Automata for partial binding of services ⋆

Carlos G. Lopez Pombo[1,2], Ignacio Vissani[1], and Ezequiel Davidovich Caballero[1]

[1] Universidad de Buenos Aires. School of science, Department of computing.
[2] CONICET–Universidad de Buenos Aires. Instituto de Investigación en Ciencias de la Computación (ICC).

Distributed software resulting from emerging paradigms such as *service-oriented computing* (SOC), Cloud/Fog computing and the Internet of Things are transforming the world of software systems in order to support applications able to respond and adapt to the changes of their execution environment, giving impulse to what is called the API's economy. The underlying idea of the API's economy is that it is possible to construct software artifacts by composing services provided by third parties and previously registered in repositories. This envisages a generation of applications running over globally available computational resources and communication infrastructure, which, at run-time, are dynamically and transparently reconfigured by the intervention of a dedicated middleware, subject to the negotiation of a *Service Level Agreement – SLA*[1].

Under this paradigm software services are accessed by their API. Thus, a key element is the existence of formal languages, together with associated analysis techniques, capable of fully expressing the API behavioral contract.

Asynchronous Relational Nets [2] is a language for describing the orchestration of services, supporting the explicit dynamic reconfiguration of a system by declaring ports behaving either as provide points or as require points, depending on their role in the service binding and execution. In [3] it was given a formal operational semantics supporting the transparent run-time composition of services. In [4], *Communicating Finite State Machine* [5] were used to label provides points and *Global Graphs* [6] to label communication channels, in order to describe a procedure for automatically checking interoperability of services.

The way in which these formalisms are defined prescribe that correctness of the communication (usually reduced to the absence of certain configurations[7]: 1) *deadlock*: a participant is in a state in which it can only consume messages from participants whose corresponding message queues are empty, 2) *unspecified reception*: a participant is in a state in which it can only consume messages from participants which are not in the corresponding message queues, and 3) *orphan message*: all participants are in a state of no outgoing transition and there is a non-empty buffer.) can only be asserted in the presence of all the participants involved. Thus, when they are used for formalising an interoperability check, all participants must commit to be bound at the beginning of the communication.

⋆ Carlos G. Lopez Pombo's research is supported by Agencia Nacional de Promoción Científica y Tecnológica by grant PICT 2013-2129, and Consejo Nacional de Investigaciones Científicas y Técnicas by grant PIP 11220130100148CO.

Generalised Multi-party Compatibility (GMC), defined in [7], is a sufficient condition for the correctness condition detailed above, requiring the system to satisfy: 1) *representability*: each trace t of each participant P of the system is "represented" by a trace t' in the global transition system in which the action taken by P in t appear in t' in the exact same order, 2) *branching property*: each branching configuration of the global transition system either consists of independent transitions (that is they all interleave) or they represent a distributed choice with a unique participant making the choice, and any other participant involved in it has a different first action on each two different branches of the choice, and 3) for each participant P involved in a choice, there cannot be a race condition between the messages that P can receive (only one input message gets to P during the choice.

In this work we study: 1) a new class of CFSMs, called *Multichannel Communicating Finite State Machines – mCFSMs*, with an explicit definition of the communication channels enabling, for a participant, the possibility of having more than one channel with the other participants 2) a definition of the GMC property for systems of mCFSMs, 3) a class of *Asynchronous Communicating Automatas – ACAs* with the capability of internalising the communication as read / write operations on internal buffers, enabling partial composition of communicating automata, and 4) a method for mapping an ACA to a mCFSM providing a checking mechanism of the GMC property for the class of ACA.

References

1. Fiadeiro, J.L., Lopes, A., Bocchi, L.: An abstract model of service discovery and binding. Formal Aspects of Computing **23**(4) (2011) 433–463
2. Fiadeiro, J.L., Lopes, A.: An interface theory for service-oriented design. In Giannakopoulou, D., Orejas, F., eds.: Proc. of 14th International Conference Fundamental Approaches to Software Engineering – FASE 2011. Vol. 6603 of LNCS, Springer-Verlag (2011) 18–33.
3. Vissani, I., Lopez Pombo, C.G., Ţuţu, I., Fiadeiro, J.L.: A full operational semantics for asynchronous relational networks. In Diaconescu, R., Codescu, M., Ţuţu, I., eds.: Proc. of 22st International Workshop on Algebraic Development Techniques (WADT 2014). Vol. 9463 of LNCS, Romania, Springer-Verlag (September 2015) 131–150
4. Vissani, I., Lopez Pombo, C.G., Tuosto, E.: Communicating machines as a dynamic binding mechanism of services. In Gay, D., Alglave, J., eds.: Proc. of 8th International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES. Vol. 203 of Elect. Proc. in Theoretical Computer Science. (April 2016) 85–98
5. Brand, D., Zafiropulo, P.: On communicating finite-state machines. Journal of the ACM **30**(2) (1983) 323–342
6. Deniélou, P.M., Yoshida, N.: Multiparty session types meet communicating automata. In Seidl, H., ed.: Proc. of 21st European Symposium on Programming, ESOP 2012. Vol. 7211 of LNCS, Berlin, Springer-Verlag (2012) 194–213.
7. Lange, J., Tuosto, E., Yoshida, N.: From communicating machines to graphical choreographies. In Rajamani, S.K., Walker, D., eds.: Proc. of 42rd. Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. POPL '15, USA, ACM (2015) 221–232