Anamika: Distributed Service Composition Architecture for Pervasive Environments

Dipanjan Chakraborty, Anupam Joshi
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
{dchakr1,joshi}@cs.umbc.edu

Technological advances in semiconductor processing and design as well as wireless networking are leading us towards the vision of Pervasive Computing. We envision that in the (near) future, devices all around a person, either embedded as a part of smart spaces, or being carried by other people in the vicinity, will provide an array of services that she might want to use. Development of customized services by integrating and executing existing ones has received a lot of attention in the last few years with respect to wired, infrastrutcure based web-services. However, service composition in web-based environments is performed in a centralized manner with the help of a central registry/composition engine. Moreover, wired infrastructure-based service discovery and composition architectures do not take into consideration various factors arising for services in a pervasive environment. We enumerate some of the various factors that are most important and should be considered in any service composition architecture in pervasive environments.

- **Heterogeneity:** Devices with varying resources and capabilities exist in the geographical vicinity of one another.
- **Limited Connectivity:** The devices are connected to other devices in their vicinity with the help of short-range ad-hoc connections. Each device is directly connected to only a *few* devices in its vicinity (devices within its radio range) and *not* to all devices in the whole neighborhood.
- **Mobility:** Many of devices are mobile, with varying mobility rate. Hence they may move out of the environment at any point of time.
- **Service Availability:** Each device has one or more *services* that provides either resources/data or computation capabilities and can be invoked over the network from other remote devices. The term *service* can be defined as any hardware or software entity that resides on any mobile device or platform and has distinct functional attributes that characterizes it. Multiple services can combine themselves to produce a qualitatively different new service.
- **Reliability:** Devices and hence services may have a short "switched-on" time, and are also sometimes unable to process remote requests due to system overloading. This, combined with mobility and the fact that the underlying ad-hoc network is disconnection and error prone makes these services unreliable.
- **Request Origin:** Request to compose multiple services (a composite request) could originate from any one such device in the environment.

In this project, we introduce a distributed fault-tolerant *reactive* technique to carry out service composition in pervasive environments. Reactive service composition refers to the type of composition that is executed only upon request. It may be mentioned here that research in Service Composition primarily focuses on two issues, viz. *Request Planning* and *Execution of the Composite Request.*

- **Request Planning:** This deals with coming up with a process model or plan specifying the multiple different services that need to be executed to satisfy the query. There has been considerable research work in the fields of workflow management, planning to handle this problem. Recent efforts in the field of web services have tried to come up with languages like WSFL, WSDL, DAML-S to describe a complex request in terms of the individual services required to satisfy it.
- **Execution of Composite Request:** This refers to the actual process of discovery, integration and execution of the multiple components/services in a planned order (as specified by a process model) to execute a composite request. This problem can be relatively easily tackled in the context of static wired networks with the help of a centralized architecture for execution of composite requests. However, such centralized architectures are unsuitable for pervasive

environments with the different characteristics mentioned at the beginning of this section.

Our project focuses on developing a distributed architecture for service composition in pervasive environments. Thus its main contribution is in the development of the architecture and not on request planning. Central to our system is the concept of a distributed broker that can execute in any node in the ad-hoc environment. A *broker* is a device that coordinates the discovery, integration and execution of different services/components to calculate the result for a composite request. An individual broker handles each composite service request in the environment, thus making the design of the system immune to central point of failure. A broker may be selected based on various parameters like resource capability, geometric topology of the nodes and proximity of the node to the services that are required to compose a particular request. Our composition architecture takes into consideration the resources/device capabilities of the neighborhood or neighboring devices to distribute the task of *executing a composite request*. Our composition architecture thus primarily *deals with* the *execution of a composite request*. The problem of *Request Planning* is outside the scope of our present work. The current implementation of our architecture assumes that the process model of the composite task would be provided to it. It is a relatively straightforward exercise to plug in an external planning system that will provide the system with a process model of execution for a composite service.

We have developed a proof-of-concept level system (using Bluetooth as the underlying networking layer) that is capable of reactive service composition in an ad-hoc environment by dynamically discovering, integrating and executing individual services available in nodes that are connected in an ad-hoc manner. We have developed a decentralized peer-to-peer caching based distributed service discovery architecture that helps the composition engines to discover services in the vicinity. Services are described in a semantic manner (with an ontology developed using DAML+OIL) and the discovery process is capable of handling semantic knowledge-based search, apart from the standard unique identifier, attribute or interface-based searches.

We describe a composite service using DAML-S semantic markup language. The individual services are described based on inputs, outputs, functionality classification, functional similarity to other services and invocation mechanisms. The service description also incorporates platform specific information like the current execution platform and resources available in that platform. When a request for a composite service comes to the composition engine, it takes help of the underlying semantic service discovery mechanism to obtain information about the availability of the necessary services in the vicinity. The composition engine decides on the best available platform to carry out the composition based on the resource availability of the platforms. It obtains this information while doing service discovery. Each request in this environment is thus be assigned a separate platform to carry out the composition. Reactive service composition in highly dynamic environments where the services around a device are rapidly changing (due to mobility) is done by discovering and executing services in a sequential manner. In environments where the services are liable to be available for a longer duration (say a group meeting in a conference room), the discovery process is done first, followed by execution where certain optimizing criteria can be applied. We have carried out a few scalability experiments with our current implementation. This is an ongoing work and we aim to implement two other techniques of service composition in our system. We have not mentioned the related work and references due to lack of space.