Executable Modeling: Retrospective and Prospective

(Keynote)

Stephen Mellor
Email: StephenMellor@StephenMellor.com

***Index Terms*—Executable Models; Action Languages; UML**

I. Abstract

In the mid-Eighties, there were thirty object-oriented notations and thirty object-oriented methods. These fell into three main groups: sketches of software structure, depictions of software (usually with gaps to fill in the code) and what we now call executable models. By the late-Nineties, there was one notation and zero methods. Executable models were hard to find. The method wars were over. We (all) lost. Worse, models as graphical descriptions of code are "just documentation", overhead that gets in the way of the real thing: code. The Agile Manifesto of 2001 elevates "working software" over models. But "software" can also mean executable models.

The difference between a programming language and an action language is, as always, the level of abstraction. Programming languages are "aware" of data structures and threads of control. Action languages can abstract these away. Over more than a decade an action language has been added to UML and open-source implementations made available. Defined semantics are now on the horizon for state models. Perhaps UML can be made executable after all.

That will help, but it's not enough. We need our executable models to be composable so that when two components have been tested and the manner of their interactions tested too, we can have confidence that the combined component works. We need this now we are connecting industrial control systems to the Internet. Composable execution units are needed to bring executable modeling to growing industrial applications.

II. Biography

Stephen Mellor is the Chief Technical Officer for the Industrial Internet Consortium, where he directs the standards requirements and technology and security priorities for the Industrial Internet. In that role, he coordinates the activities of the several engineering, architecture, security and testbed working groups and teams.

He is a well-known technology consultant on methods for the construction of real-time and embedded systems, a signatory to the Agile Manifesto, and adjunct professor at the Australian National University in Canberra, ACT, Australia. Stephen is the author of Structured Development for Real-Time Systems, Object Lifecycles, Executable UML, and MDA Distilled.