Semantics Formalisation: Some Experience with the Theory Plug-in (Extended Abstract)

Thai Son Hoang[1][0000−0003−4095−0732], Colin Snook[1][0000−0002−0210−0983], Karla Vanessa Morris Wright[3][0000−0002−0146−3176], Laurent Voisin[2][0000−0002−2426−0101], and Michael Butler[1][0000−0003−4642−5373]

[1] ECS, University of Southampton, Southampton SO17 1BJ, United Kingdom
{t.s.hoang,cfs,m.j.butler}@soton.ac.uk
[2] Systerel, 1115 rue René Descartes, 13100 Aix-en-Provence, France
laurent.voisin@systerel.fr
[3] Sandia National Laboratories, 7011 East Avenue Livermore, California 94550, USA
knmorri@sandia.gov

In [3], we model the semantics of SCXML [2] using standard Event-B constructs, i.e., contexts and machines (Approach 1). The Event-B contexts capture the SCXML's syntactical elements while SCXML's semantical elements are formalised using Event-B machines. In this talk, we report on our experience formalising SCXML using the Theory Plug-in [1] (Approach 2), in particular in comparison to Approach 1.

Approach 1. Formalisation using Event-B contexts and machines. The formalisation using the contexts and machines is summarised in Figure 1. The main features of this formalisation are:

– The use of constants to define the syntactical elements of SCXML.
– The use of context extension to build the syntactic model gradually.
– The use of axioms to define the syntactic constraints.
– The use of variables and events to capture SCXML's semantical elements.
– The use of invariants to specify the constraints for the consistency of the semantics.
– The use of the composition mechanism to combine different parts of SCXML, namely untriggered statecharts and run-to-completion scheduling.

Approach 2. Formalisation using Theories Plug-in. The formalisation using theories can be seen in Figure 2 The main features of this formalisation are:

– The use of operators and datatypes to define the syntactical elements of SCXML.
– The use of theory inclusion to build the syntactic model gradually.
– The use of well-definedness (WD) operators to define the syntactic constraints.
– The use of operators and datatypes to capture SCXML's semantical elements.

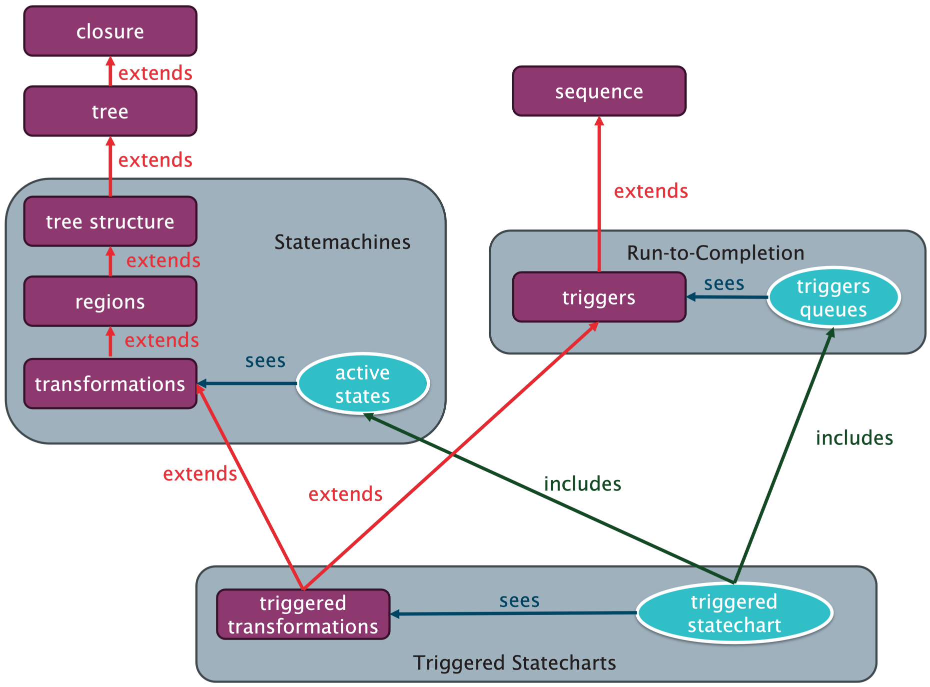

Fig. 1. Formalisation of triggered statecharts using Event-B contexts and machines

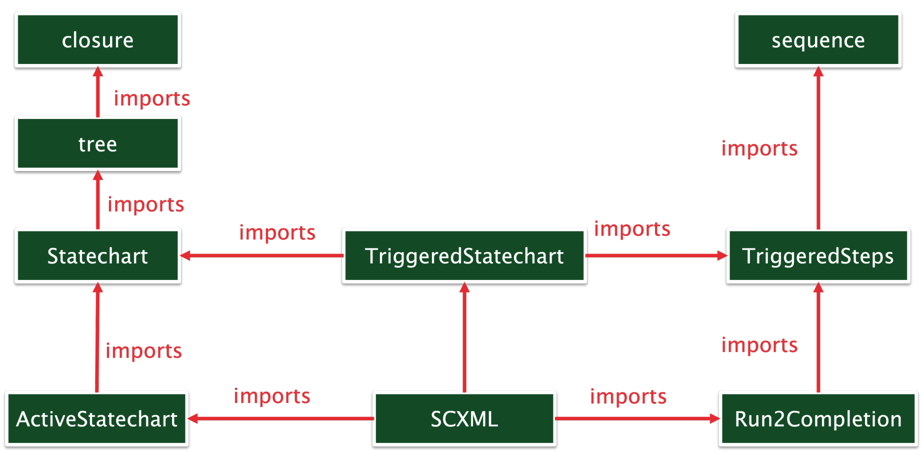

Fig. 2. Formalisation of SCXML statecharts using theories

Approach 1. Standard Event-B	Approach 2. Theory Plug-in
– Model a single SCXML statechart	+ Model a datatype of SCXML statecharts
= Syntactical elements are captured using contexts	= Syntactical elements are captured using theories
+ Syntactical elements are gradually added to the model using context extension	– Gradually introduce syntactical elements results in nested datatype
= Syntactic constraints are represented as context axioms	= Syntactic constraints are represented as WD operators
– Combination of different parts of the language using the composition plugin (i.e., outside of standard Event-B)	+ Composition is done by defining composite datatypes.
= Semantical consistency is encoded as machine invariants	= Semantical consistency is enconded as theory theorems
+ Consistency proof obligations are decomposed automatically (per individual invariants)	– Must manually construct theorems for decomposing the consistency proof
– No customisation for the provers to discharge proof obligations	+ Define proof rules for the provers to discharge proof obligations
– Model-related properties (e.g., refinement) requires additional tool	+ Model-related properties (e.g., refinement) can be stated as theory theorems

Table 1. Comparison between standard Event-B and Theory plug-in

– The use of theorems to specify the constraints for the consistency of the semantics.
– The use of theory inclusion to combine different parts of SCXML, namely untriggered statecharts and run-to-completion scheduling.

Comparison Summary. The comparison between Approach 1 and Approach 2 can be seen in Table 1.

References

1. Butler, M.J., Maamria, I.: Practical theory extension in Event-B. In: Liu, Z., Woodcock, J., Zhu, H. (eds.) Theories of Programming and Formal Methods - Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday. Lecture Notes in Computer Science, vol. 8051, pp. 67–81. Springer (2013), https://doi.org/10.1007/978-3-642-39698-4_5
2. W3C: SCXML specification website. http://www.w3.org/TR/scxml/ (September 2015)
3. Wright, K.V.M., Hoang, T.S., Snook, C.F., Butler, M.J.: Formal language semantics for triggered enable statecharts with a run-to-completion scheduling. In: Ábrahám, E., Dubslaff, C., Tarifa, S.L.T. (eds.) Theoretical Aspects of Computing - ICTAC 2023 - 20th International Colloquium, Lima, Peru, December 4-8, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14446, pp. 178–195. Springer (2023), https://doi.org/10.1007/978-3-031-47963-2_12