Chapter 3
Dynamic Reconfiguration of Middleware for Ubiquitous Computing

Antonio Corradi
University of Bologna, Italy

Enrico Lodolo
University of Bologna, Italy

Stefano Monti
University of Bologna, Italy

ABSTRACT
Ubiquitous computing scenarios call for middleware support solutions able to cope with changes in environment conditions and user requirements. Changes greatly impact on the application logic the middleware is able to carry out (content/service adaptation, multichannel content delivery, mobility management, and so on), but also on the non-functional support logic middleware exploits to enforce the application logic itself (e.g., naming, persistence, communication infrastructures). In this work, the authors propose a novel middleware for ubiquitous computing scenarios that is able to reconfigure both application and non-functional features, in order to cope with increasingly complex and heterogeneous ubiquitous and pervasive landscapes.

INTRODUCTION
Recent technology advances in wired and wireless network connectivity and increasingly powerful and rich mobile end user devices are more and more making concrete the once only envisioned ubiquitous and pervasive computing scenarios. Users are more and more able to exploit services and contents at any time, anywhere, and by means of any device. Furthermore, services and contents much more tailor to fit user needs and characteristics (e.g., device in use), and to adapt to environment conditions (such as network connectivity type, status, user location ...).
Heterogeneity and dynamic nature of these scenarios make system reconfiguration and adaptation crucial and challenging tasks to provide users with a dynamic support to tailor applications to specific user and environment conditions.

Mainstream approaches typically adopt a service-oriented perspective to push application logic outside of the middleware support layer and model such logic as a set of small and well-decoupled business units with well-defined interfaces. The middleware layer thus implements only core support features - typically not oriented to a specific service - such as naming, persistence, and messaging, and it is in charge of conveniently aggregating and coordinating application logic pieces. When environment conditions change, the intrinsic modularity of this approach allows to easily rearrange such compositions by changing, adding, and removing business units.

However, this approach lacks to face a fundamental facet of the problem too often neglected: heterogeneity and dynamic nature of ubiquitous and pervasive scenarios typically have a great impact also on non-functional aspects of middleware platforms. Changes in the environment conditions (e.g., network conditions and available devices) obviously greatly influence runtime behavior of the non-functional part of support features. As an example, traditional middleware platforms adopt underlying asynchronous messaging facilities to help coordinate higher level entities of the platform itself; in case of network congestion, it could be necessary to switch from a connection-oriented message transport implementation to a connectionless one, to save bandwidth and to speedup communication, so renouncing to delivery guarantees.

In our opinion, a truly viable ubiquity support platform must be able not only to reconfigure its application logic layer, but also to reconfigure its offering of non-functional support layer features, for instance by allowing the coexistence of multiple different alternatives to choose among and to arrange according to current environment conditions.

The design of such a dynamic, flexible, and reconfigurable ubiquitous support system is inherently complex, since both the platform elements (application and non-functional logic) to reconfigure and the reconfiguration tasks themselves - determining whether, when, and how to reconfigure elements - are extremely heterogeneous, dynamic, and can vary in time to react to changes in system status and user requirements, and to adapt to availability of novel technologies and contents.

Separation of concerns (Dijkstra, 1982) is a long established approach to address increasingly complex problems with the right abstraction level and in a manageable way, and to promote the design of extremely extensible and flexible architectural solutions. In our opinion, separation of concerns is crucial in the design of a fully reconfigurable ubiquity support platform: this paper describes the design and implementation of a novel ubiquitous middleware whose architecture stems from separation of concerns principles to flexibly and dynamically support the reconfiguration and adaptation of both application and non-functional features to changes in user requirements and environment conditions.

The following section provides background knowledge of ubiquity issues and reconfigurable systems, highlighting also the most significant contributions in the domain of reconfigurable ubiquitous systems. Design Principles section sketches out the main principles followed in the design of our proposal. The Architecture section describes the resulting overall architecture, whereas Reconfigration Details section depicts the key choices in designing reconfiguration logic. Implementation section provides relevant implementation details and Case Study section provides an example of a reconfiguration case study. Finally, the last section reports conclusions and illustrates further work our proposal has led to.
Related Content

Interactive Genetic Algorithms for Optimal Assignment of Blocks into Workspaces of Shipbuilding Industry
[www.igi-global.com/article/interactive-genetic-algorithms-for-optimal-assignment-of-blocks-into-workspaces-of-shipbuilding-industry/130757?camid=4v1a](www.igi-global.com/article/interactive-genetic-algorithms-for-optimal-assignment-of-blocks-into-workspaces-of-shipbuilding-industry/130757?camid=4v1a)

Differential Evolution-Based Optimization of Kernel Parameters in Radial Basis Function Networks for Classification
[www.igi-global.com/article/differential-evolution-based-optimization-kernel/75825?camid=4v1a](www.igi-global.com/article/differential-evolution-based-optimization-kernel/75825?camid=4v1a)

Elements of Utility Theory
[www.igi-global.com/chapter/elements-utility-theory/74435?camid=4v1a](www.igi-global.com/chapter/elements-utility-theory/74435?camid=4v1a)

An Improved Fuzzy Voting Scheme for Fault Tolerant Systems
[www.igi-global.com/article/an-improved-fuzzy-voting-scheme-for-fault-tolerant-systems/131447?camid=4v1a](www.igi-global.com/article/an-improved-fuzzy-voting-scheme-for-fault-tolerant-systems/131447?camid=4v1a)