A Full Operational Semantics of Asynchronous Relational Networks*

Ignacio Vissani[1,2], Carlos G. Lopez Pombo[1,2], Ionuţ Ţuţu[3,4], and José Luiz Fiadeiro[3]

[1] Department of Computing, FCEyN, Universidad de Buenos Aires, Argentina
[2] Consejo Nacional de Investigaciones Científicas y Tecnológicas, Argentina
[3] Department of Computer Science, Royal Holloway University of London, UK
[4] Institute of Mathematics of the Romanian Academy, Research group of the project ID-3-0439
ivissani@dc.uba.ar, clpombo@dc.uba.ar, ittutu@gmail.com, jose.fiadeiro@rhul.ac.uk

In the context of Service-Oriented Computing (SOC), the structure of software systems is intrinsically dynamic since the software applications that currently run over globally available computational and network infrastructures may require external services, and thus they may need to procure, on the fly, other applications that provide those services, and bind to them so that, collectively, given business goals can be fulfilled. In particular, development is no longer a process in which subsystems are developed and integrated by skilled engineers: in SOC, discovery and binding are performed, at run-time, by dedicated middleware.

In this talk, we provide an extension of the trace semantics of the service component algebra presented in [1] – Asynchronous Relational Networks (ARNs) – to account for the fact that, because of run-time discovery and binding, ARNs can be reconfigured at the same time as they compute. Our work follows the formalisation developed [2] in terms of hypergraphs whose nodes, or points, correspond to structured sets of messages that can be exchanged between the network components, and whose hyperedges capture those elements of networks that account for computation and/or communication, i.e. the processes and the channels attached to them through connections.

Every ARN determines sets of interaction-points whose associated messages can be regarded as information that is provided or required by the network. This supports a notion of execution of an ARN through which a network can discover and bind, at run time, to service modules of a given repository whenever it triggers an action related to the publication or the delivery of a requires-message.

Within this context, we define a new operational semantics for ARNs that relies on specialised forms of temporal-logic concepts such as execution fragment, path and trace. More precisely, we formalise the execution of an ARN as a sequence of networks and local states of their components; such an execution starts with the ARN under consideration and the initial states of its processes

* This work has been supported by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS)

and channels, and unfolds through the effect of publication/delivery actions, which may trigger the discovery and binding of other networks.

This means that executions of an ARN can be regarded as sequences of generalised states connected by a transition relation, where a transition from one state to another can be the result of either the execution of an internal transition of the ARN, i.e. the effect of an action that is not associated with any of the ARN's provides- or requires-points, or the binding of one of the requires-points of the ARN to a provides-point of a service module taken from the repository as a consequence of executing an action associated with that requires-point.

The approach we propose here combines, in an integrated way, the operational semantics of processes and channels, and the dynamic reconfiguration of ARNs. It captures a full operational semantics of ARNs through labelled transition systems built from the local semantics of the considered networks, together with the semantics of those ARNs that are provided by a repository of services by means of stepwise execution, discovery and binding. This gives us a more refined view of the execution of ARNs than the logic-programming semantics of [2]. Moreover, it allows us to use various forms of temporal logic to express, and even to verify through standard model-checking techniques, properties concerning the behaviour of ARNs that are more complex than those considered before. It is possible, for example, to determine whether or not a given service module of a repository is used or may be used during the execution of an ARN, or to identify the differences between the nondeterministic behaviour of a component, reflected within the execution of an ARN, and the nondeterminism that arises from the discovery and binding to other ARNs.

References

1. Fiadeiro, J.L., Lopes, A.: An interface theory for service-oriented design. Theoretical Computer Science **503** (2013) 1–30
2. Ţuţu, I., Fiadeiro, J.L.: A logic-programming semantics of services. In Heckel, R., Milius, S., eds.: Algebra and Coalgebra in Computer Science. Volume 8089 of Lecture Notes in Computer Science., Springer (2013) 299–313