Dynamic Replication and Deployment of Services in Mobile Environments

Gabriel Guerrero-Contreras, Carlos Rodríguez-Domínguez,
Sara Balderas-Díaz, and José Luis Garrido

University of Granada
Software Engineering Department, E.T.S.I.I.T.
C/ Periodista Daniel Saucedo Aranda s/n, Granada, Spain
{gjguerrero,carlosrodriguez,jgarrido}@ugr.es
sarabd@correo.ugr.es

Abstract. Currently, mobile environments are gaining importance, and new promising paradigms, like *Mobile Cloud*, are arising. However, these environments pose new challenges and are mainly characterized, among others, by frequent changes in their execution context. This particularly is challenging for software architects in the design and implement this kind of systems. For instance, the Service Oriented Architecture (SOA) paradigm has not been devised to operate in dynamic network environments, where centralized services or static deployments must be avoided, in order to provide a higher availability of services. Therefore, in order to provide a reliable SOA implementation in mobile environments, it must be complemented with techniques and methods of Autonomic Computing. In this work, a self-adaptive SOA approach is proposed. This architecture will provide support to the dynamic replication and deployment of services in mobile environments. The approach is based on the context information modelling and management as key aspect to achieve the required architecture adaptation.

Keywords: Context-awareness, context modelling, software architecture, mobile environments, Service Oriented Architecture (SOA).

1 Introduction

The Service Oriented Architecture (SOA) paradigm proposes a modular distribution of the functionalities of a system through services [10]. It provides the foundations to build an interoperable and scalable system thanks to the use of standard protocols (e.g., SOAP) and service composition (i.e., service orchestration and choreography). However, SOA paradigm has not been designed to operate in dynamic network environments [2], such as mobile environments. Thanks to the increase of the capabilities of mobile devices, mobile systems are gaining importance and new concepts and paradigms are arising, like, for instance *Mobile Cloud* [5]. These environments pose new challenges and are characterized, among others, by frequent changes in their execution context (e.g., changes

Á. Rocha et al. (eds.), *New Contributions in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 353, DOI: 10.1007/978-3-319-16486-1_85

in network topology or in the availability of the devices). Because of this, centralized services or static deployments must be avoided, in order to provide a higher availability of services. Thus, in these environments it is not feasible to implement a system strictly based on SOA.

For instance, in the forensic domain there are different scenarios: natural disasters, accidents, terrorist attacks, murders, etc., where security forces to apply protocols of action intended to support victim identification. Forensic support systems, like the *Mobile Forensic Workspace* [12], must allow to exchange information with nearby devices, in order to support data sharing between forensic experts. However, in some scenarios common network infrastructures may not be available. Consequently, an ad-hoc network must be created between the users that compose the working group, who may be moving around the scene. This implies unstable connections (disconnections and network partitions), and thus, the availability of the services and information can be compromised.

Therefore, systems deployed in mobile environments should aim to be autonomous sufficiently to adapt themselves to context changes in runtime, without the explicit intervention of the user. In this way, SOA paradigm must be complemented with techniques and methods of Autonomic Computing [9] for architectural software design. Autonomic Computing provides the techniques to extend a system with self-* features (i.e., self-healing, self-configuration, self-optimization and self-protection). This is fundamental in mobile environments, where systems must be proactive, and for that reason, self-adaptive architectures have been gaining importance in the research community [17].

However, current solutions are not effective. This is primarily for two reasons: (1) they are solutions designed for a particular application domain, and hence, they are difficult to apply and reuse in different domains; and (2) they are based in a limited set of context information for the dynamic deployment of services, which leads to solutions with a poor performance.

In this paper the context information that influences the adaptation process in self-adaptive service replication and deployment is identified and modelled. This context model is defined independently of specific application domains, within the mobile collaborative networks, and it takes into consideration the network topology, the computational features of the devices, and how they adjust to the requirements of each specific service. Moreover, a self-adaptive SOA approach is proposed. This architecture is based on the proposed context model and provides support to the dynamic replication and deployment of services in mobile environments.

The rest of this paper is organized as follows. Section 2 provides an overview of the literature in this field. Section 3 defines a context model and introduces a self-adaptive software architecture to support the dynamic replication and deployment of services in mobile environments. In Section 4, the Mobile Forensic Workspace case study is detailed, and it highlight how the proposed system helps to improve the availability and reliability of the services and information deployed in the system. Finally, Section 5 draws some conclusions and outlines the plans for future research.

2 Related Work

Some research works have addressed the dynamic topology of new execution environments through a self-adaptive approach can be found in the literature. Usually, the solutions adopted are based on the following context information: device position (i.e., distance between device or travel speed), device battery and connections stability (i.e., network partitions).

The device position is usually used to create device clusters. Device clusters methods are applied to turn a distributed network in a set of interconnected local clusters that can be dealt individually like a centralized network. Dustdar et al. [4] create device clusters on the basis of the distance between mobile devices. However, Wang et al. [16] show that travel speed is a better measure to create clusters of mobile devices. The clustering also implies the election of a cluster leader. In both works, the device leader is elected by its computational features (e.g., remaining battery).

Regarding to replication triggers, in [1], service replicas will be created when too many requests are made to a service from an external group. This replica will be allocated in the leader of the cluster, i.e., the replica is allocated based on the computational features of the device. Hamdy et al. [8] propose a replication protocol based on the interest of devices in the use of the service, i.e., when an application needs to access a service, it may be deployed in the same device where is hosted the application, if there are not replicas in its neighbour devices. Moreover, network partitioning detection mechanisms also help to decide when replicating a service. For example, in [3], TORA routing protocol [11] is used in combination with an estimation of the residual link lifetime of wireless links. When a device predicts a partition, this device will host a replica of the service.

Generally, these are ad-hoc solutions developed for specific scenarios, and they are based on an implicit context model. The absence of an explicit context model involves an added difficulty to customize or reuse the proposal. Then, the impossibility of extend the model according to the particular requirements of a specific scenario makes inefficient, in that scenario, the proposed solutions.

3 A Self-adaptive Service Deployment

In this section, first the context information that influences the self-adaptive replication and deployment of services in mobile environments is identified and modelled. Then, a self-adaptive software architecture, based on this context model, is proposed.

3.1 Context Model

With regard to the problem of dynamic replication and deployment of services in mobile collaborative networks, the following context aspects have been identified as relevant to provide a solution: device features (e.g., battery, memory, processing capacity, etc.), network topology (e.g., client-service distance) and service requirements, i.e., how the device features adjust to the requirements of each specific service. The diagram in Figure 1 shows the context model proposed. This is an independent application domain model, and takes into consideration different context aspects: device features, service requirements and network topology.

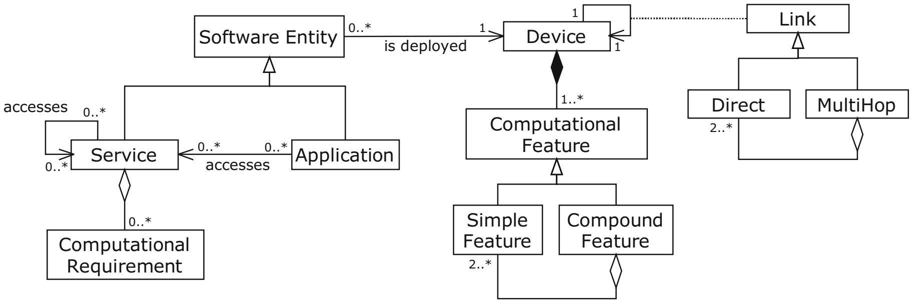

Fig. 1. A context model for self-adaptive service replication and development in mobile environments

Device Features. A mobile environment is a heterogeneous environment where different devices may offer different features. These features have been classified into *Simple Features*, such memory or storage, or *Compound Features*, such remaining battery. This is, a *Compound Feature* is a feature that is derived from two or more *Simple Features*. For example, remaining battery could be calculated from screen brightness, CPU usage, phone signal strength, and other features.
It is important to take into consideration the device features in order to choose the devices with best performance to act as service hosts, thus improving service reliability levels. Also, the current usage of these must be taken into consideration to deploy a service replica (remaining battery, CPU or memory, among others). For instance, if a device has a 85% average occupancy rate of its bandwidth, it is not advisable to deploy a service replica, owing to this device could be a bottleneck and deploy a new service replica would aggravate its situation.

Service Requirements. Although there is cross-cutting information that affects in a similar way the deployment of all services of the system, such as the device battery or client-server distance, there is other information that affects in a greater or lesser measure the deployment of services according to their requirements. For example, one service could rather need high processing power while other could need high storage capacities. For this reason the services must specify theirs *Computational Requirements* independently [6]. The service developer must indicate what features must provide the devices in which the service could be deployed, through a set of condition statements (e.g., “free storage space equal or greater than 1GB"). This would allow the system to distribute the workload among the devices of the network, providing the appropriate deployment of the services according to their requirements. Moreover the number of clients (i.e., applications and other services), dependencies between services (i.e., regarding service composition) and number of replicas of the service, are also interesting context information to provide an efficient service deployment solution.

Network Topology. This model considers an ad-hoc network where the devices themselves act as routers. Therefore, each device has two kind of *Links*: *Direct links* (one-hop) and *Multihop links*. Each *Multihop link* is made up of two or more direct links between devices. In order to simplify the management of the network topology, the model considers only the best path between two devices, but it is independent of its calculation (fewest hops, higher bandwidth, best stability, etc.). It is interesting to consider the stability of the *Link* in order to anticipate a possible network partition, and the possible loss of the communication with a service or client. Also, the devices that are part of the route between a service and its clients can be an interesting election to host a service replica, in order to reduce the response times, bandwidth utilization, and allocating the service replicas near of their clients in an uniform way within the network topology.

3.2 Self-adaptive Software Architecture

The Figure 2 depicts a general schema of a self-adaptive software architecture to support the dynamic replication and deployment of services in mobile environments. This SOA-based architecture is supported by communication middleware technologies based on Event-Driven Architecture (EDA).

Middleware technologies provide a homogeneous layer that facilitates the communication between the different entities of the network [13]. Additionally to common services in a communication middleware (publish/subscribe, dynamic discovery or routing services, among others), in this self-adaptive architecture, the middleware layer provides a *Context Manager Service* (CMS). This service is supported by different monitoring services (e.g., *Network* or *Device Monitor*), which provide information about context changes in the execution environment. CMS manages a *Knowledge Base* that stores information under the context model previously presented (Section 3.1). To obtain a complete perception of the context of the system, an intensive and costly monitoring could be necessary, due to the dynamic nature of the environment. Therefore, the information stored in the *Knowledge Base* of each device will be mostly local to its neighborhood.

The *Replica Manager Component* (RMC) is generally responsible of the maintenance of the different service replicas in the system, specifically, it is responsible of the synchronization of the operations performed on them. In the proposed architecture, it also encapsulates the adaptation logic regarding the replication and deployment of the service replicas. In order to provide a distributed solution, each service replica has a RMC. It has tree main components:

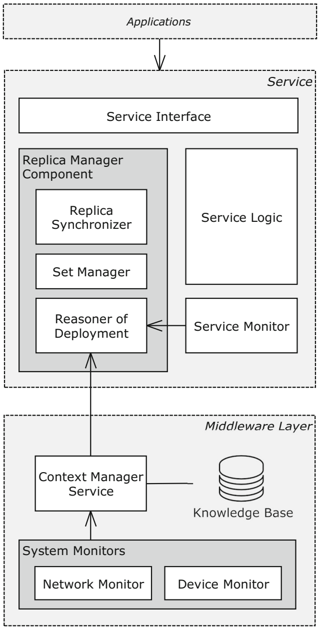

Fig. 2. Self-adaptive software architecture to support dynamic replica and deployment of services in mobile environments

- *Replica Synchronizer.* This component is responsible of the synchronization between the different replicas of a service [7]. It is strongly related to the specific nature of the service. For instance, when two service replicas, which have been isolated each other in two different partitions of a network, are joined again, probably, there will be inconsistencies between the two replicas. These inconsistencies must be resolved according to the specific nature of the resource (images, documents, annotations, etc.).
- *Set Manager.* It is responsible of the communication and coordination within the set of replicas of the service. It can adopt different replica protocols [15]: passive, such primary-copy protocol; or active, such gossip protocol. In the case of primary-copy, each *Set Manager* evaluates its execution conditions: device capabilities according to service requirements and its position within the network topology (number of direct links, distance to clients, distance to other required services, etc.), and the best ranking replica is chosen as primary copy (active replica). Also, the *Set Manager* keeps the information about the number of replicas and their location.
- *Reasoner of Deployment.* It is responsible of making decisions about when replicate a service; migrate, hibernate or delete a service replica; and where to locate the replica. It is based on the information provided by the CMS, and additionally by the information provided by the *Service Monitor.* The *Service Monitor* provides information about the state of the own service

replica, for instance, responses time or number of clients.
Its reasoning is based on a rule-based system, in order to provide adequate response times to changes in the context. Therefore, it can decide: *replicate* a service when the resources of the current host device are running out (e.g., battery), when the topology of the network changes (e.g., a partition is predicted) or when new clients appear; *migrate* a replica when the distribution of the clients changes among the network or a better device to host the replica is discovered; or *hibernate/delete* a replica when the number of clients is reduced.
Hibernation mechanism is useful in order to react faster to changes in the network topology and to reduce network traffic. However, services must be previously deployed, which in some cases may reduce the flexibility of the system.

4 Case Study

The case study mentioned at the beginning of this paper, the *Mobile Forensic Workspace* [12], is helpful to show the need and usefulness of the proposed system. In this particular case study the following assumptions can be made: (1) the devices that will be used belong to a work team, thus the necessary services can be pre-installed, allowing a hibernation mechanism; and (2) generally, the work teams in the gathering of evidences are small teams (5 or 8 members), therefore, a copy-primary replication protocol is feasible. This replication protocol is simple to manage, but it generates higher levels of network traffic as a result of the update messages for the passive replicas [15].

The Figure 3 depicts an example scenario within this case study where three member of a forensic team are gathering preliminary evidences of two victims. In the scenario two kinds of devices can be found: (1) a laptop deployed statically in a police car; and (2) three mobile devices (one for each team member). Also, different services can be found (communication, image repository, victim's information repository, etc.). In the case of Image Repository service, each forensic expert can take several pictures with his mobile device of an evidence; different forensic experts can photograph the same evidence; or the same forensic expert can photograph evidences that belong to different victims. Also, the forensic experts can add annotations to these pictures, which can be related with other pictures. Therefore, the Image Repository service must keep an ordered and consistent set of this information, and at the same time must provide high availability, to allow forensic experts to access and share pictures of the evidence found.

In this scenario the proposed context model and self-adaptive architecture can be useful to improve the availability of the Image Repository service and its data, and also the reliability of the service. This is highlighted in the following situations:

1. Initially, all forensic experts are near the victim one (*V1* in the Figure 3). All of them have connection with the laptop located in the police car. Therefore,

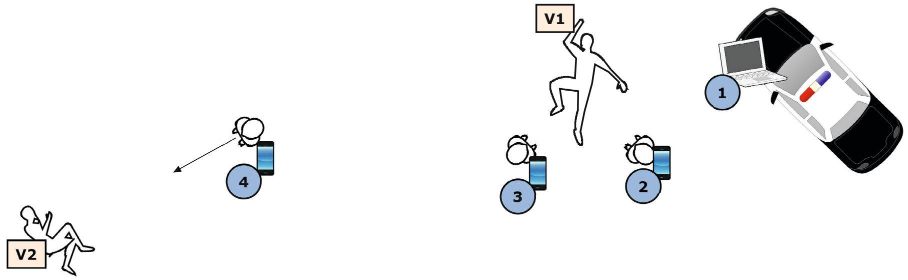

Fig. 3. Example scenario within the Mobile Forensic Workspace case study

the active service replica (copy-primary protocol) is the one deployed in the laptop (*device 1* in the Figure 3), as it has better computational features. Particularly, the Image Repository service will need storage capacity.

2. The forensic expert of the device 4 will move to the area where the victim two is located (*V2* in the Figure 3). This area is out of the laptop coverage. Then, the Image Repository service replica must be activated. This can be triggered in two ways: (1) from the active service replica located in the laptop, which is monitoring the network and predicts the disconnection of the client of the device 4 (through the link quality or the speed of travel of the device); or (2) by the *Service Monitor* of the passive replica deployed in the device 4, which detects the disconnection with the active replica. In this last case, the *Set Manager* will search other active replica. As the only replica available is itself, it will turn to active. In this point, the network is partitioned into two groups: (1) the devices 1, 2 and 3, where the device 1 hosts the active replica; and (2) the device 4, that provides service to itself.
3. Later, the forensic experts of the devices 2 and 3 move to the *V2* area. Therefore, they become part of the second team. The battery of the device 4 is running low, therefore, the *Replica Manager* will search a better place to the service. Following a distributed approach, each replica will evaluate its situation and the best ranked, according to context model, will become in the active replica. Moreover, the active service replica deployed in the laptop (device 1 in the Figure 3) will have no clients, therefore the *Replica Manager* will hibernate it in order to save resources.

5 Conclusions and Future Work

In this work, the context information model that influences the adaptation process in self-adaptive replication and deployment of services has been defined. The proposed context model takes into consideration the network topology, the computational features of the devices, and the specific computational requirements of a service. This allows the system to distribute the workload between the devices of the network, providing the appropriate deployment of the services according to their requirements. Also, within the mobile collaborative networks, the context model is independently of specific application domains, which makes it reusable.

This context model is used to support the decision-making system of a self-adaptive SOA approach. The proposed architecture provides the basis to support the dynamic replication and deployment of services in mobile environments and hence to provide a reliable SOA-base system for dynamic network environments. Moreover, the Mobile Forensic Workspace, a real world scenario, has been described. This scenario reflects the need and usefulness of the proposed self-adaptive software architecture.

At present time, a study of the proposed system in a network simulator is under development. This study will allow to investigate the performance of the system regarding to different configuration parameters (such as monitoring intervals, local/broad knowledge, replication strategies, etc.), and to empirically compare the behaviour and the performance to other approaches. As future work, the self-adaptive architecture may be combined with self-adaptive capabilities at other architectural levels, for example, in the internal functioning of the services [14], in order to improve their performance and efficiency.

Acknowledgment. This research has been partially supported by the Spanish Ministry of Economy and Competitiveness with European Regional Development Funds (FEDER) under the research project TIN2012-38600; and the scholarship program FPU, granted by the Spanish Ministry of Education, Culture and Sports.

References

1. Ahmed, A., Yasumoto, K., Shibata, N., Kitani, T.: Hdar: Highly distributed adaptive service replication for manets. IEICE Transactions on Information and Systems 94(1), 91–103 (2011)
2. Choudhury, P., Sarkar, A., Debnath, N.C.: Deployment of service oriented architecture in manet: A research roadmap. In: 2011 9th IEEE International Conference on Industrial Informatics (INDIN), pp. 666–670. IEEE (2011)
3. Derhab, A., Badache, N., Bouabdallah, A.: A partition prediction algorithm for service replication in mobile ad hoc networks. In: Second Annual Conference on Wireless On-demand Network Systems and Services, WONS 2005, pp. 236–245. IEEE (2005)
4. Dustdar, S., Juszczyk, L.: Dynamic replication and synchronization of web services for high availability in mobile ad-hoc networks. Service Oriented Computing and Applications 1(1), 19–33 (2007)
5. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: A survey. Future Generation Comp. Syst. 29(1), 84–106 (2013), `http://dx.doi.org/10.1016/j.future.2012.05.023`

6. Guerrero-Contreras, G., Garrido, J.L., Akhlaki, K.B., Balderas-Díaz, S., Rodríguez-Domínguez, C.: Towards a self-adaptive deployable service architecture for the consistent resource management in ubiquitous environments. In: Augusto, J.C., Zhang, T. (eds.) Workshop Proceedings of the 10th International Conference on Intelligent Environments. Ambient Intelligence and Smart Environments, vol. 18, pp. 206–217. IOS Press (2014), `http://dx.doi.org/10.3233/978-1-61499-411-4-206`
7. Guerrero-Contreras, G., Garrido, J.L., Rodríguez-Domínguez, C., Noguera, M., Benghazi, K.: Designing a service platform for sharing internet resources in MANETs. In: Canal, C., Villari, M. (eds.) ESOCC 2013. CCIS, vol. 393, pp. 331–345. Springer, Heidelberg (2013), `http://dx.doi.org/10.1007/978-3-642-45364-9_27`
8. Hamdy, M., König-Ries, B.: A service distribution protocol for mobile ad hoc networks. In: Proceedings of the 5th International Conference on Pervasive Services, pp. 141–146. ACM (2008)
9. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Computer 36(1), 41–50 (2003), `http://doi.ieeecomputersociety.org/10.1109/MC.2003.1160055`
10. Papazoglou, M.P., van den Heuvel, W.: Service oriented architectures: approaches, technologies and research issues. VLDB J. 16(3), 389–415 (2007), `http://dx.doi.org/10.1007/s00778-007-0044-3`
11. Park, V.D., Corson, M.S.: A highly adaptive distributed routing algorithm for mobile wireless networks. In: Proceedings of the IEEE INFOCOM 1997. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution, vol. 3, pp. 1405–1413. IEEE (1997)
12. Rodríguez-Domínguez, C., Benghazi, K., Garrido, J.L., Garach, A.V.: Designing a communication platform for ubiquitous systems: The case study of a mobile forensic workspace. In: New Trends in Interaction, Virtual Reality and Modeling, pp. 97–111. Springer (2013)
13. Rodríguez-Domínguez, C., Benghazi, K., Noguera, M., Garrido, J.L., Rodríguez, M.L., Ruiz-López, T.: A communication model to integrate the request-response and the publish-subscribe paradigms into ubiquitous systems. Sensors 12(6), 7648–7668 (2012)
14. Ruiz-López, T., Rodríguez-Domínguez, C., Ochoa, S., Garrido, J.L.: Mdubi: A model-driven approach to the development of self-adaptive services for ubiquitous systems. Sensors, 1–25 (2014)
15. Tanenbaum, A., van Steen, M.: Distributed systems: principles and paradigms. Pearson Prentice Hall, Pearson Education International, Upper Saddle River, NJ (2007)
16. Wang, K.H., Li, B.: Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks. In: Proceedings of the IEEE INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 1089–1098. IEEE (2002)
17. Weyns, D., Ahmad, T.: Claims and evidence for architecture-based self-adaptation: A systematic literature review. In: Drira, K. (ed.) ECSA 2013. LNCS, vol. 7957, pp. 249–265. Springer, Heidelberg (2013), `http://dx.doi.org/10.1007/978-3-642-39031-9_22`