Combining Open APIs(Parlay/JAIN) & Software Agents for Next Generation Mobile Services

Deepak Rautela [1], Gert Markwardt [2], and Ahmed Kamel [1]

[1] North Dakota State University, Computer Science Department, IACC 258,
Fargo, ND 58105, USA
{ Deepak.Rautela, Ahmed.Kamel }@ndsu.nodak.edu

[2] SIEMENS AG, Information and Communication Mobile, Siemensdamm 50,
Berlin 13629, Germany
gert.markwardt@bln1.siemens.de

Abstract. The Telecom/Wireless world is rich in network capabilities and is trusted by the consumer to provide reliable, stable services. The IT world is rich in applications and developers. The strengths of these two domains could be combined and new services can be developed utilizing underlying network capabilities. The Parlay Group defines a set of Open APIs that support applications external to the secure network space. The Java APIs for Integrated Networks Community is defining a Java version of the Parlay API. The above scenario will provide wireless application developers to produce new and exciting applications for end users. Wireless networks have less bandwidth and more latency compared to wired networks. In this regard Mobile Software Agents can provide better support for wireless devices. In this paper we have researched the feasibility of the above architecture where applications developed using Open APIs like Parlay/JAIN can utilize the advantages of Software Agents.

1 Introduction

In this section we give a brief introduction about the paper which is divided into two parts. The first part is a general description of Software Agents and Open APIs (Parlay/JAIN) for Next Generation Networks. The second part is a brief description of the architecture that we propose in this paper.

1.1 Problem

The second-generation mobile communications technology in use today has been “voice centric” offering the benefits of person-to-person speech communication anytime, anywhere. The voice centric feature is offered by circuit switched technology using the “numbering” as an addressing and an identification of end users. The global explosion in Internet popularity shows that the users anticipate the discovery of multimedia communications and access to instant information both in their personal and business lives. Thus the major source of revenue resides primarily in the revolution in the service application domain. The introduction and usage of the huge numbers of new services imply the need for a general mechanism for providing these services into the networks. Fast and easy development, quick and cost effective deployment of these services will be essential to accelerate the business behind these services. The major hurdle in the development of these services is that traditionally all the services have been developed and hosted by the network operators themselves within their networks using proprietary technologies. A service running in one operator’s domain may not essentially be portable to another network.

1.2 Solution

Parlay & JAIN are two such industry initiatives which are technology and network independent APIs. The Parlay/JAIN API are open and technology-independent, so that the widest possible range of market players e.g. Independent Software Vendors (ISVs) may develop and offer advanced telecommunication/wireless/IP services. The API is intended to be simple to use and extensible, so that it can be applied to (and between) different types of networks and services. The Parlay/JAIN API offers a safe and secure access of third party applications into an operator’s network as it acts similar to a firewall. As the Parlay API has been adopted for the Open Service Access (OSA) specification of the Third Generation partnership project (3GPP) UMTS specifications, it is expected that Parlay will be used on a wide scale in the vast majority of 3G mobile networks. Third party application developers can use these API's to develop Telecom/ Wireless/ IP applications without having to worry about how the network actually works. Parlay and JAIN APIs will bring lots of new applications for wireless, wireline(PSTN) and IP industry. Software Agents is a new paradigm in present day computing and its convergence with Parlay/JAIN will make Telecom applications significantly more smart, autonomous and mobile.

2 Open APIs (Parlay/JAIN)

2.1 Introduction

In this section we discuss how Open APIs will play an important role in Next Generation Networks (NGN). In today's networks, applications and services are part of the network operator's domain. This network centric approach was excellent for simple mass-market applications. The Telco world can be viewed as a layered structure in three planes:

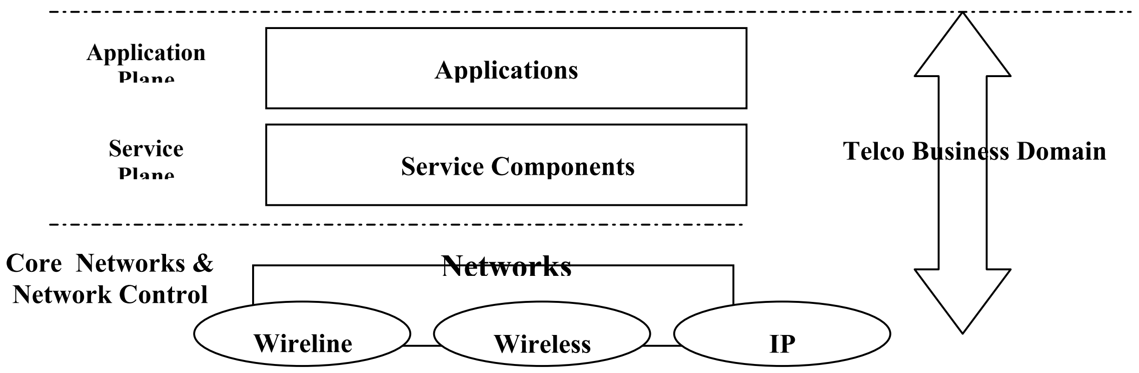


Figure 1 : Traditional Architecture

Traditionally, network services have been deployed within the network and network integrity, performance and security where the main concern of the operator. . But with technology and market advances and the emergence of mobility and IP a solution is needed that combines the benefits of the network centric approach of scalability and reliability with the creativity and power of the IT industry. This means it should be possible for the applications to be built, tested and operated by enterprises outside network domain [1]. This is achieved with a programming interface which allows applications to access the functions of the network in a secure way. A new business model has appeared in this domain. "This model involves opening up networks to third party application developers. [2] In this model the network provides services like call control, addressing, location, billing and notification. Third party applications access these network services. The biggest hurdle in this approach is to allow third party applications to access the network while protecting the network from harm. Two main industry initiatives have emerged in this field: these are the Parlay consortium and the Java APIs for Integrated Networks (JAIN). SUN has also released a JAVA technology version of Parlay 1.2 called JAIN Parlay API specification.

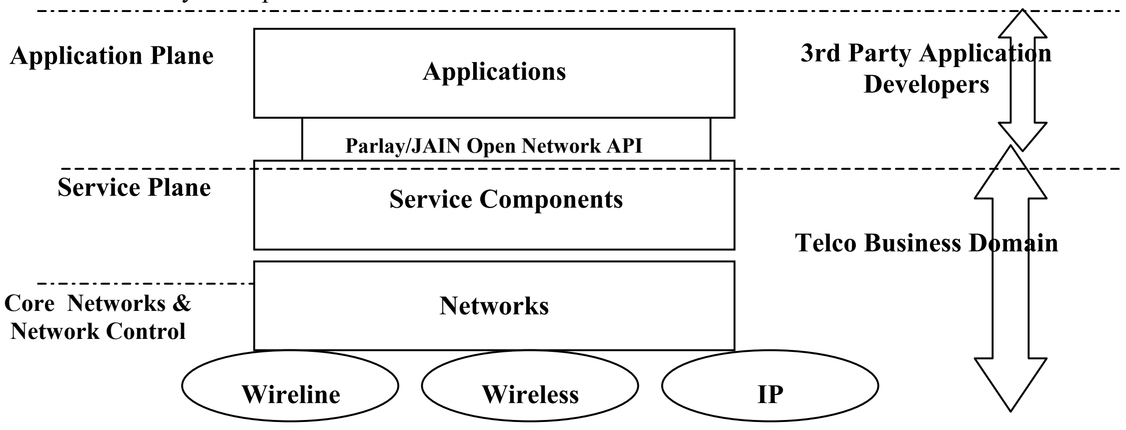


Figure 2 : New Architecture

2.2 Parlay

In March 1998 BT, Microsoft, Nortel, Siemens & Ulticom founded the Parlay Group (http://www.parlay.org). The goal of the Parlay Group is to define a set of APIs that support applications external to the secure network operator space so that the applications can have real-time control of network resources. The Parlay concept originated from an open and technology independent approach. It is designed to enable carriers, independent software vendors or third party service providers to write applications providing services across the Internet, wireless, and wireline networks [1]. The Parlay specification is a high level specification written in UML. The API defines object-oriented interfaces on both the network and client application sides in the form of network interfaces and client application callback interfaces. Callback interfaces allow the server to interact with the client.

"The Parlay API defines a set of technology-independent interfaces that specify methods, events, parameters, and their semantics to allow external (untrusted third parties) and internal (traditional network operators) application creators the control over core network resources and capabilities".[3]

Through the Third Generation Partnership Project 3GPP (http://www.3gpp.org) and OSA Application Programming Interface (API), the Parlay API has been introduced into UMTS, the next generation of mobile networks.

2.3 Architecture

From the functional point of view the Parlay Gateway consists of two main parts

- Parlay Framework.
- Parlay Network Services.

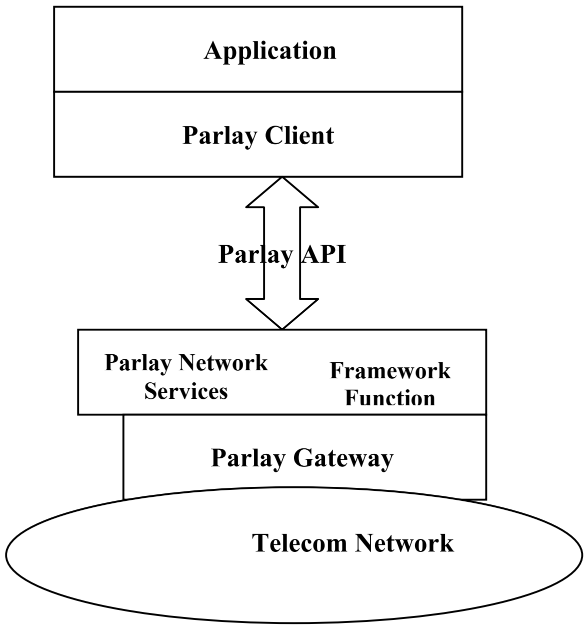


Figure 3 : Parlay Architecture

2.3.1 Parlay Framework Services

The Framework is the central part of Parlay Gateway. It provides functionality in the form of framework interfaces which are related to different management aspects. e.g. gateway access, security, network integrity and load management.

2.3.2 Parlay Network Services

The Parlay network services provide the intrinsic network related service functionality for the clients (as opposed to the Parlay framework which provides the administrative part of the Parlay functionality). The network services include[1]:

- Generic Call Control Service.
- User Interaction Service.
- Mobility management services
  - ➢ User Location
  - ➢ User Location Camel
  - ➢ User Location Emergency
  - ➢ User Status

2.4 JAIN: Java APIs for Integrated Networks

The purpose of JAIN technology is the integration of the Internet and the Intelligent Network, referred to as Integrated networks(http://java.sun.com/products/jain). JAIN brings service portability, convergence, and secure network access to telephony and Internet networks [1]. It will alter the current business structure of these networks as follows

- **Service Portability:** It is difficult to develop, maintain and market new services because of proprietary interfaces. With JAIN, proprietary interfaces are reshaped to uniform Java interfaces to develop truly portable applications.

- **Network Convergence:** JAIN allows applications and services to run on PSTN, packet (e.g. IP or ATM) and wireless networks.

- **Secure Network Access:** Applications residing outside the network can directly access network resources and devices.

The JAIN initiative takes the telecommunications/Internet market from many proprietary closed systems to single open environment able to host a large variety of services. Java and JAIN will allow carriers to extend the services and make them more feature rich. The JAIN specification is divided between two groups:

- **Protocol Expert Group(PEG),** defining JAVA APIs to TCAP, OA&M, ISUP, MAP, SIP, MGCP, H.323, and INAP protocols.
- **Application Expert Group (AEG),** defining APIs to Call Control (JAIN Call Control) and Service Logic Execution Environment (JAIN SLEE) and the JAIN Parlay, access to secure networks[4].

2.5 JAIN Parlay API

The JAIN Parlay API specification is a JAVA technology version of the Parlay 1.2 specification as defined by the Parlay Working Group. Parlay itself defines a technology independent API in the form of Unified Modeling Language(UML) and the distribution technologies CORBA and DCOM. The application developer is free to build programs using JAVA, C++, Visual C++ and Visual Basic. The aim of JAIN Parlay is to define a Java language version of the Parlay API using the best principles of Java technology while maintaining the integrity of the original Parlay API[1]. One of the benefits of defining a JAIN version of Parlay is to exploit the benefits of Java technology such as service portability, JVM reliability, ease of programmability and component framework such as Java Beans and Enterprise Java Beans. By using Java, the job of building a Parlay Gateway can be made easier through the use and reuse of other JAIN components such as JCC, JSLEE, and PEG protocols[5].

2.5.1 Architecture of the JAIN Parlay API specification

The JAIN Parlay API like its parent Parlay specification is object-oriented and consists of two categories of interfaces:

- Service Interfaces: These offer applications access to a range of network capabilities.

[1] Other services like Multiparty Call Control, DataSession & Terminal Capabilities are also available.

- Framework Interfaces: These provide ‘surround’ capabilities necessary for the Service Interfaces to be open, secure, resilient and manageable.

The services are assumed to be co-located with the framework. They are also considered trusted, such that once an application has been authorised to use particular services by the framework, then no further authorisation is required when those services are used. In some scenarios, however, additional authentication may be required for the application to use certain service features.

3 Software Agents

3.1 Introduction

Software agents comprise a new area of research and is being embedded in modern computing systems. The term “agent” is difficult to define, software *agents* are programs that assist people and act on their behalf and function by allowing people to delegate work to them [6]. Agents are often described as entities with attributes considered useful in a particular domain. This is the case with *intelligent agents*, where agents are seen as entities that emulate mental processes or simulate rational behavior; *personal assistant agents*, where agents are entities that help users perform a task; *mobile agents*, where entities that are able to roam networking environments to fulfill their goals; *information agents*, where agents filter and coherently organize unrelated and scattered data; and *autonomous agents*, where agents are able to accomplish unsupervised actions[7]. According to one of the most widely accepted definitions, given by Wooldridge and Jennings in one of their influencing journal papers, *an agent is a self-contained problem solving system capable of autonomous reactive, pro-active, social-behavior.*
Agents function by allowing people to delegate work to them and assist them by acting on their behalf. An Agent is

i) Autonomous - acts without human intervention,
ii) social - collaborates with other agents via structured messages,
iii) reactive – responds to environmental changes,
iv) proactive – acts to achieve goals.

3.2 Parlay/JAIN applications using Software Agent capabilities

The following is a set of potential applications combining the use of software agent capabilities and Parlay/JAIN:

- **Shopping**
  A commercial transaction may require real-time access to remote resources, such as stock quotes and perhaps even negotiation between involved parties. The user can give the shopping agent in its wireless device all the specification (product information, price range) [8]. This agent moves to Agent Server or WWW and there it negotiates with other agents to get the best deal for its creator.

- **Personal Assistant**
  Mobile agents ability to execute on remote hosts makes them suitable as assistants performing tasks in the network on behalf of their creators. Since Software Agents can operate independently of their limited network connectivity, their creators can even turn off their wireless devices.

- **Location specific services**
  Applications specific to User Location can be developed using Parlay/JAIN **User Location Camel Services** and Software Agents can help the application to be much more autonomous (user friendly) and get information which is real time and relevant.

- **Information and Data Seekers**
  A user is looking for some information. He/she delegates that task to the client agent in the mobile device. The client agent moves to the agent server where it meets other agents capable for performing the required task and thus delegates the task of searching to them. The user can switch off their mobile device and the next time he/she starts the device the agent would be waiting with the required information.

- **Personal Assistant**
  The use of software agents is limited not only to mobile agents; Agents can also act as 24*7 Technical support assistants ex. chatter bots. We can have a chatterbot like ALICE (http://alicebot.org) on the server and user chatting (voice) with it explaining his/her problem and the intelligent agent (programmed in Artificial Intelligence Markup Language "AIML") further delegating that task to other software agents and thus helping with most appropriate solution.

- **Monitoring and Notification**
  An agent can monitor a given information source and it can be dispatched to wait for certain kinds of information to become available. Information such as news, weather information, stock quotes can be sent to wireless users as SMS.

4 Implementation

The second part of this paper is the implementation of a Travel Guide Service which was developed at SIEMENS AG, ICM, Berlin, Germany.

4.1 Introduction

The SIEMENS Parlay/OSA Gateway (**http://www.siemens.com/parlay**)is a Service Capability Server which enables Client Applications to make use of the core network's functionality through open standard interface. The Gateway's main purpose is to provide interfaces towards external Client Applications which facilitate the access to Service capabilities of the core networks. The Parlay/OSA APIs enables network operators, service providers, and the general software developers to integrate telecommunications/wireless capability into IT software, taking advantage of information that is private to the end user. The Travel Guide Service is one such example of the above scenario which demonstrates the usage of the Parlay/OSA by directly connecting to SIEMENS Parlay Gateway.

4.2 Travel Guide Service

The Travel Guide service is intended for all travellers who are interested in location-dependent information about various topics like traffic jams, cultural events, restaurants, etc. Each user can activate the guide by sending an SMS to the server thus registering for the service along the way. The service will then continuously check the user's location. Whenever there is information for the current Cell ID stored in the Info database it is sent to the user via SMS. The user can also get information regarding current news, weather, stock prices etc. The Travel Guide is based on the **Parlay/OSA Framework, User Location CAMEL, User Interaction & Content Based Charging services.** The Travel Guide Application was demonstrated at Parlay Group Open Member Meeting, Munich, Germany (September 12 – 14, 2001). http://www.parlay.org

4.2.1 Typical Scenario

A user is starting his journey from Berlin to Munich. Before starting the journey he activates the Travel Guide Service by sending SMS "1" to the Service Phone number. When the Client Application receives SMS it checks the user's location at regular intervals and when the Cell ID changes it checks if there is some information in the Database for that particular Cell ID. If it finds some event info or other information it sends it to the user as a SMS. Before sending the information it checks if the user is a Member or a Subscriber. If the user is a member he is not charged for the service but if he is subscriber he is charged automatically from SIEMENS Pay@once server using content based charging service. The user can deactivate the service anytime by sending SMS "0" to the Service phone number. If the user needs weather information for any particular city he can request the information by sending the following SMS (W <name of city>) for ex. 'W Berlin'. Similarly for stock information the user needs to send SMS (S <Stock ticker symbol of company>) for ex. 'S SI' (SI : SIEMENS).

5 Architecture

The Travel Guide Service demonstrates the basic architecture for future Parlay/OSA services.
The Architecture consists of following modules:

- **Client Application**:
  - Travel Info Server
  - Agent Framework
  - Administrator

- **Middleware**
  - SIEMENS Parlay@vantage
  - SIEMENS Pay@Once

5.1 Agent Framework

The agent framework is basically a 3 tier Agent Architecture. We have used Open Agent Architecture (OAA http://www.ai.sri.com/~oaa/) for building the agent framework[9].

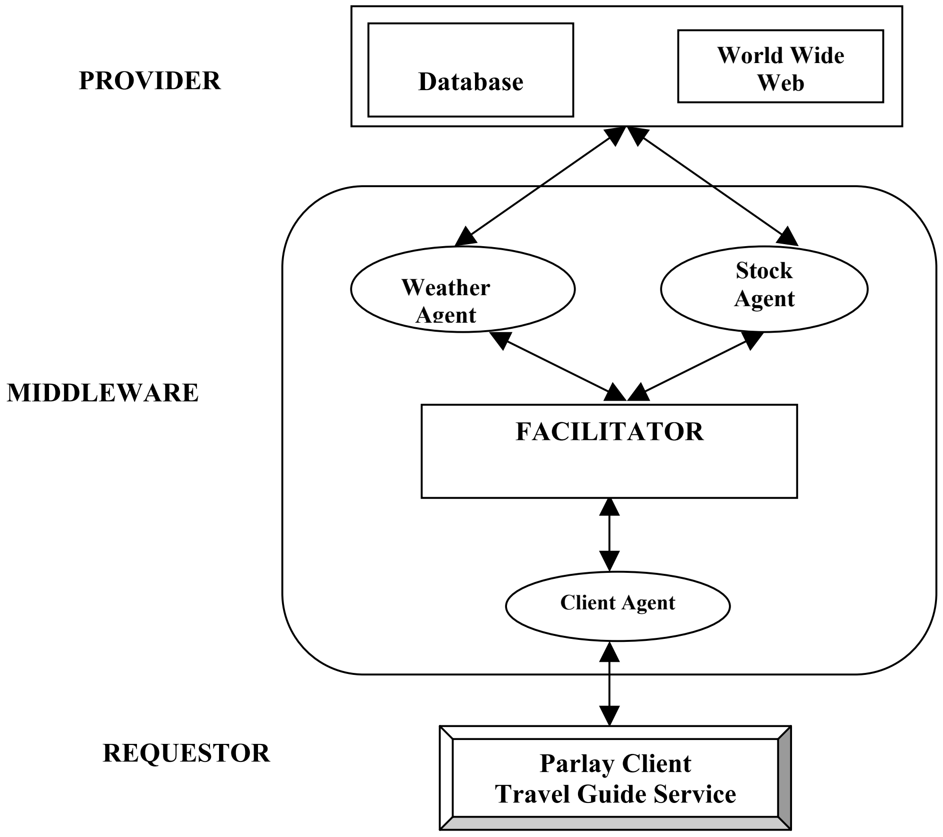


Figure 4 : Agent Framework

5.1.1 Three-Tier Agent Architecture

- The Requestor (of data or information): The information seeker, Parlay Client (Travel Guide Service) in this case. Here, the software agents task is to find out exactly what the user is looking for, what they want and if they have any preferences with regard to the information needed.

- The Provider (of data or information). This consists of individual data sources like WWW or dedicated information servers. Here, an agent's task is to make an exact inventory of (the kinds of) services and information that are being offered by its provider and to keep track of newly added information.

- Middleware. This comprises the agent architecture, i.e. here agents mediate between agents (of the other two layers), i.e. act as (information) intermediaries between (human or electronic) Requestors and Providers.

5.1.2 Advantages of the Three-Tiered Agent Architecture

- Each of the three layers only has to concern itself with only doing what it is best at.
- The Architecture does not enforce a specific type of software or hardware. This means that developers are is free to chose the underlying technique they prefer to use (such as agent architecture or programming language) to create an agent. In this implementation we have used OAA (Open agent Architecture) for our Middleware.
- It is easy to create new information structures or to modify existing ones without endangering the flexible nature of the whole system.

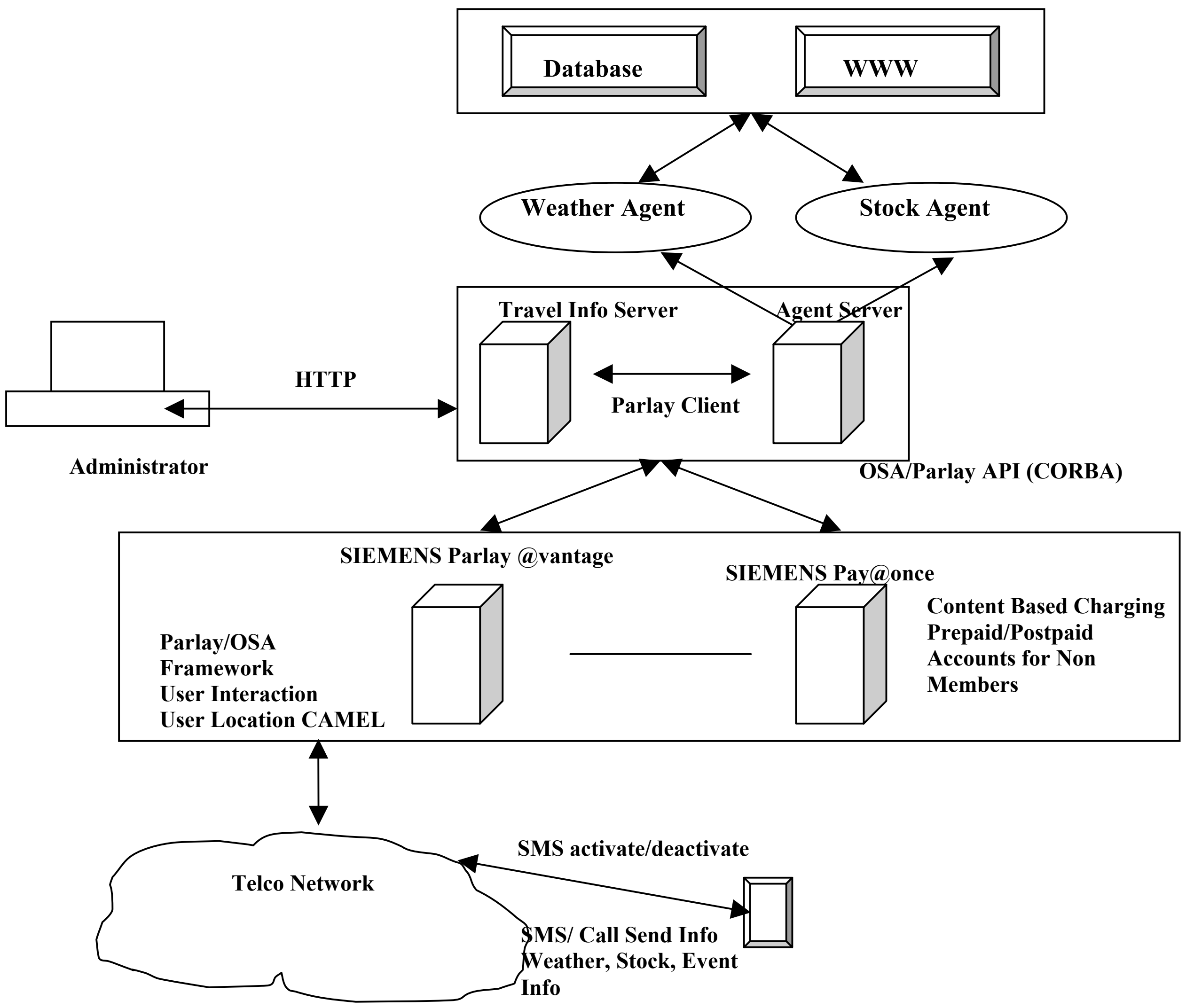
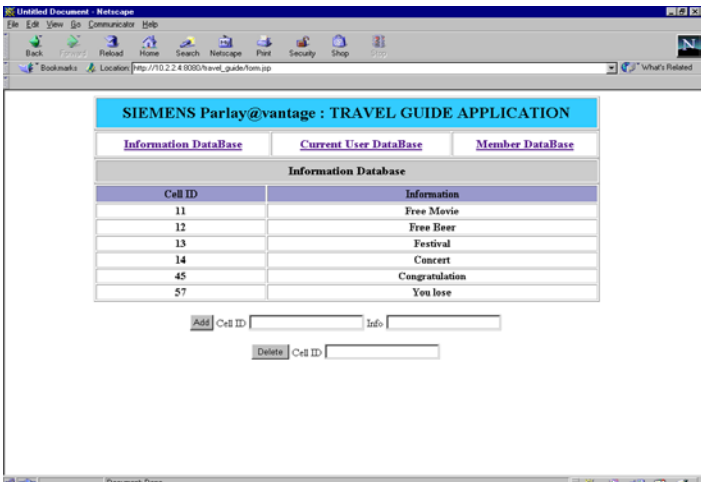


Figure 5 : Implementation Architecture

5.2 Administrator

This is a remote administration tool for administering the Travel Guide Service. It can be accessed using a Web Browser and the administrator can directly perform the following functions.

- Add/Delete Cell ID and information in **Information Database**.
- Observe the status of Users in **Current User Database**.
- Add/Delete Members in **Member Database.**


Untitled Document - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://10.2.2.4:8080/travel_guide/form.jsp
What's Related

SIEMENS Parlay@vantage : TRAVEL GUIDE APPLICATION

Information DataBase | Current User DataBase | Member DataBase

Information Database

Cell ID	Information
11	Free Movie
12	Free Beer
13	Festival
14	Concert
45	Congratulation
57	You lose

Add Cell ID Info

Delete Cell ID

Figure 6 : Administrator Web Interface

5.3 Travel Info Server

5.3.1 Introduction

Travel Info Server acts as a server to the user and as a client to the Parlay Gateway. It uses the Parlay/OSA Framework, User Location CAMEL, User Interaction & Content Based Charging services.

Parlay/OSA Framework

It provides the Parlay/OSA functionality for Trust and Security, Integrity Management, and Service Discovery and manages the Client access through Framework – Client interface.

User Interaction

The User Interaction (UI) Service provides the mechanism, which enables a client application to send information and to receive information from the end users (SMS in Travel Guide Service).

User Location Camel

The User Location Camel (ULC) service provides an interface via which a client can request and obtain network – related location information on fixed, mobile and IP based telephony users.

6 Conclusion

In this paper we presented an architecture which combines the capabilities of Open APIs(Parlay/ JAIN) and Software Agents to develop new and exciting M-Services. The current deficiency of this architecture is the lack of generally agreed upon standards, such as one for the agent communication language. Such standards are a major issue for the three-layer model, as they ensure that agents in individual layers can easily interface with agents in the other ones. Another problem is that many of the current agent systems are not compatible with other systems. This would lead to a situation where an Internet service would have to possess software for every possible type of agent that may be using the service which is an undesirable situation. Lastly if services available on the Web are to be used in mobile services then we need to look at other formats other than HTML. If we just want to display text, there's nothing wrong with HTML, but for automated Web processing, enriching documents in a way that enables computer programs (like Web robots) to do something with them, another solution, such as XML is necessary.

References

1. Anjum, F., et al., *Java in Telecommunications, Solutions for Next Generation Networks.* Communications Networking & Distributed Systems, ed. T.C. Jespen. 2001: John Wiley & Sons Ltd.
2. Bakker, L., *Rapid Development and Delivery of Converged Services Using APIs.* Lucent Technologies, Bell Labs Technical Journal, July - September 2000.
3. Beddus, S., G. Bruce, and S. Davis. *Opening up Networks with JAIN Parlay*. in *IEEE Communications Magazine*. April 2000.
4. Bhat, R. and G. R. *JAIN Protocol APIs*. in *IEEE Communications Magazine*. January 2000.
5. Tait, D., J.d. Keijzer, and R. Goedman. *JAIN: A New Approach to Services in Communication Networks*. in *IEEE Communications Magazine*. January 2000.
6. Lange, D. and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets.* 1998: Addison-Wesley.
7. Wooldridge, M. and N.R. Jennings. *Pifalls of Agent-Oriented Development*. in *Autonomous Agents*. 1998. Minneapolis/St. Paul, MN USA: ACM.
8. Maes, P., R.H. Guttmann, and A.G. Moukas, *Agents That buy and Sell.* Communications of the ACM, 1999. **42**(3).
9. Cheyer, A. and D. Martin, *The Open Agent Architecture.* Journal of Autonomous Agents and Multi-Agent Systems, 2001. **4**: p. 143-148.