Extending Web Services Architecture with Agent Models and Patterns

Emergent Research Forum (ERF)

Habibullah Aminu  
American University of Nigeria  
habibullah.aminu@aun.edu.ng

Muhammadou M.O. Kah  
American University of Nigeria  
mkah@aun.edu.ng

Kweku-Muata Osei-Bryson  
Virginia Commonwealth University  
kmosei@vcu.edu

Abstract

The Web Services Architecture (WSA), which is a conceptual model for understanding web services, is composed of four models, with agents prevalent in all the models. Despite the centrality of agents to web services, there is no coherent subset of the Architecture encapsulating agents. Thus, this paper presents how the WSA is re-conceptualized to include the Agent-Role Model (ARM) as the focal point of the Architecture towards which the other four models converge. The extended architectural model would therefore enable web services to provide the more basic level functionality of invocation; and agents provide higher-level functions by acting as autonomous entities that incorporate intelligence in using, combining and choreographing web services, that were lacking in pre-existing models.

Keywords

Web Services Architecture, Agents, Patterns, Design Science Research, Bioinformatics, EBI Web Services, Critical Realism.

Introduction

The Web Services Architecture (WSA), which is an interoperability architecture, presents a conceptual model and a context to understand web services and the relationships between the components of the model (Booth et al., 2004).

Web Services Architecture

Four models compose the reference architecture [Figure 1]:

- Service-Oriented Model - revolves around the key concepts of services being mediated by means of the messages exchanged between requester and provider agents.
- Message-Oriented Model - revolves around the key concepts of agents sending and receiving messages.
- Resource-Oriented Model - focuses on resources which agents discover by retrieving Web service-related descriptions.
- Policy Model - focuses on constraints on the behaviour of agents and services.

Figure 1: Meta-model of the WSA
Research Problem

Agents are prevalent in all the models. But despite the centrality of agents to web services, there is no coherent subset of the reference architecture encapsulating agents.

Theoretical Framework and Contribution

The different roles agents play in the WSA have been identified in Table 1:

**Agent Role Model (ARM)**

<table>
<thead>
<tr>
<th>Context</th>
<th>Object</th>
<th>Agent Role</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message-Oriented Model (MOM)</td>
<td>Message</td>
<td>Sender</td>
<td>send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Receiver</td>
<td>receive</td>
</tr>
<tr>
<td>Service-Oriented Model (SOM)</td>
<td>Service</td>
<td>Requester</td>
<td>request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider</td>
<td>provide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Broker</td>
<td>link</td>
</tr>
<tr>
<td>Resource-Oriented Model (ROM)</td>
<td>Resource</td>
<td>Requester</td>
<td>discover</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer</td>
<td>use</td>
</tr>
<tr>
<td>Policy Model (PM)</td>
<td>Policy</td>
<td>Contractor</td>
<td>comply</td>
</tr>
</tbody>
</table>

**Table 1. Agent Roles in the Web Services Architecture**

The agent roles have been conceptualized into an Agent Role Model (ARM). Because of the centrality of agents to WSA, the ARM has been positioned as the focal point of the WSA towards which the other four models converge [Figure 2].

**Figure 2: The ARM positioned as the focal point of the Architecture**

**Agent Communication Model (ACM)**

The Agent Communication Model is used to structure agent communication channels and organize the roles and responsibilities of agents in the communication.

**Figure 3: The Conceptualized Agent Communication Model (ACM)**

Comparing the agent communications in the MOM and the SOM reveal that the communication between the service requester and the service provider is translated into the message passing between the message
sender and the message receiver. Figure 3 illustrates the translation of the SOM as a message-based model.

![Figure 4: MAS Design & Development based on DSR Methodology](image)

**Design Patterns**

Design patterns, which are a valuable mechanism for emphasizing structure, have been investigated:

The ARM manifests the **Role Pattern**, while the ACM manifests the **Proxy design pattern** and the **Chain of Responsibility** (Gamma et al., 1995).

**Theoretical Background**

The theoretical contribution of this study is in the modification (or improvement) to an existing conceptual framework. This falls within the purview of **meta-design** (Fischer and Giaccardi, 2006), which deals with the crucial aspect of “designing design”, here denoting higher conceptual designed structure of Web services.

**Research Approach**

**Research Methodology and Philosophical Paradigm**

Our approach focuses on the development of solutions for practical problems and, thereby, on accomplishing utility (Hevner et al., 2004). In order to actually achieve utility, Peffers et al. (2007) describe a DSR methodology for the construction and presentation of DSR artefacts in the Information Systems context. The literature and contributions on the phenomena of an Agent-centric WSA is grossly inadequate and we will not be constrained in our selection of research methodology because we subscribe to the critical realism philosophical paradigm, with its commitment to methodological eclecticism (Archer et al., 2016).
Extending WSA with Agent Models and Patterns

Agent-based Systems Engineering

The Design and Development phase from the DSR Methodological Framework in this study, presupposes the translation of the conceptual models into implementation of agent roles and posits a WSA based on Multi-agents in which the Multi-Agent System (MAS) is positioned as the focal point of the Architecture.

In line with (García-Sánchez et al., 2009), the main idea behind agents within the proposed Architecture is not to be able to provide services, but to act as autonomous entities that incorporate intelligence, which allow them to show a goal-directed behaviour, and to interact with other software entities in order to satisfy their design objectives.

The phases for the implementation of an Agent-centric WSA are highlighted [Figure 4] and explained below:

1) **Requirements Analysis:** Analysing requirements for the design and development of the MAS leverages methods and techniques from goal-oriented analysis ((Mylopoulos et al., 1999) & (Nakagawa et al., 2006)).

2) **Analysis and Design:** For the design of the MAS, the Gaia methodology is proposed. This is informed by the fact that Gaia defines the structure of a MAS in terms of a role model (Moraïtis et al., 2003), and for its simplicity yet comprehensible set of models (Karageorgos et al., 2002).

3) **Implementation:** Considering the limitations of the Gaia methodology for not directly dealing with implementation issues (Zambonelli et al., 2003); JADE ((Moraïtis et al., 2003), (Bellifemine, 2007)), a FIPA-compliant agent development framework is being used in the engineering of the agents with the Gaia methodology.

Empirical Situation

The case study being considered is the utilization of the European Bioinformatics Institute (EBI) web services, in the determination of protein function from genomic data via the following analytical steps (Kellis, 2016):

- the determination of protein sequence from DNA sequence;
- the determination of protein structure from protein sequence, and the determination of protein function from protein structure.

Composition of Bioinformatics Workflows

Modern biological data analysis requires the integration and execution of Web services in a particular order, creating workflow among participating entities. A typical workflow (updated from Mcwilliam et al., 2009) is highlighted below:

- The workflow retrieves initial DNA sequence and then uses EMBOSS (Pairwise Sequence Alignment (PSA) tools) to discover the coding regions.
- BLAST + (a Sequence Similarity Search (SSS) tool) is used to search for previously-characterized sequences that are homologous to these regions, which are then retrieved and used to create a Multiple Sequence Alignment (MSA) via CLUSTALWO.
- It also searches the InterProScan 5 (Jones et al., 2014) database for domain families (Protein Feature Detection) and retrieves annotation from scientific literatures and ontologies.

In the platform proposed to run the system, there are 6 types of agents, grouped in three main categories:

- Agents that perform management tasks: Broker Agent and Coordinator Agent
- Agents that act on behalf of service owners: Provider Agent and Service Agent
- Agents that act on behalf of service consumers: Consumer Agent and Selection Agent

The actual implementation of agents is not determined at compile time, but rather at run-time. For this reason, ‘roles’ are introduced as abstractions for agent collaboration and mobility (Kendall, 2000), and as a way to document patterns of interaction. The introduction of the role concept presents a number of benefits (Zhao et al., 2004): dynamicity and flexibility, responsibility-driven, and context-sensitivity. The

---

1 Foundation for Intelligent Physical Agents
preliminary Agent roles identified in the extended MAS-based Web Services Architecture are the Invoker, Selector, Composer, Broker, Service Representative, Requester Representative, Primary and Secondary Service Provider Representatives.

REFERENCES