Thesis Proposal

For the obtention of the grade of docteur ès Sciences Économiques et Sociales, mention Systèmes d’Information

Manuel Oriol

October 2002

Candidate: Manuel Oriol
Thesis director: Prof. Dimitri Konstantas
Commission:
Prof. Michel Léonard, Prof. Dimitri Konstantas, Dr Ciarán Bryce.


An increasing number of applications need to run anytime. These applications may be servers that perform a wide range of tasks on the Internet as well as applications managing physical devices like mobile phones, cars, satellites or even nuclear power plants. Nevertheless, such applications need to evolve over time, when bugs are discovered and fixed, or when functionalities are added or when topology changes. The primary goal of this thesis is to show that it is possible to program applications such that their evolution may be seamlessly integrated smoothly at runtime without stopping either the application that evolves or other applications that use it.

We describe dynamic systems as the wide set of applications that may evolve over time part by part. This set includes component-based applications, and in particular agent-based applications, as well as peer-to-peer (P2P) systems for which the topology evolution is a natural behaviour. Our approach consists in modularizing applications and let them be cut in several independent parts that may be considered as components, modules, agents or network nodes; we designate them by the generic term of entities. Fundamentally entities provide services and may request services. Our architecture relies on the principle that entities, willing to invoke a service, do not designate it: they describe what they want. Entities, providing services, describe and announce them. The communication infrastructure chooses the best service to invoke dynamically.

Our approach concentrates on defining: (a) a mini-language for describing services, it is used both by the caller and the callee of a service; and (b) a general communication infrastructure for invoking, announcing and removing services. In the mini-language, we define how to describe services in order to have associative and anonymous invocations. In the general infrastructure, we describe an asynchronous invocation mechanism, and integrate the anonymous
and associative aspects. A service invocation is associative, since the desired service description is compared to the proposed services, and the best adapted of the proposed services is actually invoked. The invocation itself is then anonymous as it is not possible to have any reference or name describing the entity that serves or requires a service. Invocations are asynchronous because calling entities do not wait for the invocation’s end. They continue their execution and possibly create a return service to get an answer.

The first part of the thesis presents related work in this area. The second part of this thesis defines our model: service description, service invocation and the communication infrastructure. In a third part of this thesis, we describe two implementations of our model: (a) a central infrastructure that allows local component-based applications to evolve dynamically; and (b) a distributed infrastructure for allowing wide-scale applications to evolve dynamically. The first implementation shows that these infrastructure principles are adapted for making entities, being components of the application, evolve at runtime. The second one shows that our approach may be used successfully for managing distributed invocations in networked dynamic systems (entities being nodes of the network). In a fourth part of the thesis we show how these concepts may be applied to other domains like Peer-to-Peer (and JXTA in particular); or Web services. Finally, we evoke the research leads that our contributions open.

References


