VIRTUAL ORGANIZATION MANAGEMENT USING WEB SERVICE CHOREOGRAPHY AND SOFTWARE AGENTS

Adomas Svirskas[1], Ioannis Ignatiadis[2], Bob Roberts[2], Michael Wilson[3]

[1]*Department of Computer Science*
Vilnius University, LITHUANIA
adomas@svirskas.com

[2]*Centre for Applied Research in Information Systems*
Kingston University, UK
{I.Ignatiadis, R.Roberts}@kingston.ac.uk

[3]*Business and Information Technology Department*
CCLRC Rutherford Appleton Laboratory, UK
m.d.wilson@rl.ac.uk

The purpose of this paper is to discuss an approach for the automated management of Virtual Organizations using Web Service Choreography and Software Agents. Web Service Choreography is important for the support and management of commonly agreed collaboration scenarios. In order for the scenarios to be understood and accepted by members of a VO, the scenarios need to be specified in a standard, machine-readable form and mapped to the implementation mechanisms of the VO participants. Software agents can then serve as the background support mechanism for the automation and management of the phases of identification, formation, operation and termination of a VO.

1. INTRODUCTION

A Virtual Organization (VO) is a temporary or permanent network of geographically dispersed entities (individuals, groups, organizational units or entire organizations). Virtual Organizations tend to be viewed as a radical approach to management, or a strategic approach that leads to dynamically reconfigurable enterprises (Sieber and Griese, 1999; Saabeel et al., 2002). The key enabler for Virtual Organizations is electronic business, with firms joining together to form short-term partnerships and/or long-term strategic alliances. Those partnerships are formed in response to more intensive competition, shorter product life-cycles, more specialized markets and faster technological change (Davidow and Malone, 1992; Hagel and Armstrong, 1997; Aldrich, 1999). In Virtual Organizations traditional external boundaries of organizations start to blur. However, the enablement of better exchange and sharing of information through inter-organizational systems can lead to greater efficiency, flexibility, and ability to respond to market requirements (Bovet and Martha, 2000; Timmers, 1999).

Virtual Organizations can be implemented using Service-Oriented Architecture (SOA) solutions, and Web Services in particular. Integration of those services requires standardized choreographies, which are definitions of the "conversations" between cooperating applications that allow them to work together correctly (Vinoski, 2001). Choreography is important to ensure that the sequence of states, operations and conditions that manage and control the interactions of the relevant services is carried out correctly (Web-Services-Architecture, 2004). In addition, enterprises involved in Virtual Organizations may utilize agent technology to simplify management and control of their operations, both internal and external. Agent technologies use principles of distributed decision-making, parallel and distributed computing, component-based software engineering, autonomous computing, and advanced methods of interoperability and software integration (Marik and McFarlane, 2005). Software agents can aid the formation of business alliances, planning short or long-term cooperation deals, and managing (including reconfiguration and dissolving) cooperation. This paper will therefore discuss the importance of Web Service Choreography and the use Agent technology for the operation and management of Virtual Organizations.

The rest of the paper is structured as follows: Section 2 presents the background on Web Service Choreography, and in particular Web-Service Choreography Description Language (WS-CDL). Section 3 presents the background on Multi-Agent Systems (MAS) and discusses interrelationship between the Web Service Choreographies and Agent Technologies. Section 4 discusses the importance of Web Service Choreographies and Agent Technologies for the management and operation of Virtual Organizations, while section 5 presents the conclusions of this paper.

2. WEB SERVICE CHOREOGRAPHY

Before discussing Web Service Choreography in more detail, it is useful to distinguish between choreography and orchestration, as there is sometimes confusion between the two. Orchestration specifies the behavior of a participant in choreography. This is achieved by defining a set of "active" rules whose execution decides what to do next. Once the rule is computed, the corresponding activities are then executed. Orchestration assumes the existence of a central point of control, which governs the overall workflow of activities, which effectively means the composition of a new service from existing services. Choreography, on the other hand, is meant to be enacted at runtime by peers in a Virtual Organization, without an intermediary. The choreography definition can be used to verify that everything is proceeding according to the plan (Ross-Talbot, 2005). Choreographies are defined in declarative description format, and are enacted by the collaboration participants at run-time.

Of course, it is important for collaborating parties to use the same language for choreography (business protocol) description. One of the service choreography description standardisation initiatives is W3C WS Choreography Working Group. It coordinates creation of the WS-CDL language (WS-CDL, 2005), which is the means to define a technical multi-party contract, primarily in Web Services domain. WS-CDL specification is aimed at being able to precisely describe collaborations between any types of participants regardless of the supporting platform or programming model used by the implementation of the hosting environment, thus addressing heterogeneity issues (WS-CDL, 2005). Choreographies must also completely hide component-level implementation details. Moreover, the same choreography definition (potentially involving any number of parties or processes) needs to be usable by different parties operating in different contexts (industry, locale, etc) with different software (e.g. application software) (WS-CDL, 2005).

Choreography definition using WS-CDL allows building of more robust services because they can be validated statically and at runtime against a choreography description, verification of absence of deadlocks and live-locks, etc. It also helps to ensure effective interoperability of services, which is guaranteed because services will have to conform to a common behavioural multi-party contract, mentioned earlier (Ross-Talbot and Brown, 2005).

There is a difference, between executable languages such as Java, C#, BPEL on one hand, and declarative description languages such as WS-CDL on the other hand. The latter capture a global view of messaging activity and are not designed to provide information about how participants implement their individual tasks. Therefore, there is a need for generating role-specific code skeletons from choreography description in order to facilitate faster and more convenient implementation of individual functionality. The choreography description language uses roles to differentiate between the participants in choreographies; this can be used efficiently for end-point code generating.

There are several ways to implement choreography support functionality, which involves both local and global tasks and relies on different types of information to perform them. For example, when a choreography description is first published or modified, it needs to be distributed to all the participants; the latter will negotiate the terms and quality of service, etc – this is global information. On the other hand, when analyzing an incoming message, sent by a peer in a choreographed exchange, the choreography support service relies on some local information – e.g. business rules, mapping definitions etc. The choreography support service, therefore, needs to interact intelligently with the corresponding services of the peers and those of the supporting and/or managing entities, for example reputation, policy decision, VO management services. One of the available implementation options is the Multi-Agent Systems (MAS) approach, which is discussed in the next section.

3. MULTI-AGENT SYSTEMS AND RELATIONSHIP WITH WEB SERVICE CHOREOGRAPHIES

Agent technologies are suitable mainly for domains of highly complex problems and systems with widely distributed information sources, domains with dynamically changing environment and problem specification, and for the integration of a high number of heterogeneous software systems (Jennings and Bussmann, 2003). Therefore, the agent technologies are suitable for usage in a Virtual Organization, where the participants are geographically distributed, usually with heterogeneous software systems, and where the environment is dynamically changing in response to market needs and requirements.

(Software) agents are autonomous, which is very desirable in unknown scenarios (which usually tend to appear in the real world), where it is difficult to control directly the behaviour of complex business collaborations. Even though it is possible to encapsulate some behaviour by specifying "private" methods, agents must decide by themselves whether to execute their methods according to their goals (agents must be proactive), preferences and beliefs. Agents are also flexible, they have to learn from, and adapt to, their environment. This is important, since when designing an agent system, it is impossible to foresee all the potential situations that a particular agent might encounter, and specify agent behaviour optimally in advance. This kind of situation is highly probable in the most of non-trivial VO interactions.

In a VO setting, a multi-agent system can be employed for supporting internal processes (intra-enterprise level) of the enterprise (e.g. planning and control, resource allocation, production process simulation - (Pechoucek et al., 2002)), and on the other hand, it can support cooperation and negotiation among enterprises (extra-enterprise level) across a value chain (e.g. customers, suppliers, material and service providers, etc). Both types of agents can coexist in an organization.

In addition to the external-internal dimension of agents' classification, there is another one, which is based on the *specific purpose* of the agent services. The reason for making this distinction explicit is the fact that the business services themselves can be considered as agents as they satisfy most of the agents' characteristics. (Maximilien and Singh, 2005a) in their work of cataloguing Web services interaction styles argue that viewing services as agents enables us to augment the interaction styles of Web services as interactions between and among service provider agents and service consumer agents. Therefore, it is important to denote the areas of responsibility of the business services and the supporting agents.

As we have explained above, Web services are characterized not only by an interface but also by the business protocols (choreographies) they follow. While business protocols are application specific, much of the software required to support such protocols can be implemented as generic infrastructure components (Alonso et al., 2004). For example, the infrastructure can a) maintain the state of the conversation between a client and a service, b) associate messages to the appropriate conversation, or c) verify that a message exchange occurs in accordance to the rules defined by the protocols (for example WS-CDL). Part of the task of the infrastructure is also the execution of meta-protocols, which are protocols whose purpose is to facilitate and coordinate the execution of business protocols. It is convenient to think of the agents as the meta-protocol enablers, paving the way for the business services.

For example, before the actual interaction can begin, clients and services need to agree on what protocol should be executed, who is coordinating the protocol execution, and how protocol execution identifiers are embedded into messages to denote that a certain message exchange is occurring in the context of a protocol. In the Web Services domain, WS-Coordination is a specification that tries to standardize these meta-protocols and the way WSDL and SOAP should be used for conveying information relevant to the execution of a protocol (Alonso et al., 2004). In the Multi-Agent System (MAS) domain, there are other protocols for agents' interaction, which can be useful for implementing the meta-protocols.

Having distinguished between the agents and the services, we need to make sure that these two types of entities coexist peacefully within a single architecture and interoperate properly. (Maximilien and Singh, 2005b) propose a framework that augments a typical Service-Oriented Architecture (SOA) with agents. Their principal idea is to install software agents between service consumers and each service that they consume. These consumer service agents expose the same interface as the service. However, they augment the service interface with agent-specific methods. The consumer communicates its needs via the augmented agent interface. Service method invocations are done via the service agent who, in turn, monitors and forwards all calls to the selected service. Both business and meta protocols can be modelled, validated and verified using the WS-CDL language and tools.

A good example of a consistent set of meta-protocols is the VO Management domain, where the business collaboration partners (peers) interact by the rules agreed by all the VO members and VO managers. These rules are enacted partly by direct interaction between the peers, and partly by the peers and the VO Management. We discuss Virtual Organization Management in the next section.

The concept of multiple agents can also be useful in general-purpose Web service composition. (Maamar et al., 2005) present an agent-based and context-oriented approach that supports the composition of Web services. To reduce the complexity featuring the composition of Web services two concepts are put forward in their work, namely, software agent and context. During the composition process, software agents engage in conversations with their peers to agree on the Web services that participate in this process. Conversations between agents take into account the execution context of the Web services. The security of the computing resources on which the Web services are executed constitutes another core component of the agent-based and context-oriented approach presented by (Maamar et al., 2005).

4. MANAGEMENT OF VIRTUAL ORGANIZATIONS

VOs follow a life cycle of four phases: (A) Identification (Opportunity Identification and Selection), (B) Formation (Partner Identification and Selection, and Partnership Formation), (C) Operation (Design, Marketing, Financial Management, Manufacturing, Distribution), and (D) Termination (Operation Termination and Asset Dispersal) (Strader et al., 1998). The management of a VO through those phases can be described just as a type of business process (or collaboration) that uses the same mechanisms as for "operational" business processes (or collaborations). The collaboration agreement of a VO then specifies the processes that are related to the administration of the VO itself, for example changes to the VO membership (Svirskas et al., 2005).

In some cases the management of a VO may take the form of a process carried out by an entity centrally located, so it might resemble an orchestration-type of management. This may be typical in vertically organized industries, e.g. aerospace, where a large vendor centrally controls a supply chain VO. However, in most cases a VO is amongst peers, those being either a VO member, or an enabling service (such as management, monitoring, security, etc). The interaction is usually binary, however in some cases (e.g. negotiation and need for consensus), the interaction may involve more than two parties (Svirskas et al., 2006). In addition, within a VO its members may belong to different organizations with different application and workflow implementation technologies, therefore resisting external control of their back-end applications (Papazoglou and Dubray, 2004). Also, the number of VO management collaborations (e.g. membership management, trust provisioning, monitoring) is limited. These considerations suggest that management of a VO centrally is not a suitable option, but that VO management interactions should be choreographed. The VO management protocol then comprises of the definition of the choreographies between the peers in the VO. As such, the previously described Web Services Choreography Description Language (WS-CDL) is well suited for this task. The VO management protocol exposes the common knowledge which the VO members need to share, while at the same time leaving the implementation of the protocol to the individual VO members (Svirskas et al., 2006).

With regards to agent technologies, those can bring various advantages to the domain of management of e-collaborations. The technological and integration aspect is covered by the Foundation for Intelligent Physical Agents (http://www.fipa.org), which tries to maximize interoperability across agent-based applications, services and equipment. Within a Virtual Organization, intelligent software multi-agents can take some of the load in each of the phases of identification, formation, operation and termination of a VO, by automating the relevant processes. Various agent technologies can also be used for the agents' private knowledge, maintenance, specification of various ontologies, and ensuring service interoperability across the value chain.

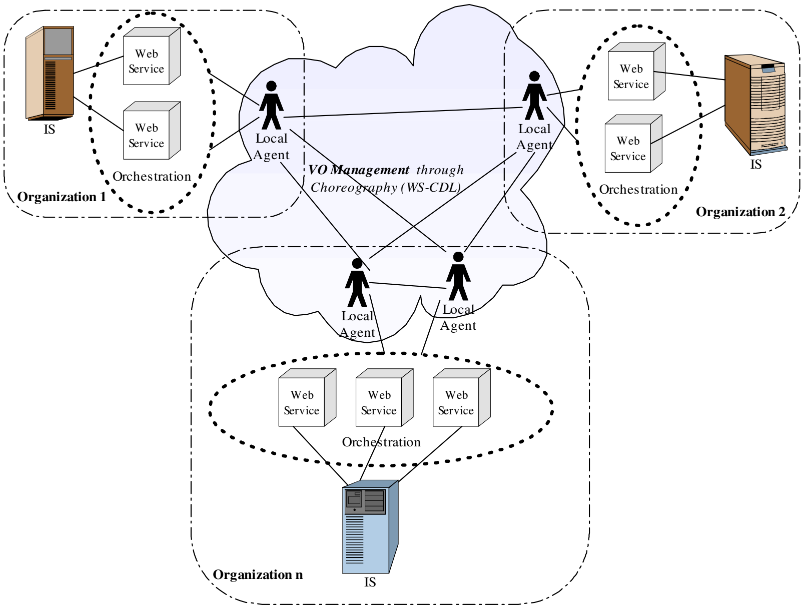

Figure 1 –Virtual Organization Management with Choreography and Agents

Figure 1 above illustrates our proposal for the management of Virtual Organizations using Web Service Choreography and Software Agents. The innovation lies in the fact that whereas Web Service Choreography can be used to coordinate the interactions between Web Services and their consumers, software agents can be inserted in front of those services, and their actions choreographed.

In Figure 1, the proposed solution is a generic one, in the sense that it does not distinguish between the number of agents or their type (e.g. per service, business process, or enterprise). It assumes however, that at least one local agent exists per each organization that participates in a VO. The interaction between the organizations in the VO is carried out with interactions between the respective agents. The latter communicate with the Information System (IS) of the organization via the appropriate Web Services. Whereas within a single organization those Web Services follow orchestration rules, as described in section 2, the whole VO is coordinated with choreography rules that are enacted by each of the local agents assigned to an organization. In that process, agents communicate and exchange information with local agents of other organizations. The use of agents adds flexibility to the operations of the VO, whereas at the same time the use of choreography rules ensures the efficient management of a VO without the need for a centralised service.

The process of the creation of a Virtual Organization has its counterpart in the cooperative team creation or coalition formation processes in the agent technologies domain. In this domain, a group of cooperating agents (coalition) is formed to fulfil a common goal. The individual agents are self-oriented and they don't share all information or their intentions. The agent technologies in this case classify the knowledge as public, private and semi-private. This has a high potential for the management of Virtual Organizations, where there is not a central point of control, but the e-collaborations are rather peer-to-peer, in which case it is important for each peer to have control over the availability of its own information to the other peers in the network and restricting access to the confidential data.

5. CONCLUSIONS

The purpose of this paper has been to present an approach for the automatic management of Virtual Organizations, using Web Service Choreography and (software) agents. Virtual Organization management in big conglomerates dominated by a large organization (e.g. in aerospace or automotive industries) can be implemented with a centralised approach, where the VO participants have to follow the mandates from the leading organization. However, most interactions in VOs are dynamic and among peers, which would make a centralised approach not viable. As such, this paper offers an important dimension in discussing the role of Web Service Choreographies and Multi-Agent systems for the (automatic) operation and management of a VO, without the need for a central hub of control.

Web Service Choreography, and particularly WS-CDL, need to be further developed and supported by the industry in order to successfully implement and support VO management. As was described in this paper, it is also important to link choreography descriptions to agent technologies, and this is an area of research worth pursuing in more detail. Some of the authors of this paper are currently engaged in designing the proposed architectural framework for an EU-funded FP6 IST project, entitled "Collaborative Process Automation Support using Service

Level Agreements and Intelligent dynamic Agents in SME clusters" (PANDA-Project, 2006). The results of this project will contribute towards that research.

6. REFERENCES

1. Aldrich DF. Mastering The Digital Marketplace. Chichester: John Wiley, 1999.
2. Alonso G, Casati F, Kuno H, Machiraju V. Web Services Concepts, Architectures and Applications. Springer-Verlag, 2004.
3. Bovet D, Martha J. Value Nets: Breaking the Supply Chain to Unlock Hidden Profits. Chichester: John Wiley, 2000.
4. Davidow WH, Malone SM. The Virtual Corporation - Structuring And Revitalizing The Corporation For The 21st Century. New York: Harper Collins, 1992.
5. Hagel J, Armstrong A. Net.Gain: Expanding Markets Through Virtual Communities. Harvard: Harvard Business School Press, 1997.
6. Jennings NR, Bussmann S. Agent-based control systems. IEEE Control Systems Magazine 2003; 23:3, 61-74.
7. Maamar Z, Mostefaoui SK, Yahyaoui H. Toward an Agent-Based and Context-Oriented Approach for Web Services Composition. IEEE Transactions on Knowledge and Data Engineering 2005; 17:5, 686 - 697.
8. Marik V, McFarlane D. Industrial Adoption of the Agent-based Technologies. IEEE Intelligent Systems 2005; 20:9, 27-35.
9. Maximilien EM, Singh MP, Multiagent System for Dynamic Web Services Selection. 2005a, http://www.csc.ncsu.edu/faculty/mpsingh/papers/mas/aamas-socabe-05.pdf.
10. Maximilien EM, Singh MP. "Toward web services interaction styles", In Proceedings of the 2005 IEEE International Conference on Services Computing (SCC'05), Orlando, Florida, USA, 11-15 July, 2005b, pp 147-154.
11. PANDA-Project, EU IST-027169, www.panda-project.com.
12. Papazoglou MP, Dubray JJ, A Survey of Web Service Technologies. Trento, Italy: University of Trento: #DIT 04-058; 2004, http://eprints.biblio.unitn.it/archive/00000586/01/mike.pdf.
13. Pechoucek M, Riha A, Vokrinek J, Marik V, Prazma V. ExPlanTech: applying multi-agent systems in production planning. International Journal of Production Research 2002; 40:15, 3681-3692.
14. Ross-Talbot S. "Orchestration and Choreography: Standards, Tools and Technologies for Distributed Workflows", In Proceedings of the NETTAB Workshop - Workflows management: new abilities for the biological information overflow, Naples, Italy, 2005, http://www.bioinformatics.org/NETTAB/2005/docs/NETTAB2005_Ross-TalbotOral.pdf.
15. Ross-Talbot S, Brown G. "Dancing in time with the services: WS-CDL", In Proceedings of the NY Java SIG, NYC, USA, 2005, http://www.javasig.com/Archive/lectures/JavaSIG-CDL-SRT.ppt.
16. Saabeel W, Verduijn TM, Hagdorn L, Kumar K. A Model Of Virtual Organization: A Structure And Process Perspective. Electronic Journal Of Organizational Virtualness 2002; 4:1.
17. Sieber P, Griese J. "Virtual Organizations As Power Asymmetrical Networks", In Proceedings of the 2nd International VoNet Workshop: Organizational Virtualness And E-Commerce, Zurich, 1999.
18. Strader TJ, Lin F, Shaw MJ. Information Structure for Electronic Virtual Organization Management. Decision Support Systems 1998; 23, 75-94.
19. Svirskas A, Wilson M, Roberts B. "Role and Applications of Pluggable Business Service Handlers in Web Services Choreography", In Proceedings of the WWW 2006, Edinburgh, UK, May 22-26, 2006.
20. Svirskas A, Wilson MD, Arenas AE, Lupu EC, Tuptuk N, Chadwick D, Giambiagi P, Dimitrakos T, Roberts B. "Aspects of Trusted and Secure Business-Oriented VO Management in Service Oriented Architectures", In Proceedings of the The First IEEE International Workshop on Service oriented Solutions for Cooperative Organizations (SoS4CO' 05), July 19, 2005.
21. Timmers P. Electronic Commerce: Strategies and Models for B2B Trading. Chichester: John Wiley, 1999.
22. Vinoski S. The Truth about Web Services. Web Services and Component Technologies 2001, http://www.proinfo.com.cn/chinese/menu2/3/102501.ppt.
23. Web-Services-Architecture, W3C Working Group Note, http://www.w3.org/TR/ws-arch/.
24. WS-CDL, Web Services Choreography Description Language Version 1.0, W3C Candidate Recommendation, http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/.