AN AGENT-ORIENTED EXECUTIVE MODEL FOR SERVICE CHOREOGRAPHY

MOHAMMAD ZAHIRI, MOHAMMAD R. KHAYYAMBASHI

Department of Computer Eng. and Information Technology, University of Sheikh Bahaei, Isfahan, Iran
Computer Department, Faculty of Engineering, University of Isfahan, Isfahan, Iran

ABSTRACT

Quick changes in requirements and opportunities in world market needs different levels of cross-organizational collaboration for integrating distributed information systems, information sharing and coordination of organizational processes. Nowadays, Web Services are the most common technology to meet these requirements. Web Services Choreography Description Language (WS-CDL), a World Wide Web Consortium (W3C) choreography-based standard, describes how a number of services coordinate to obtain the goal of such collaboration. Only a few WS-CDL based executive models have been proposed so far. Software Agents are other alternatives for solving Inter-Organization coordination problems. This paper presents an execution framework for WS-CDL using software agents. This framework provides the Web Services collaboration layer based on choreography model and automatically of agent generation of WS CDL. It also follows Web Services stack and native features of agents and Web Services.

Keywords: *Web Services (WS), Choreography, Multi-Agent System (MAS), WS-Choreography Description Language (WS-CDL), Java Agent Development framework (JADE).*

I. INTRODUCTION

Organizations are faced very dynamic environment. Quick requirements and opportunities changing in world market needs different levels of inter-organizational collaboration for integrating distributed information systems, information sharing and coordination of organizational processes. Central systems are being replaced by business networks in which each organization provides some services while uses other organization's services.

During these evolutions, Virtual Enterprise (VE) concept is presented for covering these requirements and utilizing of opportunities in contrast to prior organizations with fixed and predefined frameworks. VE originated some new issues such as developing and automatic management of inter-organization business processes. For practical use and implementation of VE, organizations are faced two major problems: distribution and heterogeneity [1].

Web Service (WS) technology solves these

problems by creating a multi-layer distributed architecture which is compatible with Service Oriented architecture (SOA). Wide-spread use of WS encouraged organizations as well to implement inter-organizational collaborations using them. Therefore World Wide Web Consortium (W3C) for collaboration and composition presents two WS-Business Process Execution Language (WS-BPEL) and WS-CDL standards. In spite of WS's capabilities, WS is not solely able to satisfy all cross-organizational collaboration requirements such as automation, adaptation, flexibility and distribution native of collaboration problem [2], [3].

In according to WS's specifications it can concluded that WS are more suitable for implementing operational requirements and central intra-organization coordination rather than being used them for distributed collaboration requirements.

Software Agents, with their features like autonomy, synchronous and asynchronous interactivity, distribution native and event orientation are another option for implementing for inter-organizational collaborations. The use of agents in both operational and collaboration requirements makes the system implementation complicated and very time consuming. This problem is more severe in distributed and heterogeneity systems. In according to software agent's specification it can be concluded that software agents are more suitable for just implementing distributed collaboration requirements rather than use for both operational and collaboration requirements. A lot of frameworks and tools have been implemented for agent-base programming. JADE is the most conventional framework for agent oriented programming [1], [4].

Combination of WS and software agents will build future of computing paradigm [1]. This paper presents a model for implementation of collaboration layer in WS as MAS. The MAS is generated of WS-CDL automatically. In the other word, this model makes WS-CDL executable following the WS principles by using MAS. At this model each agent assign to one organizational role. Assigned agent is responsible for calling internal Web Services and interacting with other generated agents to achieve defined goal.

This paper is organized as follows: basic concepts are introduced in section II, in section III relatives works in making WS-CDL executive are stated, the proposed model is described in section IV, and conclusion and future works are explained in section V.

II. BASIC CONCEPTS

In this section the basic concepts and technologies that used is this model are introduced.

A. Service Oriented Architecture

SOA is a computing paradigm for generating distributed and heterogeneous software systems. Services are basic elements for generating application in SOA. Service is a platform independent, reusable and loosely coupled software component that is described, registered and discovered. The most obvious capability of SOA is assembling of several services to create a new service [2], [3].

There are two basic models for service coordination in SOA: Orchestration and Choreography. W3C describes choreography as " the sequence and conditions under which multiple cooperating independent agents exchange messages in order to perform a task to achieve a goal state" and orchestration as "the sequence and conditions in which one Web service invokes other Web Services in order to realize some useful function"[5]. WS-CDL [6] and WS-BPEL [7] are W3C standards for two mentioned coordination models.

Two main difference exist between these models are: 1) Existence of central coordinator component in orchestration models. 2) Using of orchestration model let to create new composite service [2]. Based on native specifications of two models it can be said that orchestration relates to intra-organization coordination but choreography refers to inter-organization collaboration (Table I) [8], [9], [10].

Table I. orchestration vs. choreography

Choreography	Orchestration
The Collaboration layer specification	The Composition layer specification
Information Driven	Explicitly Invoking
Among participants	Within single participant
Distributed controlling	Centralizing controlling
Peer-to-peer	Centralized executer
Dynamic topology support	-

B. An overview of WS-CDL

Because of following SOA principals and utilizing of XML (Extended Markup Language) exclusive features in introducing basic standard, Web Services technology is the most common way to implement SOA [8].

WS-CDL is an eXtended Markup Language (XML) based standard used for describing peer-to-peer collaborations among multiple services to achieve a specific goal. WS-CDL specifies the service behavior from the global view point. The position of WS-CDL in WS stack is shown in Fig.1 [6].

Various WS-CDL elements can be categorized as shown in Fig. 2 [11]. In the following paragraphs basic elements of WS-CDL are introduced [6], [12].

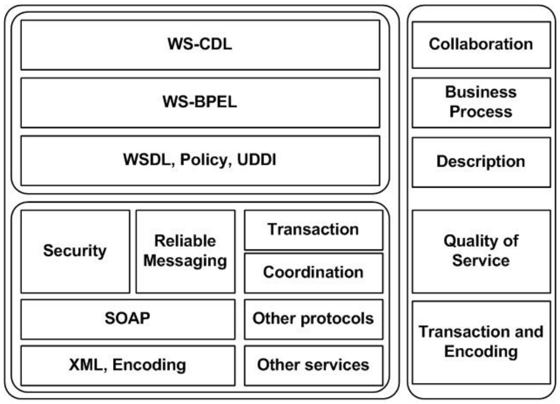

Fig.1Web Service's Stack

Collaborations. The collaborations of choreoghraphy are specified by defining *participentTypes*, *roleTypes*, *relationshipTypes* and *channelTypes*. These declarations define the collaborating participants and their couples.

A *participentType* declares an entity playing a particular set of roles in the choreography. Thus *participentType* definition contains one or more *roleType* definitions.

A *roleType* defines a role that enumerates the observable behavior that a participant can exhibit in order to interact throughout a message exchange.

The relations between roles are defined through *relationshipType* definitions. A *relationshipType* always contains exactly two *roleTypes*, restricting the *relationshipType* definition to 1:1 relations.

A *channelType* definition specifies where and how information between *roleTypes* is exchanged. This *roleType* reference indicates the behavior interface which is used throughout the information exchange.

Information handling. The information and handling of information within a choreography is perform by *informationTypes*, *variables* and *tokens*.

Information used within choreography is specified by *informationTypes* which do not directly reference data types but rather reference type definitions. Such a referenced type definition can be a Web service Description Language (WSDL) 1.0 message type or an XML schema type, a WSDL2.0 schema element or an XML schema element.

Variables capture information about objects in choreography such as information exchanged or the observable information of *roleType*s involved and are either bound to *informationType* or *channelType* definitions. These *informationTypes* can either belong to application or state information.

Tokens are enumerated as aliases for *informationTypes*. *Tokens* can be used in correlation of exchanged messages by declaring in channels or defining as the identifiers.

Activities. A choreography comprises three different types of activities, namely ordering structures, workunit and basic activities.

Ordering structures are block structured, enclosing a number of activities or ordering structures which can be used recursively. Such activities include *sequence* for handling activities in sequential order, *parallel* for a parallel execution of activities and *choice* for handling data and event-driven conditions.

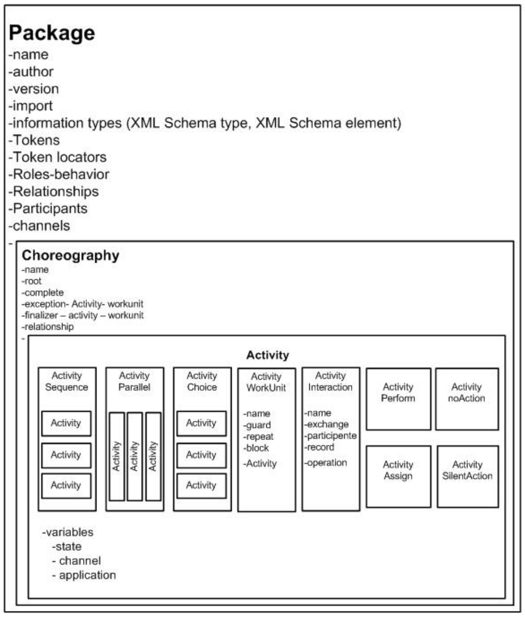

Fig.2 WS-CDL's structure

Workunits prescribe the conditional execution of an activity. This conditional execution can either be repetitive, competitive or blocking.

Basic activities define interactions, actions or variable assignments of choreography flow. An *interaction* activity defines the information to be exchanged and by what means this information exchange will be performed.

The other basic activities include *assign*, *silentAction*, *noAction* and *perform*. The *assign* activity enables the creation and manipulation of variables within choreography. The *silentAction* defines a non-observable behavior which is either performed by one or all participants.

The *noAction* activity represents a point in choreography in neither which no special task is performed by a roleType, neither manipulation of state information (non-observable behavior) nor interactions (observable behavior).

The *perform* activity is also used to call another choreography to be performed within the context of executing choreography. The called choreography may be defined within the same package as the caller, or it may be from a completely separate package that has been imported.

Some of researchers believe in WS-CDL as an exclusive descriptive standard. In contrast, other researchers know WS-CDL as an executive standard that is only a standard compatible with SOA definitions. These researchers implemented some tools such as WS-CDL+ [10] and Pi4soa [13].

C. Multi-Agent System and JADE

A MAS is a society of intelligent software agents that interact with each other to achieve a common goal. Agent is a software component with some unique features such as autonomy, state-full, social ability, having internal knowledge base, goal orientation, reflectivity, interaction with other agents and environment. Agent oriented programming are usually used for solving distributed controlling problems [4], [14].

JADE is one of the most common open source agent oriented middle wares. So of its benefits are full compatibility with the Foundation for Intelligent Physical Agents (FIPA) specifications, reliability and fault tolerance, having completed libraries and documents, implementation with pure java codes and ability to use java capabilities easily in development new multi-agent systems [4].

III. RELATED WORKS

In [10], [13], [15], [16] available problems in executing WS-CDL have been discussed and needed requirements for implementation WS-CDL have been specified.

Two various models for execution WS-CDL have been proposed: WS-CDL+ and Pi4soa that use central WS-BPEL engine over intra-organization WS-BPEL engines to execute WS-CDL. At this model inter-organization collaboration logic is executed by the central engine [10], [13], [15]. The major fault of this model is existence of central collaboration engine that alleviates flexibility. In [16] WS-CDL is executed by using aspect oriented concepts that is applied in web service' handlers. At this model collaboration logic is executed by aspects that glow to web service's handlers. At this model collaboration layer concepts are integrated with operational layer concepts that aren't compatible with SOA. If one service participates in different collaborations with distinct logics it causes some problems and may decreases generality of service.

Some other works have tried to combine web service and agent to create new component software. The used approach in generating such component is against to separate coordination tasks from operational tasks. Developing new system with such component needs new methodologies with a cumbersome and complicated process [1].

IV. PROPOSED MODEL

Inter-organization collaboration implementation encounters two major challenges: 1) Creating an integrated infrastructure over a heterogeneous and distributed environment. 2) Use of distributed mechanisms to apply collaborations. In according to above discussions, it can be said that WS can be a suitable choice for solving first challenge and software agents technology is coping with the second one.

Inter-organization collaborations actually are performed by some organization's experts that are aware of their organization's operational capabilities and can also interact with other organization's experts based on special contracts.

By mapping organization's experts to intelligent agents, organization's operational capabilities to Web Services and inter-organization's contract to WS-CDL, the proposed model is formed.

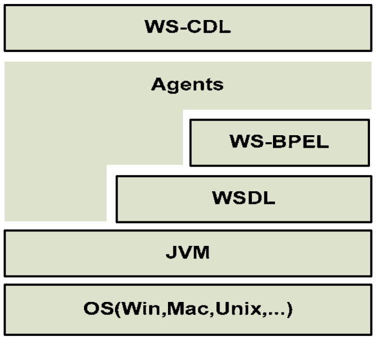

Fig.3 the proposed multi layer model

This model is presented in web service architecture. In the other hand web service's collaboration layer that presents with WS-CDL standard, is implemented as multi-agent systems. This model just focuses on collaboration layer and lets the other layers being left intact. The Agents are able to call atomic, composite Web Services and use java classes (Fig.3).

A WS-CDL file, which is designed from the real Web Services, is the main input for generating software agents and corresponding deployment file based on JADE specifications (Fig.4).

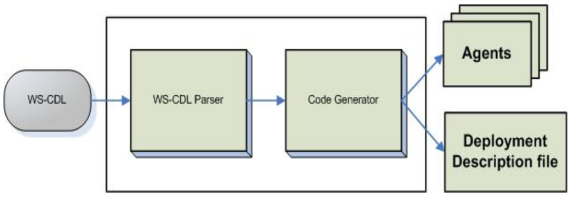

Fig.4 Agents and deployment file generate from WS-CDL

A business process has various perspectives. The overall perspective specifies business goal. Control perspective specifies required knowledge about different execution procedures. In operational perspective required activities are indentified. Informational perspective includes data that are produced, consumed or exchanged during process execution. Organizational perspective specifies organization's assigned activities (Fig.5) [17].

The *participentType* and *roleType* elements specify organizational perspective. The *informationType* and *variable* element specify informational perspective. The Control perspective is specify by *sequence*, *parallel*, *choice* and conditions part in the *workunits*. Basic activities and behaviors in *roleTypes* are shown operational perspective.

In WS-CDL standard, one *roleType* controls process execution at any time and process control is transferred to any other *roleType* simultaneously. Using of orchestration or choreography models for this transformation depends on the location of *roleType*, being in the same *participentType* or any other *participentType*.

Each *roleType* is implemented by one software agent and each behavior in *roleType* is mapped to one agent's behavior. Each Agent depends on its *participentType* located in one JADE platform in corresponding organization. At intra-organization scope agent acts as an orchestration engine and calls internal Web Services. At inter-organization boundary, the same agent acts like one of distributed collaboration components and interacts with other agents (Fig.6).

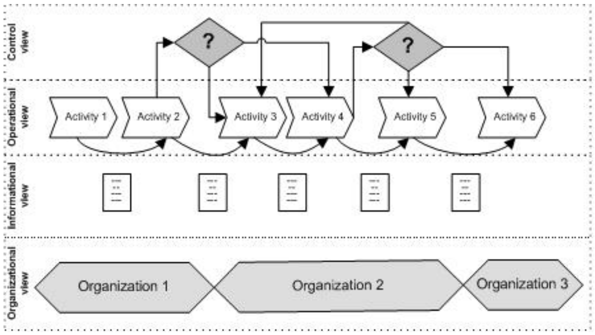

Fig.5 business process's various perspectives

V. CONCLUSION

This model used MAS for execution WS-CDL. This model has sum benefits in comparing with prior models:

- Automatic generation MAS and don't need to development's process for making distributed collaboration system.
- No need to have a central executive engine.
- Separation of collaboration and operational issues.
- Transparency in intra-organization business process level.
- Ability to apply MAS techniques for adding semantics concepts to this model and utilizing service discovery.
- Use adaptively and cooperatively in MAS for make inter-organization collaboration adaptively.

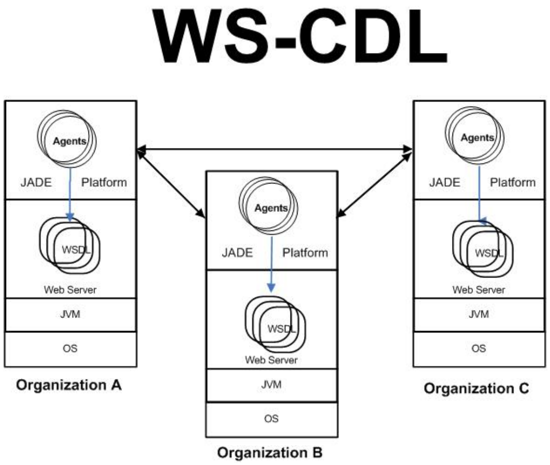

Fig.6 distribured deployment architecture

REFERENCES:

[1] N. Protogeros, *Agent and Web Service Technologies in Virtual Enterprises*. Information Science Reference, 2008, pp. 1–153.

[2] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, August 2005, ch.1, ch.2.

[3] N.M. Josuttis, *SOA in Practice*, O'Reilly, August 2007, pp.1-229.

[4] F. Bellifemine, G. Caire, D. Greenwood, *Multi-Agent Systems with JADE*, wiley, 2007,20-80.

[5] Daniel F. and Pernici B. ,"Insights into Web Service Orchestration and Choreography", International Journal of E-Business Research, 2(1), 58-77, January-March 2006.

[6] W3C. "Web Service Choreography Description Language (WS-CDL)", Nov. 2005. URL: http://www.W3.org/TR/ws-cdl-10/ (Last accessed: April. 19, 2009).

[7] OASIS. "Web Service Business Execution Language 2.0". 2006. URL:http://www.oasis-open.org/committees/tc _home.php?wg _ abbrev=wsbpe (Last accessed: June. 5, 2009).

[8] T. Erl, *Service-Oriented Architecture*: *Concepts*, *Technology*, *and Design*, Prentice Hall PTR, August 2005, pp. 235-245.

[9] C. Peltz," Web Service Orchestration and choreography", IEEE Computer Society, 2003, pp. 46-52.

[10] Z. Kang, H. Wang, P.C.K. Hung, "WS-CDL+: An Extended WS-CDL Execution Engine for Web Service Collaboration", IEEE International Conference on Web Services (ICWS 2007), 2007, 928-935.

[11] Alistair B, Marlon D, Phillipa O, "A Critical Overview of the Web Services Choreography Description Language(WS-CDL)", March 2005,URL: http://www.bptrends.com/publicationfiles/03-05%20WP %20WS-CDL%20Barros%20et%20al.pdf

[12] F. Rosenberg, C. Enzi, A. Michlmayr, C. Platzer and S. Dustdar, "Integrating Quality of Service Aspects in Top-Down Process Development using WS-CDL and WS-BPEL", Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference, 2007, 15-25.

[13] Pi4 Technologies Foundation. *pi4soa*, 2007. URL: http://sourceforge.net/projects/pi4soa/ (Last accessed: February. 10, 2009).

[14] Weming S., et al.," An agent-based service-oriented integration architecture for collaborative intelligent manufacturing", 2006 Elsevier Ltd, Available from: http://www.elsevier.com/locate/rcim.

[15] lars-Ake Fredlund, "Implementing WS-CDL", LSIIS,2008, from: http://babel.ls.fi.upm.es/~fred/papers/jsweb2006.pdf

[16] Thomas C, Tzilla El, "Executable Choreography Processes with Aspect-Sensitive Services", IEEE, 2006, from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.9285&rep=rep1&type=pdf.

[17] Schmidt R., "Sercomp: A Component Oriented Method for Flexible Design and Support of Interorganizational Service Processes", online 28 September 2006 in Wiley InterScience, from: http://www.interscience.wiley.com