen from the potential small-steps of a snapshot.

Hierarchy: In this semantics some of BSMLs use the hierarchal structure as a way to assign priorities to ancestrally related control states. On the basis Hierarchy semantics elements of a transition can take: Source, Destination, or Arena options.

Arena-Child is a Hierarchy priority semantic giving high priority to the transition with lowest arena. Where the arena of the transition is the smallest Or state including both source and destination states of a transition. The advantage of using Hierarchy priority semantics is that easily understandable or represented graphically. The disadvantage is that in some models it may not be possible to provide prioritization in sets of potential small steps.

Explicit: This option uses explicit ways such as assigning numbers to transition to describe their priorities. The advantage of this option is that the modeler has control over specifying the priorities. The disadvantage is that it can be annoying to assign priorities manually to each transition.

Negation of Triggers: The negation semantics uses the negation of events and conditions in the event trigger condition to assign priorities among transitions. The advantage of this semantic is no extra syntax or semantics need to be considered while a disadvantage is the modeler should know all the transitions that should have a lower priority.

4. Conclusion and Future Work

In this report semantics of BSMLs have been deconstructed in orthogonal semantic aspects along with their advantages and disadvantages. The main focous of this report is to make the decision of choosing a suitable modeling language easier and the modeler can also be informed about the pros and cons of his choice. The main objective of this report was to understand the details and semantics of BSMLs. In the future a subset of these semantics would be mapped on petri nets to do perform analysis on it.