

Design Abstractions for Context-Awareness

Antónia Lopes¹ and José Luiz Fiadeiro²

¹Department of Informatics, Faculty of Sciences, University of Lisbon
Campo Grande, 1749-016 Lisboa, PORTUGAL
mal@di.fc.ul.pt

² Department of Computer Science, University of Leicester
University Road, Leicester LE1 7RH, UK
jose@fiadeiro.org

Abstract

One of the major challenges in modern distributed computing is to deal with highly dynamic environments. As components are entitled to move across a network whose nodes can, themselves, be mobile, the availability and responsiveness of resources and services are often unpredictable and out of their control [GCI01]. For instance, in the presence of mobility of code, computational resources such as CPU and memory are no longer fixed as in traditional distributed systems. As pointed out in [Bou03], the same happens with services: a piece of mobile code will fail to link with a required library when visiting a site that does not provide it. Other kind of resources, such as network connectivity and bandwidth, can also affect the behaviour of mobile computing systems in a fundamental, often critical way.

In this setting, it is no longer reasonable to consider that attempting to use a resource that is absent, or gain access to a service that is unavailable, is an exception. Instead, systems should have the means to observe their environments and should be developed taking into account the different conditions in which they can be executed. *Context-awareness* is the emergent computing paradigm that addresses this kind of concerns as raised by the development and deployment of mobile systems.

By *context* one refers to the part of the environment of a running system that may affect its behaviour. Different kinds of applications typically require different notions of context. Moreover, as applications evolve, the notion of context on which they rely may change. Hence, it is important that formalisms for designing mobile systems consider contexts as first-class design entities and support their explicit design and evolution. If, instead, a specific notion of context is forced through the formalism, as happens with Ambients [CaG98], the encoding of a different notion of context, if at all possible, is in general cumbersome and becomes tangled with other aspects. Furthermore, the explicit modelling of a notion of context allows this aspect to be progressively refined (through the addition of detail) without interfering with the parts of the system already designed, which is essential for taming the complexity of designing mobile systems.

As part of a longstanding collaboration effort on the architectural modelling of complex systems, we are currently investigating essential features of contexts and proper abstractions for modelling context-awareness. Under the assumption that the context that a component perceives is determined by its current position, we are exploring the use of algebraic techniques for defining design primitives through which different notions of context can be modelled explicitly as part of the application domain.

Our approach provides means for modelling explicitly how contexts affect the behaviour of systems. Each specific notion of context defines a specific set of constructs that can be used in the specification of the enabling conditions of system actions. In this way, we can model context-aware patterns of computation, coordination and mobility. On the other hand, we support the introduction of context-awareness in architectural models in a non-intrusive way: aspects of context-awareness can be added without interfering with the options that are made at the level of other aspects such as those concerned with Computation, Coordination and Mobility.

We present our approach over CommUnity, a language and associated mathematical semantics that we have been developing to support architectural design of complex systems [FLW03]. Within the GC1 project IST-2001-32747 (AGILE – Architectures for Mobility), CommUnity has been recently extended in order to support the description of mobile systems [LFW02]. This extension assumes a specific notion of context – one that is centred on the notions of connectivity and reachability of positions; it does not address the availability of computational resources and services at the locations in which components are placed. In order to support the definition of application-specific notion of contexts, these extensions of CommUnity are now being taken one step further. It is this work in progress that we propose to present.

References

- [GCI01] IST Global Computing Initiative, <http://www.cordis.lu/ist/fet/gc.html>, 2001.
- [Bou03] G.Boudol, "ULM A Core Programming Model for Global Computing", available at <http://www-sop.inria.fr/mimosa/personnel/Gerard.Boudol.html>.
- [CaG98] L.Cardelli and A.Gordon, "Mobile Ambients", in Nivat (ed), *FoSSACs'98*, LNCS 1378, 140-155, Springer-Verlag, 1998.
- [FLW03] J.L.Fiadeiro, A.Lopes and M.Wermelinger, "A Mathematical Semantics for Architectural Connectors", in *Generic Programming*, R.Backhouse and J.Gibbons (eds), LNCS 2793, 190-234, Springer-Verlag 2003.
- [LFW02] A.Lopes, J.L.Fiadeiro and M.Wermelinger, "Architectural Primitives for Distribution and Mobility", *Proc. SIGSOFT 2002/FSE-10*, 41-50, ACM Press, 2002.