Process algebra for event-driven runtime verification: a case study of wireless network management

Muffy Calder* and Michele Sevegnani

School of Computing Science, University of Glasgow, UK

Abstract. Runtime verification is analysis based on information extracted from a running system. Traditionally this involves reasoning about system states, for example using trace predicates. We have been investigating runtime verification for event-driven systems and in that context we propose a higher level of abstraction can be useful, namely reasoning at the level of user-perceived system events. And when considering events, then the natural formalism for verification is a form of *process algebra*.

We employ a universal process algebra that encapsulates both dynamic and spatial behaviour, based on Robin Milner's *bigraphs* [Milner09]. Our models are an extension of his bigraphical reactive systems. These consist of a set of bigraphs that describe spatial and communication relationships, and a set of bigraphical reaction rules that define how bigraphs can evolve over time. We have extended the basic formalism to bigraphical reactive systems *with sharing* [SevCal10], to allow for spatial locations that can overlap.

In this talk we present a case study involving wireless home network management and the automatic generation of bigraphical models, and their analysis, in real-time. Wireless home networking is chosen as our case study because it is notoriously difficult to install and manage, especially for non-expert users. The Homework network management system [SveKol+11] has been designed to provide user-oriented support in home wireless local area network (WLAN) environments. The Homework user interface includes drag and drop, comic-strip style interaction for users, and the information plane uses a stream database to record (raw and derived) events. Events include network behaviours such as detecting that a new machine has joined the network, resulting in new links and granting a DCHP lease, and user-intiated behaviours such as enforcing or dropping a policy. Policies forbid or allow access control; for example, a policy might block UDP and TCP traffic from a given site. All network and policy events (simple and derived) are recorded as a stream of tuples in the stream database. This part of the management system is illustrated in the left hand side of Figure 1.

On the right hand side of Figure 1 we depict our addtion to the Homework system: additional runtime verification components, and feedback

* This work is supported by the Engineering and Physical Sciences Research Council, under grant EP/F064225/1.

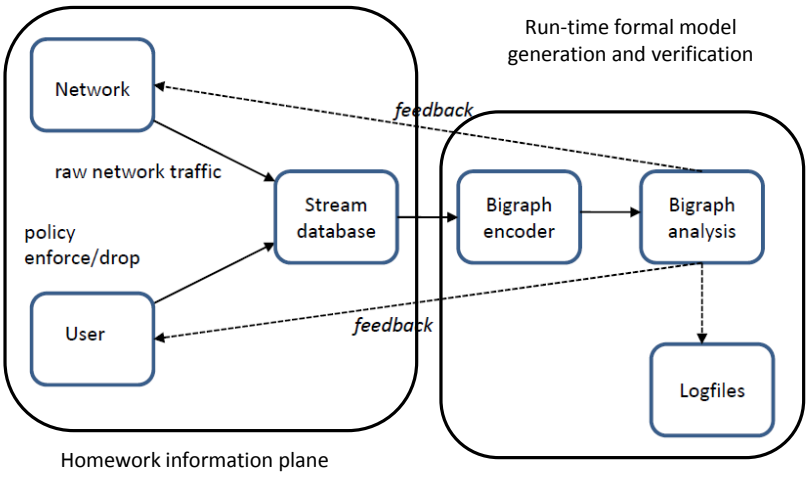

Fig. 1. Run-time model generation, analysis and feedback.

from the verification to the network and users. In this talk we focus first on the bigraphical representations of networks topologies, the encodings of events that modify topologies as bigraph reaction rules, and the encodings of access control policy enforcements and revokations as bigraph reaction rules, and second on how the two components are deployed at run-time and their interplay. Both components are part of a larger bigraph evaluation and rewriting toolkit [Bigrapher].

Briefly, the *Bigraph encoder* component encodes events (network topology or policy) as bigraphical reaction rules, in real-time, as they are stored in the stream database. The *Bigraph analysis* component has two roles. First, it generates the bigraphical representation of the current configuration of the WLAN, according to the sequences of reaction rules received from the *Bigraph encoder*. Namely, a sequence of bigraphs is generated. A simple example bigraph of a WLAN with one router (R), one machine (M1), and their respective wireless signals (S), is given in Figure 2. Second, it analyses the current configuration by checking predicates encoded as instances of bigraph matching. These predicates encapsulate properties required for correct encoding of topology or policy events, as well as system properties, including detecting configurations that violate user-invoked access control policies. Example predicates include: "Machine 01:23:45:67:89:ab is in the range of the router's signal", "Host Laptop has access to the Internet", and "TCP traffic is blocked for machine with IP address 192.168.0.3". The results are logged and fed back to the system, or to the user, when a verification fails. An explanation of the failure, or a counter-example can be displayed to a user, using the graphical bigraph notation. An indication of failure is also sent to the

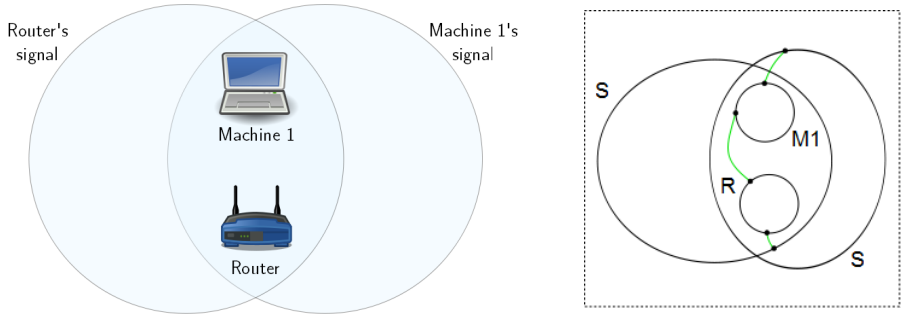

Fig. 2. Simple WLAN on the left and bigraph model on the right.

network, if appropriate, e.g. to deny activation of a policy, and/or simply stored in a logfile.
The encoding and analysis components have been implemented on the router itself, and we give some empirical evidence of runtime verification from experiments using actual and synthetic network data.

References

[Milner09] R. Milner. *The space and motion of communicating agents.* Cambridge University Press, 2009.

[SevCal10] M. Sevegnani and M Calder. Stochastic bigraphs with sharing. *Glasgow University Computing Science Technical Report TR-2010-310*, 2010.

[Bigrapher] Bigrapher. http://www.dcs.gla.ac.uk/∼michele/bigrapher.html.

[SveKol+11] J. Sventek, A. Koliousis, O. Sharma, N. Dulay, D. Pediaditakis, M. Sloman, T. Rodden, T. Lodge, B. Bedwell, and K. Glover. An Information Plane Architecture Supporting Home Network Management. *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management*, 2011.