Suggestions to Solve the gap for Flows and Behavior for SysML 1.7

One of the problems that has been partially solved by SysML was the addition of the concept of flows (through Flow Properties and Item Flows). But even though SysML added these concepts they have never been fully completed. For example, the standard is interested in InterfaceBlocks having their Operations bound to either methods or AcceptEventActions but there is no such implement requirements for flows.

There are three kinds of flows that a modeler wants to represent, namely: Blocks, Signals, and Invocations. Currently, on an ItemFlow one can represent Blocks and Signals but not Invocations. In Behaviors, one can easily represent Signals and Invocations but not Blocks. This has lead to “work-arounds” such as representing everything as Signals to get around these issues.

To solve these issues, the following steps are suggested:

1. Add a stereotype <<ObjectEvent>> on AnyEvent, with a Tagged Value of classifier:Class[0..*] this allows StateMachines and Activities to fully take advantage of an <<ObjectEvent>>.
2. Extend the stereotypes <<overwrite>>, <<noBuffer>>, <<rate>>, <<probability>>, <<continuous>>, <<discrete>>, ordering, upperBound, ControlType, isControlType should all be able to be set on Ports as well as Pins, there should also be OCL constraints to constrain these when Pins and Ports are “connected”.
3. Extend the stereotypes <<rate>>, <<probability>>, <<continuous>>, <<discrete>> on an Association or a connector.
4. Fix the fact that one cannot make a distinction for Interactions and StateMachines between Signals and Calls. These can have the same name and are still distinguishable due to the fact that one is a Classifier and the other a Signal. Perhaps, for a Signal make it C(()). One would also need to make <<ObjectEvent>> somehow distinguishable.
5. One should also be able to use ChangeEvents across objects, such that a ChangeEvent on block A can be sent on to block B. These should be able to be sent on an ItemFlow (but ChangeEvents are not a Classifier).
6. StateMachines need to be able to transition on receipt of an Object. Currently, there is no way to do this. This can be done if there is an <<ObjectEvent>>. One needs a way to designate an <<ObjectEvent>> in all the behavioral forms: Interaction, Activity, and StateMachines. Currently, Interactions can be done in Interactions with a Message without () (nothing needs to be added). Activities can be done through using a SendObjectAction (<<continuous>> can be supported currently) and an AcceptEventAction using an <<ObjectEvent>> (<<continuous>> being supported currently), so Activity is covered. But StateMachines need a way to show that a receipt of an object can trigger a transition. Probably, there would also need to be some sort of way to count <<ObjectEvent>>s of a particular type (e.g. transition on an <<ObjectEvent>> only after 4 Objects of a particular type have been received in a guard). Count is for <<discrete>> objects, there is some analogous form for “amount” for <<continuous>> objects. <<Rate>>s should also be able to be used in guards.
7. Invocations need to be able to be used in ItemFlows. Perhaps, C(4):5 type of thing could be used (either arguments or parameters given) from Interaction forms. Question is whether Synchronous is a two-way flow or considered a one-way flow (I would vote for one way).

8. Also, so all of this could be done on BDDs (and not have to involve an IBD), let’s allow associations to be between Ports on BDDs with the interpretation being that an Association between Ports has the semantics of the Association being between the types of the Ports specified.

9. There should be some way to be able to connect flows from various ports. Let’s consider the following simple problem. How does one specify that g1 connects to hb. Perhaps this kind of connectivity belongs to the Association. This needs to be fixed. Probably needs to be able to connect g1 to both ha and hb. Note that these connections can be <<conserved>> (perhaps we need to add this in from SysPhS) or <<nonconserved>> could be in the standard or in SysPhS.

10. Standardize the connection between a structural ItemFlow and FlowProperty and get it into a behavior. The issue is that now in addition to a port that can be specified, one could specify a flow. For an <<ObjectEvent>> on a Port, that <<ObjectEvent>> could come from various flows on a Single Port. So in Interactions, Flows would need to be available as a PartDecomposition on the PartDecomposition of the Port. Actions can have “via <port>” and “<<from>>(<port>)”. Now in addition, one could define a flow as well, so it might become “via <port>&<flow>”. This could be done by restricting a port to having a single flow.

Thanks,

G. Edward Roberts

http://www.elparazim.com  eroberts@elparazim.com  +1 817-688-5321  skype: g.edward.roberts