# Semantics of UML Composite Structures meeting (Berlin, 20/06/2013)

## Participants

Yves, Conrad, Alan, Sandy, Charles, Ed, Arnaud

## Summary

* Discussions were focused on the semantics of SysML.
* Some issues to be addressed by the SysML RTF have been identified (semantic variation points for Push/Pull, Depletive/Non-depletive semantics, of read/write on flow properties).
* The normative part of the specification has to deal with untyped connectors.
* Discussions on the mechanisms available to extend the normative execution model (new semantic visitors, new execution factory, etc.).
* Discussions on instantiation of links.
* Initial test cases have been constructed starting with the simplest case of two parts that are connected by an untyped connector. The first test case includes no ports, and the second case includes a proxy port that is behavioral. The parts and the type of the proxy port include a flow property that is typed by a value type and a block.There is is agreement on the semantics for these cases.
* The initial test cases do not include BindingConnectors.
* Examples from Nicolas have not been reviewed, and correspond to “more advanced” test cases than the one we have been working on during the meeting. The review of SysML semantic clauses has not been reviewed as well.

## Test Cases and Permuations

* For this discussion, we focused on the simplest test cases noted above. There are many permuations to consider including:
	+ Port
		- With/Without
		- Kind – Proxy, Full, Unspecified
		- Behavior - True or Faluse
		- Typed - True or False
		- Nested - True or False
		- Port multiplicity
	+ Flow property
		- Direction - in, out, in-out
		- Type is value type, block, or signal
		- Multiplicity
	+ Non-flow property
		- Type is value type, block
		- Multiplicity
	+ Operation
		- With/Without
		- Direction
	+ Connector
		- Nested connector end - True or false
		- Binding – True or False
		- Typed – True or False
		- Connector end multiplicity
	+ Interaction with Classifier Behavior
		- Activity with parameters

## Semantics of FlowProperties:

* Discussion on Push/Pull semantics:
	+ Currently in SysML, Push semantics (cf. sub-clause 9.3.2.7)
		- Write to flow property with direction out, propagates value to matching flow property at opposite end of connector
	+ Pull semantics could be useful as well.
		- Read value of flow property with direction out at opposite end of connnector
	+ **SysML should introduce a semantic variation point on this topic, and/or some specific notations/abstract syntax.**
* Discussion on Depletive/non-depletive semantics of ReadStructuralFeatureActions on FlowProperties
	+ Semantics in UML is non-depletive (for “regular” properties)
	+ SysML does not say anything about this, but it could be useful.
	+ But, it sounds like putting too much behavior description in the structure (i.e., flow properties would behave like object nodes in an activity).
	+ **SysML may introduce a semantic variation point with strategies for dealing with this, and/or specific notations/abstract syntax**.
		- **For example, a strategy may impose one semantic if the classifier behavior associated with the sending part is an activity**.
* Rules for matching FlowProperties:
	+ Matching of FlowProperties relies on direction, type and name.
	+ Something could be said about multiplicities as well.

## Dealing with untyped connectors in the normative part

* Currently, the normative part of the proposed submission requires that Connectors are typed by associations
* This enables one to represent Connector instances by Links
* This also enables to instantiate “manually” connectors using LinkActions
* UML 2.5 now enables Links to be untyped. fUML/Composite Structures should support this as well
* However, note that UML does not say how to create untyped links “manually” (LinkActions rely on LinkEndData, which themselves rely on Associations)
* An option could be that, in case of untyped connectors, there is no instantiation of links
	+ Semantics of propagation has to deal with connectors existing between roles played by instances.
	+ Multiplicities on connector ends may cause problems.
* **Solutions have to be proposed (Arnaud)**

## Mechanisms for extending the execution model (“hooks”)

* Definition of new semantic visitors, extending fUML visitors and/or composite structure visitors
* The examples covered so far would require new visitors for ReadStructuralFeatureAction and AddStructuralFeatureValueAction
* These new visitors would typically account for the application of stereotype FlowProperty of read/written features.
* Definition of a new ExecutionFactory, extending the Composite Structures ExecutionFactory, for instantiation of the new visitors.
* If the SysML RTF introduces semantic variation points for Push/Pull, Depletive/Non-Depletive semantics, there should be a new SemanticStrategy to deal with this (and it should be used by the new visitors)
* But, a reasonable option is to formalize semantics as it is today in SysML 1.3 (which means, no new SemanticStrategy)
* The actual extension has to be implemented, but too early to start (=> The test cases have to be completed, and we nehed a global agreement on what their semantics should be).

## Basic test cases

* The first test cases to consider shall cover variants of:
	+ Multiplicity on parts, on ports, on connector ends, and on flow properties
	+ Connectors typed / untyped
	+ flow properties typed by ValueType/Block/Signal
* Category 1: Direct assembly of parts
	+ Cases covered at the meeting
		- Multiplicity 1 on parts, connector ends and flow properties
		- Connectors are not typed
		- Flow properties are typed by ValueType, or Blocks
	+ (Reproducing one of the variant here for people who could not attend)





* Category 2: Assembly of parts, through behavior proxy ports
	+ Cases covered at the meeting
		- Multiplicity 1 on parts, ports, connector ends and flow properties.
		- Connectors are not typed
		- Flow properties are typed by ValueType, or Blocks
* **Test case models done in the meeting to be included (by Arnaud) in the SysML Test Suite initiated by Nicolas**
* **These basic test cases have to be completed. The first test cases to be added concern variants of the two categories above, where flow properties are typed by signals (Nicolas?)**

## Instantiation semantics

* In the normative part, there is a semantic strategy for instantiation of composite structures
* The registered strategy applies when invoking a <<Create>> Operation on an object, in the case where this Operation has no method.
* SysML users would typically not used this facility. They would rather specify their own construction behavior.
* This requires a mechanism to deal with untyped connectors which cannot be instantiated manually (cf. previous items)