Middleware for Distributed Services in Ambient Intelligence Environments

Overview

Middleware for Distributed Services in Ambient Intelligence Environments is a set of programming libraries and programs (services) that constitute an indivisible platform offering a comprehensive abstraction over the complexities and potential heterogeneity of the target problem domain.

Middleware enables programs written in different programming languages to interoperate seamlessly which is especially important in an AmI environment, where the basic system is built using diverse technologies from diverse research fields that traditionally utilize different programming languages.

Target Applications

A multi-language environment is essential for enabling many different research fields to expose their individual technologies and allow access to them through a uniform, well-defined set of methods. For this reason, it was deemed essential for the middleware to target a wide range of popular programming languages.

Middleware for Distributed Services provides libraries and tools to enable software developers to create services with a true Object-Oriented Application Programming Interface (API). These services can be developed and used from any program written in any of the supported programming languages which are: (a) C++, (b) .NET languages, (c) Java, (d) Python, and (e) Flash/ActionScript. The middleware effectively allows services to be distributed across the network, hiding the details of network connections and data serialization from the programmers.
Additional Information

The service middleware ecosystem comprises of the following four **software components**:

a) software libraries for facilitating communications and service distribution,

b) a set of core services for providing discovery and deployment functionality for the services infrastructure,

c) development tools that facilitate the creation, usage and deployment of services, and

d) client programs for managing and monitoring the computers that are used for running services.

The main **implementation characteristics** in the service middleware ecosystem are summarized below:

- Full support for creating and consuming services in five popular programming languages (C++, .NET languages, Java, Flash/ActionScript, Python).

- Synchronous request/response and asynchronous event-based communication.

- Statically checked type-safe invocation of Service methods.

- Object-oriented programming model for creating and consuming services.

- Automated service deployment and failure recovery.

- Tools for describing, managing, and configuring the deployment and update of services.

Contact details: Constantine Stephanidis  
Yannis Georgalis

cs@ics.forth.gr

www.ics.forth.gr/ami