CONNECT: Emergent Connectors for Eternal Software Intensive Networked Systems

Valérie Issarny, Consortium Connect

CONNECT: Emergent Connectors for Eternal Software Intensive Networked Systems

CONNECT Consortium

http://connect-forever.eu

Contact: Valérie Issarny, Project leader

INRIA Paris-Rocquencourt, France

Valérie.Issarny@inria.fr

ABSTRACT

The CONNECT Integrated Project aims at dropping the interoperability barrier by adopting a revolutionary approach to the seamless networking of digital systems, that is, synthesizing on the fly the connectors via which networked systems communicate. CONNECT enables the dynamic synthesis of CONNECTors by introducing a formal foundation for connectors, which allows learning, reasoning about and adapting the interaction behavior of networked systems.

Keywords

Interoperability, formal methods, dependability, middleware, specification generation, system synthesis.

1. INTRODUCTION

Our everyday activities are increasingly dependent upon the assistance of digital systems that pervade our living environment. Key technologies such as the Internet, the Web and wireless computing devices and networks are now calm technologies in the sense of Marc Weiser's definition [1]. They can indeed be qualified as ubiquitous, even if converged computing and networking technologies have still not reached the maturity envisioned by the ubiquitous computing and subsequent pervasive computing and ambient intelligence paradigms.

However, the ubiquitous computing vision is hampered by the often extreme level of heterogeneity in the underlying infrastructure, which in turns impacts on the ability to seamlessly interoperate. Indeed, the efficacy of integrating and composing networked systems is proportional to the level of interoperability of the systems' respective underlying technologies [2]. This leads to a landscape of technological islands of networked systems, although interoperability bridges may possibly be deployed among them. Further, the fast pace at which technology evolves at all abstraction layers increasingly challenges the lifetime of networked systems in the digital environment.

The CONNECT project that started in February 2009 aims at dropping the heterogeneity barriers that prevent networked systems from being eternal, thus enabling the continuous composition of networked systems to respond to the evolution of functionalities provided to and/or required from the networked environment, independently of the embedded software technologies.

The next section further discusses the design rationale of CONNECT, with its goal of offering future-proof universal interoperability. Section 3 then briefly introduces the core concept of CONNECT, which is to dynamically realize the networking of systems based on the systems' respective behavior, as opposed to underlying technologies. Section 4 concludes with a summary of the challenges that will be addressed by the CONNECT project.

2. FUTURE-PROOF INTEROPERABILITY

Middleware stands as the conceptual paradigm to effectively network together heterogeneous systems, specifically providing upper layer interoperability. It is a distributed system software layer that resides between applications and the underlying operating systems, network protocol stacks, and hardware. The primary role of middleware is to functionally bridge the gap between application programs and the lower-level hardware and software infrastructure in order to coordinate how application components are connected and how they interoperate, especially in the networked environment. As a matter of fact, middleware is yet another technological block, which also creates islands of networked systems.

Interoperable middleware have been introduced to overcome middleware heterogeneity. However, the solutions remain rather static, requiring either the use of a proprietary interface or a priori implementation of protocol translators. In general, interoperability solutions solve protocol mismatch among middleware at the syntactic level, which is too restrictive. This is even truer when one considers the many dimensions of heterogeneity, including software, hardware and networks, which now arise in ubiquitous networking environments, and that require fine tuning of the middleware according to the specific capacities embedded within the interacting parties. Thus, interoperable middleware can at best solve protocol mismatches arising among middleware aimed at a specific domain. Indeed, it is not possible to a priori design a universal middleware solution that will enable effective networking of digital systems, while spanning the many dimensions of heterogeneity now arising in networked environments and which will also increase dramatically in the future.

A revolutionary approach to the seamless networking of digital systems is to synthesize on the fly the connectors via which networked systems communicate. The resulting *emergent connectors* (or CONNECTors) then compose and further adapt the interaction protocols run by the connected systems, which realize application- down to middleware-layer protocols. Hence, thanks to CONNECT, networked digital systems will survive the erosion and further emergence of interaction protocols and step towards a post middleware world.

CONNECT enables the dynamic synthesis of CONNECTors by introducing a formal foundation for connectors, which allows learning, reasoning about and adapting the interaction behavior of networked systems. Further, compared to the state of the art foundations for connectors, CONNECT represents a drastic shift by learning, reasoning about and synthesizing connector behavior *at run-time*. Indeed, the use of connector specifications pioneered by the software architecture research field has mainly been considered as a *design-time* concern [3], for which automated reasoning is now getting practical even if limitations remain. On the other hand, recent efforts in the semantic Web domain has brought ontology-based semantic knowledge and reasoning at run-time but networked system solutions based thereupon are currently mainly focused on the functional behavior of networked systems, with few attempts to capture their interaction behavior as well as non-functional properties. In the approach taken by CONNECT, the interaction protocols' (both application- and middleware layer) behavior will be learnt by observing the interactions of the networked systems, where ontology-based specification and other semantic knowledge will be exploited for generating CONNECTors on the fly.

3. BEHAVIORAL NETWORKING

In our view, the key to eternal networked systems is to make their networking agnostic to their specific technological middleware[1]. Then, networked systems should seamlessly integrate and compose with networks of systems according to functional and non-functional properties each one of them provides to and requires from the digital environment rather than according to their underlying middleware technology.

Specifically, in the CONNECTed world, networked systems run discovery protocols to advertise their presence and locate systems with which they need to interact at a specific time and place. The initial networking association of systems is then solely based on the systems' provided and required application-specific behavior, which is characterized semantically, thanks to ontologies. Following, CONNECT enablers present in the networked environment set up needed CONNECTors among the interacting systems, at run-time, effectively bridging semantically their respective protocols, from the application- down to the middleware-layer. As a result, networked technology-dependent systems interact behaviorally within the CONNECTed world, based on their respective interaction semantics. In light of the above, key concepts of CONNECT are as follows:

- As pioneered by the pervasive computing domain, the *dynamic networking of digital systems* is at the heart of making networked applications evolvable and further eternal. In this way, a networked system is able to compose with others, based on the respective provided and required functionalities within the network, without requiring a priori knowledge about the systems that are actually networked at a given time and place. Then, our only assumption is that a networked system runs some discovery protocol and further characterizes provided/required (application-layer) networked functionalities using ontologies.
- CONNECT sustains future-proof dynamic networking among any digital systems, from the legacy to the yet-to-come, by dynamically generating CONNECTors, thus bringing universal interoperability. Emergent connectors are synthesized on the fly according to the behavioral semantics of application- down to middleware-layer protocols run by the interacting parties. Also, emergent connectors are dependable, unobtrusive, and evolvable to indeed meet the promise of eternality, while not compromising the quality of software applications.

CONNECTors are implemented through a comprehensive dynamic process based on: (i) extracting knowledge from, (ii) learning about and (iii) reasoning about, the interaction behavior of networked systems, together with (iv) synthesizing new interaction behaviors out of the ones exhibited by the systems to be made interoperable, and further (iv) generating and deploying corresponding CONNECTors implementations to actually realize networking of the involved systems.

4. CONCLUSION

The core objective of CONNECT is to effectively support the aforementioned dynamic process for the actual implementation and deployment of emergent connectors. This requires devising: (i) a semantic foundation of connectors enabling automated reasoning and synthesis of their behavior, and (ii) associated networked enablers, which are the actual actors of the connector generation dynamic process. Emergent connectors specifically result from a learning, reasoning and synthesis process that is able to elicit the "modus operandi" for carrying out the interoperable communication.

The above raises a set of unique challenges in the area of software systems engineering, from theoretical foundations to specify the interaction behavior of networked systems to run-time methods and tools to turn specifications into running protocols, and vice versa. These are the challenges on which the CONNECT project will concentrate.

5. ACKNOWLEDGMENTS

The project CONNECT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the ICT theme of the Seventh Framework Programme for Research of the European Commission .

6. REFERENCES

[1] M. Weiser. "The computer for the twenty-first century". In Scientific American, 1991.

[2] A. Tanenbaum and M. van Steen, "Distributed Systems: Principles and Paradigms", Prentice-Hall, ISBN 0-13-239227-5, 2007.

[3] M. Shaw and D. Garlan. "Software Architecture: Perspectives on an emerging Discipline". Englewood Cliffs, NJ, Prentice Hall, 1996.

[1]Middleware is referred to here as any software solution, from very minimal to very sophisticated, below the application layer and above the transport layer. In other words, we refer to middleware as the generic realization of the OSI presentation and session layers.