UML-B: A Plug-in for the Event-B Tool Set*

Colin Snook and Michael Butler

University of Southampton,
United Kingdom
{cfs,mjb}@ecs.soton.ac.uk

UML-B is a graphical formal modelling notation that relies on Event-B for its underlying semantics and is closely integrated with the 'Rodin', Event-B verification tools. UML-B is similar to UML but has its own meta-model. UML-B provides tool support, including drawing tools and a translator to generate Event-B models. When a UML-B drawing is saved the translator automatically generates the corresponding Event-B model. The Event-B verification tools (syntax checker and prover) then run automatically providing an immediate display of problems which are indicated on the relevant UML-B diagram. The UML-B modelling environment consists of a UML-B project containing a UML-B model. Four interlinked diagram types (package, context, class and statemachine) are available. Package Diagrams are used to describe the 'refines' and 'sees' relationships between top level components (machines and contexts) of a UML-B project. UML-B mirrors the Event-B approach where static data (sets and constants) are modelled in a separate package called a *'context'*. The context diagram is similar to a class diagram but has only constant data represented by *ClassTypes*, *Attributes* and *Associations*. ClassTypes define 'carrier' sets or constant subsets of other ClassTypes. ClassTypes may own immutable attributes and associations which represent constant functions. The behavioural parts (variables and events) are modelled in a Class diagram which is used to describe the *'machine'*. Classes represent subsets of the ClassTypes that were introduced in the context. The class' associations and attributes are similar to those in the context but represent variables instead of constants. Classes may own *events* that modify the variables. Event *parameters* can be added to an event, providing local variables to be used in the transition's guards and actions. Class events utilise a parameter, *self*, to non-deterministically select the affected instance of the class. State machines may be used to model behaviour. Transitions represent events with implicit behaviour associated with the change of state. Additional guards and actions can be attached to the transition. UML-B retains sufficient commonality with UML for the main goals of approachability to be attained by industrial users. Since UML-B automates the production of many lines of textual B, models are quicker to produce and hence exploration of a problem domain is more attractive. This assists novices in finding useful abstractions for their models. We have found that the efficiency of UML-B and its ability to divide and contextualise mathematical expressions assists novices who would otherwise be deterred from writing formal specifications. UML-B is also a useful visual aid for more experienced formal methods users.

* This work was carried out under the EU projects, Rodin [IST-511599] and ICT project Deploy [IP-214158].

E. Börger et al. (Eds.): ABZ 2008, LNCS 5238, p. 344, 2008.