Mobile Agents: Application to services in mobile computing and architectural improvements

Vasileios A. Baousis*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
bbaous@di.uoa.gr

Abstract. Mobile and more generally distributed computing are two basic research areas of Information Technology since they find many applications in our daily life, mainly due to the rapidly increasing growth of technologies of mobile networks and the Internet. The penetration of these technologies in our contemporary life has by far changed the way we work, communicate, seek information and organize our lives. In the present thesis are presented subjects that concern these two research areas. Main characteristic of this thesis is the adoption of Mobile Agents (MA) technology and their combination with other technologies and methodologies (such as the Semantic Web, the Case Based Reasoning methodology and same principles from Game Theory) into the introduced frameworks, so that lend to better results concerning the already used technologies. This thesis focuses in two basic axes, a. Modeling, description and comparative evaluation of application frameworks that allow, with transparent means, service selection and provision to mobile users, in mobile networks with the use of MA and b. Modeling, description and evaluation of a framework for resource management in MA networks.

Keywords: Mobile computing, Mobile Networks, Semantic Web, Distributed systems, Resource Management.

1 Introduction

Nowadays the ability to access services and obtain information anywhere and anytime, irrespective of the network and terminal is imperative to meet the requirements of contemporary users. From the viewpoint of network operators and service providers, meeting these requirements is a challenging issue, since many access types and service technologies should be seamlessly combined in order to deliver advanced services to the end users. Moreover, network operators and service providers, should adapt to the new emerging technologies in a reasonable time in order to preserve their customers/users and be competitive in the market. One of the service provisioning aspects in UMTS is the capability to provide to users services with the same look and feel, same interface capabilities, taking into account user's service specific prefer-

* Dissertation Advisor: Lazaros Merakos, Professor

ences as if they were in their home network irrespective of the access network and device capabilities. This option has been addressed in the 3rd Generation Partnership Project (3GPP) as part of the service aspects in UMTS and is generally referred to as the Virtual Home Environment (VHE) [1].

Furthermore, efficient execution of wireless applications is of paramount importance due to the highly dynamic wireless network conditions. Link outages occur in a near-stochastic pattern, thus, rendering the execution of applications quite tedious and uncertain. Research on mobile computing has longly focused on this specific aspect of wireless application engineering. Mobile computing is not the only development that significantly impacts the computer industry nowadays. Service-oriented architectures (SOA) are gradually changing the contemporary structure of the Internet and become a key facilitator for electronic commerce applications and related application domains. We try to incorporate both the discussed technologies in our wireless/mobile computing framework. Mobile agents are dispatched by mobile terminals in order to efficiently and safely satisfy the specific computing needs of their nomadic owner.

With the rapid proliferation of Internet technologies, distributed computing is becoming a key area in contemporary computer science. It attracts significant research attention and develops very rapidly to mature technologies applied extensively in the computing industry. Mobile Agents have been introduced as a key enabler for distributed computing. Mobile Agent (MA) technology can be used for managing, transporting and communicating functional components in a broader network infrastructure. A MA integrates application logic and has the unique ability to autonomously transport itself from one network node to another and resume execution. The MAs operate asynchronously and independently and convey the intelligence required to accomplish their task. The relocation of a MA is a critical characteristic as it allows the functional component to exploit computing resources that are available elsewhere and establish synergies with other MAs or distributed components in their current locations/nodes.

In this dissertation, the mobile agent technology is combined with other technologies and methodologies (such as the Semantic Web, the Case Based Reasoning methodology and same principles from Game Theory) into the introduced frameworks, so that lend to better results concerning the already used technologies.

This dissertation focuses on two basic axes: a. Modeling, description and comparative evaluation of application frameworks that allow, with transparent means, service selection and provision to mobile users, in mobile networks with the use of MA and b. Modeling, description and evaluation of a framework for resource management in MA networks. For the sake of brevity, here, we only present a representative contribution of the dissertation, and therefore a general description of the proposed frameworks is only discussed.

2 Service selection and provision to mobile users

In this section two frameworks are presented that offer services to mobile users. The first framework concerns the realisation of VHE using Mobile Agent (MA) technology, whereas the second framework allows the dynamic discovery and integration of semantically enriched Web Services (WS) with Mobile Agents (MA).

2.1 A Multiagent Platform for Ubiquitous Service Provision

A framework for the realisation of VHE using Mobile Agent (MA) technology is presented in this section. In the proposed architecture, MAs contain and transport service logic or data from one network entity to another in order to realize the VHE. Stationary agents encapsulate the network functionality and collaborate with MAs. Such collaboration involves the provision of network resources, execution environment and the appropriate logic. A general description of the architecture is depicted in Figure 1. The agents participating in the proposed architecture have the following functionalities and responsibilities:

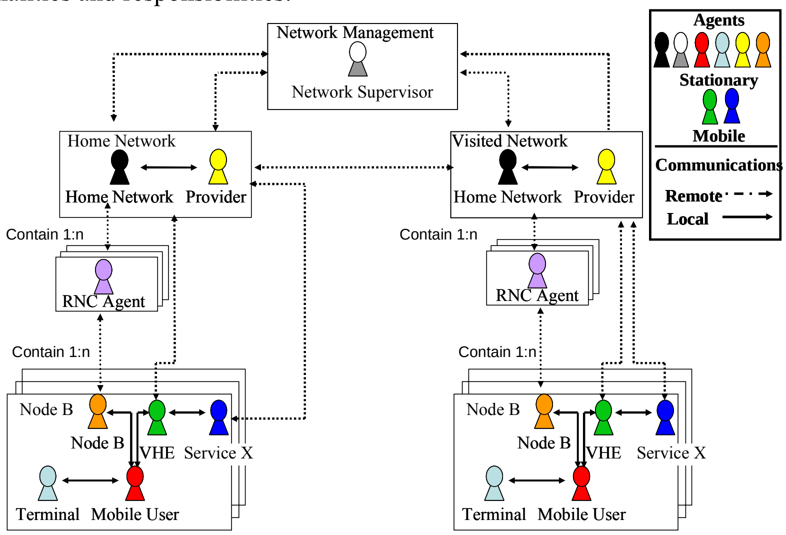


Figure 1: Mobile agent VHE architecture

<u>**Network management agents**</u>

- **Network Supervisor Agent (NSA)** is a stationary agent responsible for network interconnection and management. Associated networks have to register with NSA. Throughout this operation the NSA maintains a detailed list of the services offered by the collaborating networks. NSA also provides information on service availability. The role of the NSA could also be performed by any (properly enriched) Network Agent.
- **Network Agent (NA)** This entity is a stationary agent responsible for the network management and interconnection. Its main responsibilities include user authentication, and registration, supervision of the correct operation of RNC Agents and, finally, message routing, to and from other networks.
- **Provider agent (PA)** is a stationary agent that manages all supported services (Service Agents). The PA maintains an overview of all available services within the provider (network) domain. The PA is responsible to create the Service agents that contain the service logic of a service and dispatch them to the appropriate network entity, in order to provide a service to the requesting user. PA also creates VHE agents with user specific data (user service profile).

- **RNC Agent (RNCA)** is a stationary agent responsible for the interconnection of Node Bs and message routing from Node B Agents to Network management entities (Network and Provider agents) and vice-versa.
- **Node B Agent (NBA)** is a stationary agent responsible for Node B management and message routing for the users to the RNCA that the NBA belongs to and vice-versa.

VHE platform agents

- **Service Agent (SA)** is a mobile agent that represents an offered service. Depending on the service implementation scenario, the SA carries data, files, service logic or service results or all of them from the user's Home Network (HN) or a Service Provider to the visited network where the user currently roams.
- **Mobile User Agent (MUA)** represents the (mobile) user within the network.
- **VHE-Agent (VHEA)** is a mobile agent that is created by the PA and is dispatched to the place where the MUA is located. It contains information about user's service profile, such as personal data, preferences regarding subscribed services etc.
- **Terminal agent (TA)** is a stationary agent that allows the terminal to inform the network system about its capabilities (expressed using CC/PP).

2.2 Semantic Web Services and Mobile Agents integration for efficient Mobile Services

In this section a novel framework for dynamic discovery and integration of semantically enriched Web Services (WS) with Mobile Agents (MA) is discussed. The proposed framework is mostly intended for wireless environments where users access Semantic Web Services (SWS) in the fixed network. This framework enhances the fixed network with the intelligence needed to dispatch the service requests of the wireless user in an efficient, reliable and transparent manner. The proposed approach enables users to execute multiple services with minimum interaction, without the requirement of being online during their entire session. Additionally, the proposed framework provides better fixed network utilization since unnecessary communication overhead is avoided and reliable delivery of the service results is provided.

The proposed framework consists of the mobile user that uses SWS, the MA representing the user in the fixed network, the service registry and the SWS provider. The last two entities are implemented as stationary agents. According to the service implementation scenario (Figure 2), a mobile user accesses the proposed system and places service requests specifying some criteria. Subsequently, the system creates a MA (step 1) that migrates to the registry to find the WS that best meets the user requirements (step 2). Service registry allows for a capability search to be performed, since it is enriched with semantic information. The MA, after acquiring the WS listing and technical details, migrates to service provider(s), invokes the WS, collects the results (steps 3-5) and returns to the service requestor to deliver the results to the mobile user (step 6). In the presented scenario the SWS that matched the service request were three thus MA migrates and invokes these three services (steps 3-5). If the service request matched more than three services during the step 2, the MA would migrate to all these matched WS (Figure 2 would include more steps). The advantages of this scenario is that the MA has the necessary intelligence to invoke only the best matched service(s) and unnecessary service invocations are avoided leading to better network utilization, and the wireless user is not required to be online and may obtain the results on future time.

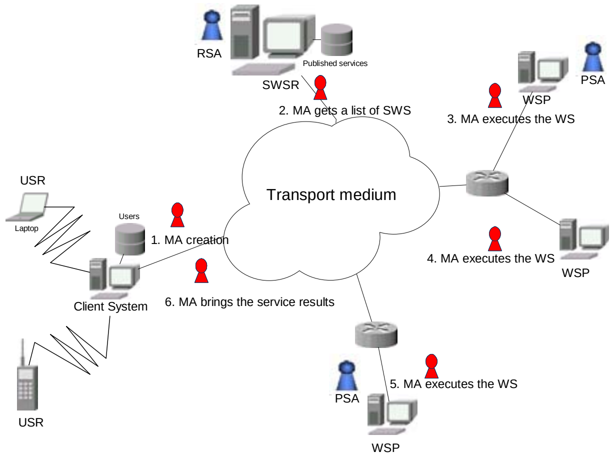


Figure 2: Service implementation scenario

In the proposed framework the route of the agent may vary, depending on the service requestor preferences and the network topology. As explained below, the user may dynamically force his MA to send its clones to the providers, invoking the services in parallel, rather than serially migrate to each one. Moreover the user may force the MA to implement different service execution strategies (e.g., execute all services locally or remotely, change timeout limit), during its itinerary and execution of service(s). The proposed framework consists of the following functional components: (1) User Service Requestor (USR) who is the user that invokes a SWS, and the Client System, the system in the fixed network that provides user access to the SWS, (2) Mobile Agent which is the representative of the user in the fixed network (3) Provider Stationary Agent (PSA) which is a stationary agent that resides in the host offering a certain WS (its implementation is optional), (4) Registry Stationary Agent (RSA) which is a stationary agent that acts as a broker between the MA and the service registry (its implementation is optional), (5) Semantic Web Services Registry (SWSR), the registry where the service providers advertise their services (6) Web Service Provider (WSP) which provides the WS to interested users.

3 Resource management in MA networks

In this work the very important issue of routing MAs within the service nodes of a broader infrastructure is investigated. Determining the right migration target node is very crucial for the efficiency of the MA infrastructure and the undertaken distributed processing task. MAs migrate whenever the following criteria jointly hold true:

- the target server has the appropriate processes or other computing resources required by the MA (qualitative constraint), and,

- the target server has adequate resources to efficiently handle the incoming process (i.e., more resources than the source node or others in the "vicinity" of the MA)

Choosing the best target server in a MA processing undertaking is a difficult task that needs to be carefully addressed. Since the MA setup involves the autonomous, concurrent operation (execution, migration) of numerous relocateable components, agent migrations need to be decided intelligently to balance the load in the broader infrastructure. Evidently, this requirement may lead to higher efficiency and increased processing throughput. In the considered architecture, there is no centralised control, thus, the risk of shifting congestions around is not negligible. Imagine N MAs ($N \geq 2$), currently found in two different nodes, who manage to locate a server node with limited load. If the relocation decision is simplistic, the two agents will arrive at the same node, hence the almost parallel migration will undermine their general processing context (e.g., agent will obtain less resources than originally planned). This is a pathological phenomenon that our server selection strategies try to avoid.

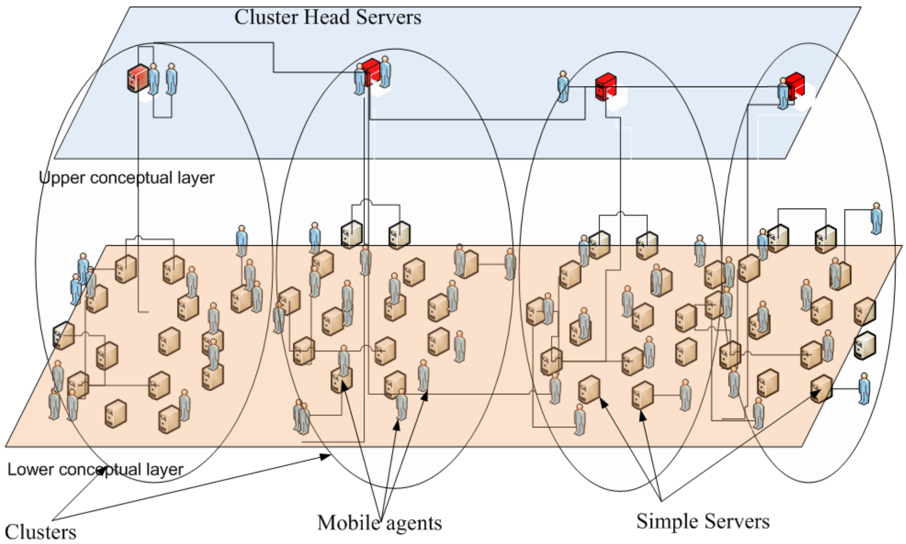


Figure 3: General Topology of the system

The general topology of the considered system is depicted in Figure 3. The system has several servers that are connected with each other in a fully meshed way and are organised in clusters. There are two conceptual layers, namely the upper and the lower layer. The upper layer includes the servers that lead the respective clusters, termed Cluster Head Servers (CHSs), and the lower layer comprises the servers that are participating to a cluster, termed as Simple Servers (SSs). A CHS has the same functionality as a SS, can provide resources to the MAs in order to perform their task, and is responsible to provide information about its cluster load to the other cluster heads. A MA is free to travel from a server (either CHS or SS) and to execute a task. A MA has the necessary intelligence to choose the best server to migrate to, in order to perform its task. The MA migration algorithm is determined by its migration strategy. In case of the SFBP algorithm each MA uses its active predictor to predict the load of the servers within the cluster that currently resides and to predict the other Cluster load as a whole. This means that the agent participate into two games (SFBP or other considered game discussed in the following paragraphs): one on the lower conceptual layer-among the SSs within its cluster and a second on the upper conceptual layer among the CHSs.

4 Experimental Results

In this section we provide a brief discussion on the results derived from the experiments conducted on each of the aforementioned frameworks. In Figure 4 we plot (using 3rd order polynomial approximation) the average of the ratios of:(a) Services found in the local cache of a Node B (noted as HITS) to the total services invoked on this Node B, (b) Total services offered successfully to users in the simulated networks, and (c) Rejected service requests to the users. The notation MA refers to a framework implemented with MA whereas WMA refers to a framework implemented without MA.As the simulation progresses the MA system outperforms the WMA system. Specifically, in the (b) case the MA/WMA ratio is kept almost constant when simulation time passes 600 sec while in the (a) case the same ratio is kept constant when time passes the 800 seconds. Finally, in case (c) MA's system rejected services are almost the half than the WMA's system rejected services through the whole simulation. From Figures 12 and 13 we observe that MA system offers more services than the WMA system even if both systems have approximately the same average load on NodeBs (time greater than 800sec). Moreover, as the rejected services to users increase on both systems, the total services decrease and both systems reach a stable state regard to the user service requests after the 1200sec till the end of the simulation.

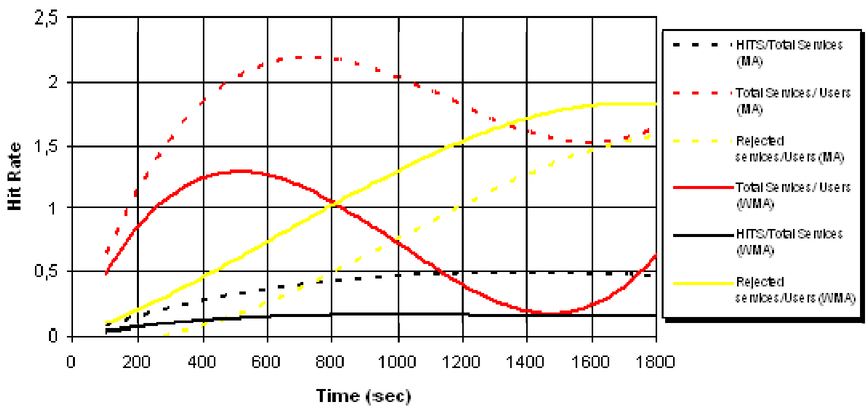


Figure 4: Service provision comparison

Following we present the results from the framework that allows the dynamic discovery and integration of semantically enriched Web Services (WS) with Mobile Agents (MA). Specifically, we compare the performance of the proposed framework against the traditional business model of WS provision. In the following description, the term “conventional WS Business Model” (WSBM), refers to the model where a user requests a service to be executed and the system dispatches (either automatically or with user intervention) the request by discovering the appropriate service(s) from the service registry, and then, sequentially, invokes these WS, receives and forwards/presents to the user the service results. All communication among the involved network entities is performed with SOAP. Moreover, in our framework the mobile agents are implemented on JADE [3] MA platform. We have developed and tested the following system: a. A WS system implemented with the “Conventional WS Business Model” (**WSBM**). b .Our framework (Semantic Web Services and Mobile Agents) (**SWS& MA**)

The SWS logic implemented in our experiments is as follows: the SWS have an extensive service description, stating unambiguously their capabilities in OWL-S. This description is published in the registry (SWSR). However, the SWS internal functionality is fairly simple, returning a pre-specified data volume subject to the service request. In our trials, these service results are 1 KB, 10 KB, 100KB and 1 MB. Moreover, six SWS have been implemented and distributed in the testing network.

In the performance evaluation scenario, a user requests a service, specifies his/her preferences and each of the above systems dispatches this request to the service registry. The service registry in the WSBM is a simple local UDDI providing a keyword service search on each service request, whereas in the SWS&MA system the registry is offering a service capability search to the placed service requests. In our evaluation the description of SWS had small differences in the OWL-S descriptions. As a result, in the WSBM system, the service search to the UDDI registry had an average of three matches per service search/request. Contrary to WSBM system, in the SWS&MA system the MA had the necessary intelligence and knowledge to filter the results from the semantic registry and invoke only a SWS where its semantic description matched the service request and user’s preferences. Consequently, in the WSBM system we considered the averaged time this system requires to execute a service and we multiplied that by 3 (the average service results from the registry), whereas in SWS&MA system we consider the average time that is needed to invoke only a SWS. Moreover, in the SWS&MA system, the average time need was used from all the system variations to execute a SWS. These system variations are: (a) a system that uses MA cloning, (b) a system that uses PSA, and (c) a system that uses both MA cloning and PSA.

Below, we elaborate on the metrics that we adopted to assess the performance of the two systems. In Equation (1), Total Service Time (TSTMA) (for the SWS&MA platform) is the sum of Registry Interaction Time (RIT), Migration of MA to a Service Provider Time (MSPT) and the Interaction Time with this Service Provider (ITSP):

$$TST_{MA} = RIT + MSPT + ITSP \tag{1}$$

In the WSBM system, Equation (1) has the form:

$$TST_{\mathrm{WSBM}} = RIT + \left\lfloor \frac{N}{2} \right\rfloor * \overline{ITSP} \tag{2}$$

where the $\overline{ITSP}$ is defined as:

$$\overline{ITSP} = N^{-1} * \sum_{i=1}^{N} ITSP_i \qquad \textbf{(3)}$$

In (2) ITSPi is the time between service request submission and service results reception.

In Figure 5, the results of the proposed system performance evaluation and comparison against a system implemented using the Conventional WS Business Model are presented. More specifically, the averaged time needed to execute 3 services for the WSBM, is plotted against the time required to invoke only one SWS in the SWS&MA for each service result size (1 KB, 10 KB, 100KB and 1 MB). We observe that the TST in the SWS&MA system is approximately half the TST in the WSBM system, irrespective of the service results size. It should be noted that the RIT in our system is considerably greater than the WSBM system, and this explains that the TST of the SWS&MA is half and not the one third (or even smaller) of the TST of the WSBM system. The high RIT of the proposed SWS&MA framework is attributed to the specific semantic registry implementation and might be less if other semantic registry is used (e.g., TUB OWLSM [4], OWLS-MX [5])

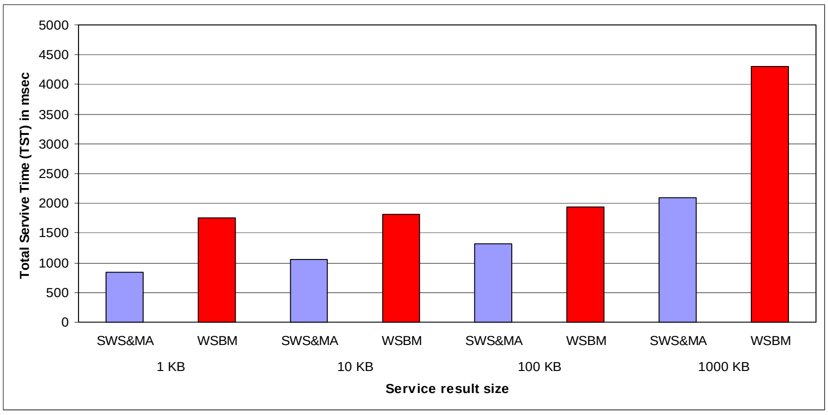
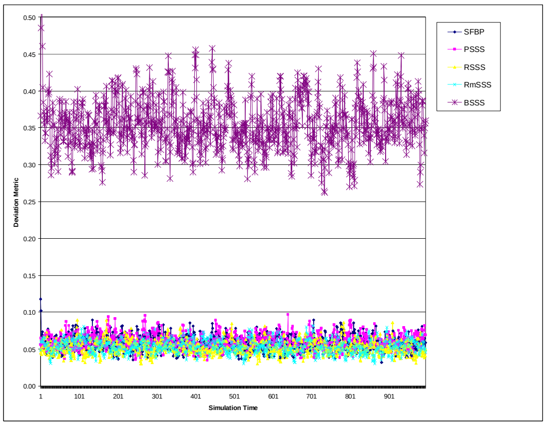


Figure 5: Total Service Time (TST) vs. Service result size

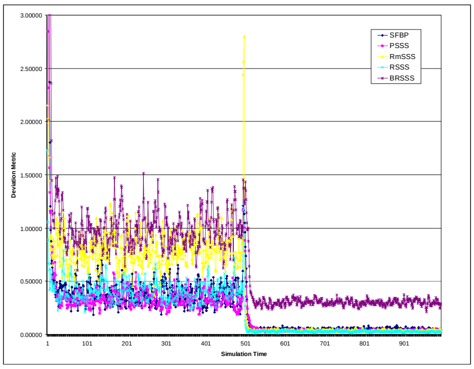
Finally the performance evaluation of the framework that performs resource management in MA network is presented. The experiments are based in the comparison of 5 different strategies that MA consider to migrate to a server in order to executed their assigned tasks. These systems are : a. The SFBP b. The BSSS (Best Response Server Selection System) which selects the lowest loaded server to migrate to c Two Random System, namely RSSS and RmSSS and d. PSSS (Probabilistic Server Selection System) where MA adopt a probabilistic algorithm to select the next server. Each system comprises five clusters, and therefore has five CHS. Each cluster encompasses 10 SSs, thus each system configuration comprises fifty five servers. The simulation lasts 1000 time slots. We consider three different simulation scenarios: a. All considered systems have 1000 MAs that perform their tasks during the simulation period and are all initially evenly distributed among the existing servers. b.All systems are initialized with 100 MAs that are uniformly distributed only to 5 servers (out of the 55) and on simulation instance 500 another 1000 new agents appear on each system (still uniformly assigned to the same 5 servers) and c. All systems are initialized with 1000 MA uniformly distributed to all available servers, and between simulation instances 500 and 700 the MAs in systems PSSS, RSSS, RmSSS and SFBP adopt the migration strategy of the BSSS.

To assess the proposed system, we consider a metric that quantifies its load balancing capacity. The metric Dev (deviation metric), is defined as follows:

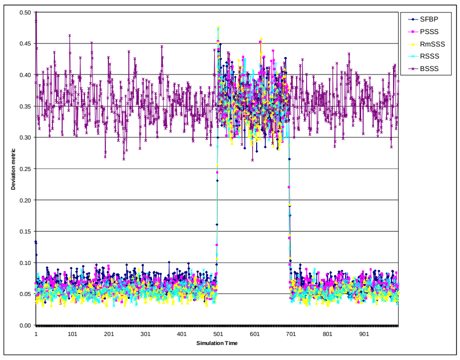
$$\mathrm{Dev} = \frac{1}{T}\sum_{t-1}^{T} \mathrm{dev}(t) \qquad \textbf{(4)}$$



Scenario 1



Scenario 2



Scenario 3

Figure 6 : Deviation metric on 3 experimental scenarios

As shown in Figure 6 all systems, except BSSS, exhibit approximately the same behaviour w.r.t. the deviation metric. Moreover, we observe that the average deviation of the BSSS is approximately 7 times greater than that of the other systems. The average deviation metric for the BSSS is 0.35 whereas for the four other systems the metric assumes a value of 0.05. Furthermore, we observe that the oscillation of the BSSS system is greater than that of the other systems and ranges from 0.30 to 0.45 whereas for the four other systems it ranges from 0.04 to 0.08. It should be noted that the SFBP system has a slightly worst performance at the beginning of the simulation and a slightly better performance at the end of the simulation period.

On second simulation scenario, we observe that all systems in simulation time 1 and 500 present high deviation values but rapidly all systems, with the exception of the BSSS, exhibit a balanced behaviour. In the first phase of the simulation, when all systems have only 100 MAs, we notice that the behaviour of all systems presents great oscillations w.r.t. the deviation metric. This is attributed to the fact that the 100 agents produce uneven load to the 55 available servers since some MAs may migrate to the same server and produce relatively high load whereas other servers have one or none MA and thus have zero or near zero load. On the second phase of the simulation, (simulation time 500), all systems have 1100 MAs and exhibit a balanced behaviour since the aforementioned oscillations cease to exist (with the exception of the BSSS).

Finally on the third simulation scenario, we observe that all systems between simulation instance 1 and 500 and between instance 702 and 1000 exhibit similar behaviour as. Moreover, in this figure it is evident the fast transition of all systems from their normal behaviour to that shown when they adopt the migration strategy of the BSSS (simulation instance 500). The same occurs when all systems re-adopt their typical migration strategy (simulation time 700).

5 Conclusions and Future Work

In this dissertation, we have introduced a set of frameworks that provide services to wireless users and a framework that allows the resource management in MA networks. The main characteristic of this thesis is the adoption of Mobile Agents (MA) technology and their combination with other technologies and methodologies into the introduced frameworks, so that lend to better results concerning the already used technologies.

On the first part of this thesis we supported the idea that VHE is realizable with the use of mobile agent technology undertaking all the required communication processes in the application level, independently of the underlying network. The proposed architecture provides a versatile framework for service provision in telecommunication networks. The use of mobile agents allows for service portability, a very important advantage in heterogeneous infrastructures similar to the ones encountered in the contemporary telecommunications landscape. In the presented system, service logic is encapsulated in mobile agents thus offering high efficiency, improved security and versatility. A contributing factor to the efficient is the caching of services in the entities of the visited network. Such caching increases efficiency, network utilization and service provisioning capacity. Besides all these aspects, our architecture inherits the obvious benefits stemming from the adoption of Mobile agents (e.g., robustness, autonomous operation). Our experiments clearly indicated the feasibility of the proposed architecture and its superiority with respect to conventional service implementation/invocation paradigms. Qualitative aspects like platform neutrality were also indicated through the presented performance assessment efforts.

On the second section we presented a framework that provides wireless access to WS using MA to find and execute WS in the fixed segment. The WS are semantically enriched and are expressed in OWL-S. Furthermore, the proposed system adopts an enhanced WS registry enriched with semantic information that provides semantic matching between service requests submitted and the service description published to them. The advantages of the presented system are: (1) users may invoke a set of services with only one interaction with the fixed network (post the request and receive the results), (2) users do not have to be connected during service discovery and invocation; the results of such operations are downloaded to their mobile devices after their network session re-establishment, (3) service invocations are performed locally or according to the user's specified policies, and unnecessary information is not transmitted over the network leading to better resource utilization, (4) the framework ensures the delivery of the service results to the user, (5) the MA dynamic behaviour improves system robustness and fault tolerance, (6) new services, agents, users and service registries can be easily integrated to the framework, thus, providing an expandable, open system. Future work includes the study of agent mobility for SWS dynamic invocation and composition that takes network events into account. Network events (e.g., node failures, overloading) occurring while the service invocation is underway, may force the MA to dynamically reschedule its itinerary. The MA will implement routing algorithms that generate itineraries by considering network information published in the WS description, network status and topology.

Finally, on the last part of this thesis, a de-centralised control scheme for managing MA migrations was introduced. MAs are updated on the saturation level of each server (potential migration target) and use their routing algorithms to independently decide on where to migrate to execute their assigned tasks. A key idea is to strengthen the load balancing properties of server selection techniques to avoid pathological situations like congestions or load oscillations. Five different migration strategies were studied and thoroughly assessed through simulations. Our findings showed that the proposed architecture provides an efficient network monitoring framework where several MA routing algorithms could be applied. Two of the considered MA migration strategies, namely SFBP and PSSS, proved very efficient. As a result the use of resources throughout the network was optimized and rationalized. Future work includes the use of more network monitoring parameters such as: server Input Output usage, available server memory that will be considered in the MA migration. Each MA will determine the next server to migrate to, by considering its task requirements, expressed as a function of the resources being monitored.

References

1. 3GPP Technical Specification 22.121 v5.3.1: "The Virtual Home Environment (Release 5)", (2002).
2. Tsetsos V., Anagnostopoulos C., and Hadjiefthymiades S.: Semantic Web Service Discovery: Methods, Algorithms and Tools, chapter in: Semantic Web Services: Theory, Tools and Applications (Ed. Dr. Jorge Cardoso), IDEA Group Publishing (2007).
3. JADE: Java Agent Development Environment. http://jade.tilab.com. (2007)
4. OWLSM : The TUB OWL-S Matcher. http://kbs.cs.tu-berlin.de/ivs/Projekte/owlsmatcher/index.html (2007)
5. OWLS-MX: Hybrid OWL-S Web Service Matchmaker. http://www.dfki.de/~klusch/owls-mx/ (2007)