60

SUPPORTING MOBILE VIRTUAL TEAM'S COORDINATION WITH SOA-BASED ACTIVE ENTITIES

Ruben Dario Franco[1], Andrés Neyem[2],
Sergio Ochoa[3] and Rosa Navarro[4]
[1,4] *CIGIP Research Centre, Polytechnical University of Valencia, SPAIN*
[2, 3] *Computer Science Department, University of Chile, CHILE*
dfranco@cigip.upv.es, aneyem|sochoa@dcc.uchile.cl, ronava@cigip.upv.es

The use of IT solutions to support preparedness, response and recovery process has been envisioned as a possible way to improve the support for collaboration among the actors participating in mobile teams operating under constrained scenarios. Software applications supporting mobility have typically been conceived as functional extensions of centralized enterprise systems hosted on corporate servers. However, in this work, we focus on those mobile scenarios which go one step further in terms of interoperability requirements. This proposal aims to support the design, deployment and execution of ad-hoc collaborative business processes (workflows) in mobile networks, operating under constrained conditions and taking advantage of the Active Entity (AE) concept.

1 MOTIVATION

Information and communication technologies (ICT) allow workers to labor outside the office and accomplish their activities while they are on the move. That is, organizations are evolving towards environments where full mobility is a requirement. For instance, one of the most ignored, but urgent and vital challenges confronting society today is the vulnerability of urban areas to "eXtreme" Events (XEs) [7, 1, 3]. These XEs include natural disasters such as earthquakes, hurricanes and floods, as well as accidental and intentional disasters such as fires and terrorist attacks.

One important lesson learned from recent disasters is the need to improve collaboration among organizations or actors involved in those disaster relief efforts [10, 2, 15]. Many pitfalls related to collaboration, such as lack of trust, information sharing, communication and coordination, have been well documented [9, 12, 7, 13].

The use of IT solutions to support preparedness, response and recovery process has been envisioned as a possible way to improve the support for collaboration among the actors participating, for instance, in disaster relief efforts [8]. This work proposes to develop an IT-based platform that allows improving the co-ordination of involved parties.

This paper presents an extension to the traditional workflow models in order to support dynamic settings. Specifically, the scenarios include mobile devices interacting among themselves using a Mobile Ad-hoc NETwork (MANET).

In this paper, we focus on those mobile scenarios which go one step further in terms of interoperability requirements. This proposal aims to support the design, deployment and execution of ad-hoc collaborative business processes (workflows) in mobile networks, operating under constrained conditions and taking advantage of the Active Entity (AE) concept.

In order to describe the strategy to support mobile workflows, this paper introduces a lightweight architecture addressing the challenges involved in such scenarios. Section 2 describes the challenges and opportunities offered by service-oriented computing to support workflows-based collaborations in ad-hoc wireless settings. Section 3 defines the structure of an Active Entity and explains the architecture for executing active entity services. Section 4 presents the conclusions and future work.

2 SUPPORTING DISTRIBUTED PROCESSES IN MOBILE ENVIRONMENTS

Software applications supporting mobility have typically been conceived as functional extensions of centralized enterprise systems hosted on corporate servers. The main design goal of these applications has been to provide a mobile environment within which mobile workers may gather information to be synchronized against a server at a later stage. This issue jeopardizes the collaboration required by mobile workers to execute a workflow process [2, 11].

On the other hand, traditional workflow models offer powerful representations of teamwork activities. These models enable efficient specifications and executions of a business process [2]. The execution of such workflows is designed to work on high-end servers connected by reliable wired connections. Under this approach, the workflow definition and control are kept in a centralized structure. This situation constrains the workflow-based collaboration, particularly in ad-hoc mobile environments, where disconnections occur frequently.

Ad-hoc networking refers to a network with no fixed infrastructure [16]. When the nodes are assumed to be capable of moving, either on their own or carried by their users, these networks are called MANETs. The nodes of the network rely on wireless communication to collaborate with each other [15].

On the other hand, Service-Oriented Computing (SOC) is a new paradigm gaining popularity in distributed computing environments due to its emphasis on highly specialized, modular and platform agnostic code facilitating interoperability of systems [14].

As a summary, high degree of freedom and a fully decentralized architecture can be obtained in MANETs at the expense of facing significant new challenges. MANETs are opportunistically formed structures that change in response to the movement of physically mobile hosts running potentially mobile code. Advances in wireless technologies allow devices to freely join and leave work sessions and networks, and exchange data and services at will, without the need of any infrastructure setup and system administration. Frequent disconnections inherent in ad-hoc networks produce data inconsistencies in centralized service directories. Architectures based on centralized lookup directories are no longer suitable. Thus, the model and technologies addressing these issues should consider all nodes as mobile units able to provide and consume services from other mobile units [8].

3 WFMS ARCHITECTURE FOR MANETS

3.1 Active Entities

The architecture supports mobile workflows by enabling distributed workflow execution and taking advantage of the concept of Active Entity (AE), which is a building block used to design abstract models of collaborative business processes. AEs are designed as abstract definitions for each kind of role involved in the process.

For example, in a construction scenario for the electrical inspection process two main roles can be identified: the project manager and the inspector. Both of them need to provide a specific service interface to interoperate to report the results of the designed process. An abstract definition of each role (AE) will include those services belonging to them and, at design time, they will be orchestrated to form a collaborative workflow process.

At execution time, specific instances of each AE are created. Consequently, abstract definitions are instantiated on specific mobile devices being handled by the project manager and electrical engineers acting as inspectors.

In order to simplify process modeling and execution phases, a key architectural decision was made. By wrapping the activities that are part of a workflow process, under services interfaces, it is possible to provide the building blocks needed to compose and execute extended and complex business processes. Although this is an achievement in terms of interoperability and flexibility, the Web service application space needs to be designed and be more useful than just a set of input and output messages. In order to deal with this issue, Yang proposed the service components be organized as an architectural pattern based on a high level self-contained composite service [17]. This architecture presents a public interface and includes a private part comprised of the composition constructs and logics that are required for its manifestation. The public interface definition, provided by a service component, describes its messages and operations. The service component messages and operations can be published and then searched, discovered, and used like any normal web service. The encapsulated composition logic and construction scripts are private (internal and invisible) to a service component.

Following Yang's ideas, AEs can be used as web-based components of business resources (roles), able to be involved in process activities. These AEs can be implemented by an abstract class of web service methods. They make up a mechanism for creating composite services via reuse, specialization, and extension.

3.2 Distributed Processes with Active Entities

A distributed business process may be conceived as a set of activities which are assigned to various roles to be accomplished [6]. An execution unit is a complete piece of work that can be assigned to some entity of the process which has the proper knowledge and capability required for its accomplishment. Thus, Active Entities can be redefined as follows:

Active Entities are service providers which by means of their public service interfaces, are able to accomplish some execution unit, by providing and consuming third-parties (others AE) services. Those services are modeled at design time and, *then, they are encapsulated in an abstract class that will be instantiated at each resource of the problem's domain.*

From the business process perspective, AE can be used at modeling time to describe patterns of service compositions. Then, each process instance is executed by means of their invocations. Service interfaces are known by them because they are embedded in the entity modeling process. Figure 1 shows the AE structure.

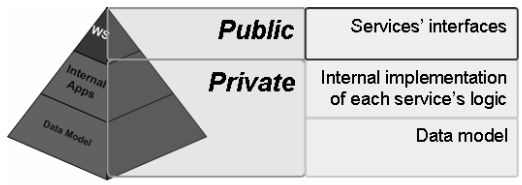

Figure 1. Active entity structure

AEs may represent companies, organizational units, resources or mobile workers. On top of this message-oriented bus (see Figure 2), other applications may also gain access to those services in order to compose value-added functionality. Therefore, the execution model is based on the orchestration of execution unit accomplishments; the execution units are those previously assigned to AEs.

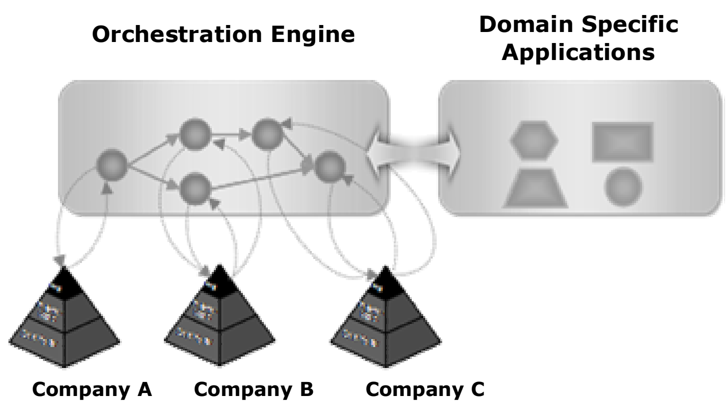

Figure 2. Organizations and organizational resources modeled as active entities

At this level, a business process model can be described as a set of peer-to-peer service invocations among AEs. These relationships are described by means of a Service Level Agreement (SLA) among them (contracts). This model establishes a directed graph of peer-to-peer information exchanges, representing the process flow or path of the task (Figure 3).

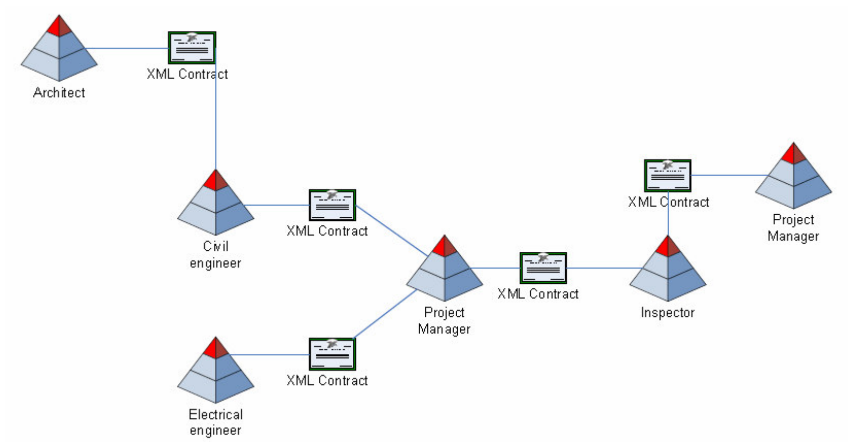

Figure 3. A process can be designed using active service entities and contracts

3.3 The Architecture

The architecture (Fig.4) consists of a set of components extending traditional server-based workflow engines to be accessible by mobile workers. It has to consider thin clients due to the hardware limitations of mobile devices.

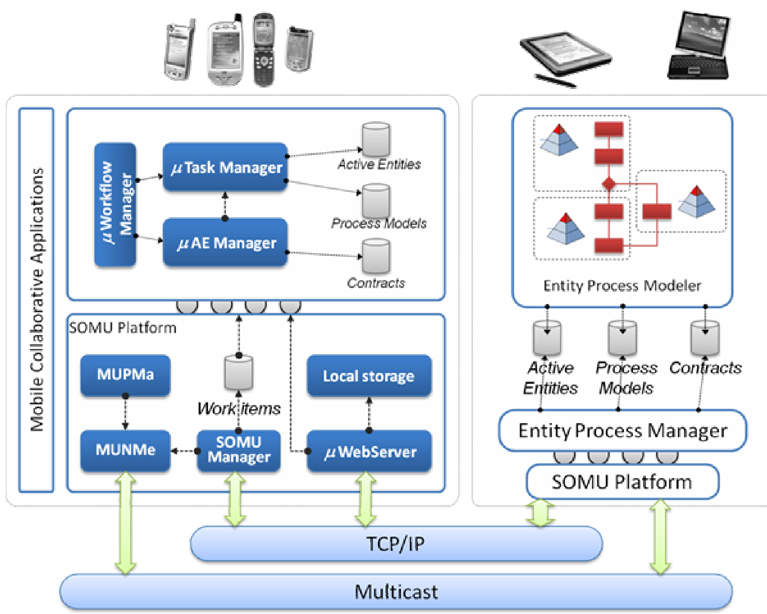

Figure 4. Architecture to support mobile workflows by using AE

- Entities Process Modeler: This component supports the process definition including modeling of activities assigned to active entities, control connectors among AEs, input/output containers and entity contracts. A process is represented as a directed acyclic graph with activities (nodes) and control/contracts connectors (edges). The graph is manipulated via a built-in, event-driven and compliant graphic editor. The entity contracts are specified as XML structures. These structures are described via XML Scheme facilities. The AEs Process Modeler distinguishes two main types of activities assigned to active entities: process activities and blocks. Blocks contain the same constructs as processes, e.g., activities or control connectors.

- - AEs Process Manager: After the planning stage, this component accepts the plan from the AEs Process Modeler and passes it on to the planner for allocation of AEs. When a process is executed, activity instances are assigned to AE to perform the actions required for the activity. This may be a manual task, such as verifying a legal clause in a contract or a computerized task, such as entering the data status of the electrical facilities at a construction into the database via the program specified for the activity. The identification of the right AE for a mobile worker is based on data running the BP. Thus, roles describe the type of activity the mobile worker performs within the active entity, such as team leader or electrical engineer. The information about the organization and people involved in the process is generally stored in databases. When the process is defined, the AEs Process Modeler defines for each activity, which AE should perform it. Assigning the activity to a role provides significant advantages over the assignment of the activity directly to a mobile worker. It can be replaced without affecting the correct execution of a process. Furthermore, during execution, this component gets regular progress updates in order to monitor active entities distributed among mobile workers.
- - µWorkflow Manager: This component stores the plan for the active entities assigned to a mobile worker. This Manager uses the µActiveEntity and µTask Manager components to handle the active entities and tasks that have been allocated. This manager needs data about the plan for allocating tasks to determine (1) the active entity from which the inputs for a task are going to come and (2) the active entity to which the results will be returned. The µWorkflow Manager can be updated by the AEs Process Manager when a new plan is entered into the system (on the group leader), or by the group leader via the SOMU Platform [7], when active entities allocations are made.
- - µActiveEntity Manager: This component handles the active entities assigned to a mobile worker. This component needs status information about the tasks allocation for determining the active entity in order to perform the required actions.
- - µTask Manager: This component handles the task state transition of an active entity. A task being executed in the disconnection mode may change its state according to the current user's situation. For example, a user may terminate or suspend his/her task according to the surrounding business situation. On the other hand, there is a need to handle predictable task state mismatch when reconnected. The degree of task state mismatch may be higher than one, and it represents the difference of state levels between two task states.
- - SOMU Platform: The Service-Oriented Mobile Unit (SOMU) is a lightweight platform running on PDAs, TabletPC and notebooks [7]. It enables each mobile computing device to produce and consume Web Services from other peers. Such functionality is implemented on a lightweight Web Server. Thus,

the autonomy and part of the interoperability required by mobile workers is supported. SOMU also implements a local storage which is composed of (1) a shared storage space to allocate the files the mobile unit needs to share, and (2) a space to allocate those Web services exposed by the mobile unit. By default, SOMU provides basic

- WS for Web services description and discovery. The SOMU main components are SOMU manager, µWebServer, Mobile Units Near Me (MUNMe) and Mobile Units Profile Manager (MUPMa). The SOMU Manager is the component in charge of creating, storing and dispatching work items when a mobile collaborative application invokes WS exposed by other mobile units. The work items stored in a mobile unit represents the WS invocations that such unit needs to perform. Each work item is composed of a ticket, a mobile universal unit, the WS proxy, WS input and WS output. The ticket is the work item identifier. It is used to inform the results of a WS invocation to a mobile collaborative application. The Mobile Universal
- Identification (MUI) names each mobile unit. The MUI allows the SOMU Manager directly invoke WS running on other mobile units. WS Proxy contains the data needed to coordinate the invocation and the response of WS exposed by other units. WS Input contains the parameters to be sent by the WS Proxy when it invokes the remote WS. WS Output contains the results of a WS invocation. The broker of all interaction between two mobile units is the µWebServer. The MUNMe is the component in charge of discovering and recording the mobile units that are close to the current device. This data is used to decide a good time to start an interaction with a specific unit. This component uses a multicast protocol. It involves discovering the name, MUI and the IP address of the units belonging to the MANET. Since WS are typically accessed from various kinds of mobile devices, interoperability and personalization play an important role for universal access. The MUPMa stores and manages information related to mobile units, such as the MUI, hardware and network capabilities. WS can use this data to provide optimized contents for various clients.

4 CONCLUSIONS AND FURTHER WORK

Workflow-based cooperative technology is promising, but current systems supporting workflow based collaboration (WfMSs) are designed for stable wired networks or nomadic mobile networks. We sought to lay a foundation for WfMSs operating in unpredictable MANET environments without depending on centralized resources or reliable communication links.

We presented an architecture that was implemented to let mobile computing devices expose and consume Web services of AE to perform an activity. This prototype was programmed in C# using the functionality provided by the .NET Compact Framework. However, it could have been implemented using J2ME instead. The type of implementation allows the prototype to run on a wide range of devices from PDAs to desktop PCs.

The prototype provides a basis for the development of mobile collaborative applications based on workflow models. It intends to increase the technical feasibility of solutions in the area and reduce the development effort of MANET-based mobile collaborative applications.

The supporting infrastructure has been used as a basis for mobile workflow that will help electrical engineers to conduct inspections in construction sites. Such application has not been formally tested.

Future work includes experimentation to study the possible contributions and limitations of the proposed strategy. Moreover, the functionality provided by Web services need to be tested to determine if the proposed uncoupled interaction represents a limitation for mobile workers collaborating with ad-hoc communication.

5 REFERENCES

1. Columbia/Wharton Roundtable. “Risk Management Strategies in an Uncertain World,” IBM Palisades Executive Conference Center. April, 2002. (www.ldeo.columbia.edu/CHRR/Roundtable, last visit: Jan-2003)
2. U. Dayal, M. Hsu and R. Ladin, “Business Process Coordination: State of the Art, Trends, and Open Issues”, Proceedings of the Twenty-seventh International Conference on Very Large Data Base, Rome, Italy, Sep. 11-14, 2001, pp. 3-13.
3. Godschalk, D. “Urban Hazard Mitigation: Creating Resilient Cities”, Natural Hazards Review. ASCE. August 2003. pp. 136-146.
4. G. Hackmann, R. Sen, M. Haitjema, G.-C. Roman and C Gill, MobiWork: Mobile Workflow for MANETs, Technical Report WUCSE-06-18, Washington University, Department of Computer Science and Engineering, St. Louis, Missouri, USA, 2006.
5. J. Jing, K. Huff, B. Hurwitz, H. Sinha, B. Robinson and M. Feblowitz, “WHAM: Supporting Mobile Workforce and Applications in Workflow Environment”, Proceedings of the Tenth International Workshop on Research Issues in Data Engineering, San Diego, California, USA, Feb. 27-28, 2000, pp. 31-38.
6. R. Khalaf and F. Leymann, “E Role-based Decomposition of Business Processes using BPEL”, Proceedings of the IEEE International Conference on Web Services, Chicago, USA, Sep. 18-22, 2006, pp. 770-780.
7. Mileti, D. Disasters by Design: A Reassessment of Natural Hazards in United States. Joseph Henry Press. Washington D.C. 1999.
8. National Governors Association (NGA). “Volume Two: Homeland Security – A Governor’s Guide To Emergency Management”. NGA Center for Best Practices. 2002.
9. Nigg, J. “Emergency Response Following the 1994 Northridge Earthquake: Intergovernmental Coordination Issues,” Disaster Research Center, University of Delaware, Newark, 1997.
10. National Research Council: Board on Natural Disasters. “Reducing Disaster Losses Through Better Information”. National Academic Press. Washington, D.C. 1999.
11. A. Neyem, S.F. Ochoa and J.A. Pino, “Supporting Mobile Collaboration with Service-Oriented Mobile Units”, Proceedings of the Twelfth International Workshop on Groupware, Medina del Campo, Spain, Sep. 17-21, 2006, LNCS 4154, pp. 228-245.
12. Quarantelli, E. “Major Criteria for Judging Disaster Planning and Managing and their Applicability in Developing Societies,” Background paper for the Int. Seminar on the Quality of Life and Environmental Risk. Rio de Janeiro, Brazil, Oct., 1996.
13. Stewart, T. and Bostrom, A. “Extreme Event Decision Making Workshop Report,” Center for Policy Research. Rockefeller College of Public Affairs and Policy. University of Albany, and Decision Risk and Management Science Program NSF. June, 2002. (www.albany.edu/cpr/xedm, last visit: Jan-2003).
14. R. Sen, R. Handorean, G-C. Roman and C. Gill, “Service Oriented Computing Imperatives in Ad Hoc Wireless Settings”, Service-Oriented Software System Engineering: Challenges and Practices, Stojanovic and Dahanayake (Eds.), Idea Group Publishing, Hershey, USA, 2005, 247-269.
15. I. Stojmenovic and J. Wu, “Ad-hoc Networks”, IEEE Computer, 2004, 37(2), 9-74.
16. C. Tschudin, H. Lundgren, and E. Nordström, “Embedding MANETs in the Real World”, Proceedings of the Eighth IFIP International Conference on Personal Wireless Communications, Venice, Italy, Sep. 23-25, 2003, LNCS 2775, pp. 578-589.
17. J. Yang, “Web service componentization”, Communications of the ACM, 2003, 46(10), 35-40.