A Language for Multi-Perspective Design and Runtime Monitoring

Blair Archibald

Models allow us to design, reason about, perform verification, and even generate implementations of system. We have shown that modelling from different perspectives, or "frames of reference" [1,2 3] can be more effective than trying to encapsulate everything in one model.

For example, in wireless sensor networks [2] we develop: 1) a physical perspective that models devices, location, and physical links, 2) a configuration perspective that models sensor information (ip addresses, available sensors, etc) and data gathering, and 3) an application perspective that models the possible applications running on the sensors. While each perspective is useful in its own right, considering interactions between perspectives allows us to verify properties such as: "there two temperature sensors in the park each with >90% battery".

The question we address is how to approach, formally, multi-perspective modelling and verification. We propose the underlying formalism should be Milner's bigraphical reactive systems (BRSs) that model agents in both space, with hierarchical nesting and hyper-links, and time, by specifying a transition system. Bigraph verification, can be used both at design time and for run time system monitoring. Each modelling perspective is described as a separate bigraph, which are then placed "side-by-side" using the bigraphical product. Links between perspectives can be used to describe multi-perspective constraints. For example, we can write a transition rule that only applies if a sensor is in a particular location and it has the capability to move into a new location.

Bigraphs have both a graphical and algebraic form, and while the algebraic form is often difficult for novice users to write/reason about, the more intuitive, and accessible, graphical notation does not scale to large systems. To overcome these limitations, we propose a new multi-perspective modelling language that compiles to BRSs. The language draws on the parallels between product types in programming languages and bigraphical nesting, and references in programming languages and bigraphical linking to provide a C-style interface. By developing a new language, we can enforce a strict separation of perspectives that is not present in the core bigraph theory.

Multi-perspective models can be useful for several reasons: 1) they aid in systems design by allowing perspectives to be written by different domain experts, while still allowing constraints between perspectives; 2) bigraphical models can be analysed formally to check system properties at design time; and 3) given a bigraphical model, we can synthesise automatically a (runtime) monitor, written in ocaml, that dynamically updates the model based on observed events, allowing verification and analysis of a running system.

While the original motivation for this work is to describe wireless sensor networks, we are keen to exploring the ideas of multi-perspective modelling in other domains. These are new ideas and we are looking for feedback on multi-perspective modelling, our proposed modelling language, and to find other systems domains where such an approach may prove useful.

[1] Benford, Steve, et al. “On Lions, Impala, and Bigraphs: Modelling Interactions in Physical/Virtual Spaces.” ACM Transactions on Computer-Human Interaction, vol. 23, no. 2, 2016, p. 9.

[2] Sevegnani, Michele, et al. “Modelling and Verification of Large-Scale Sensor Network Infrastructures.” 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), 2018.

[3] Calder et al. ”Making Sense of the World: Framing Models for Trustworthy Sensor-Driven Systems”, Computers, 7(4), 62, 2018.