

JADE Web Services Integration Gateway

Whitestein Technologies and the JADE Board
Contact: Dominic Greenwood, dgr@whitestein.com

1 Introduction

Development of the WSIG - Web Service Integration Gateway - was motivated by the recognition that JADE,¹ as one of the most widely deployed open source agent systems, should have the intrinsic capability of seamlessly invoking Web services and allowing Web service clients to make use of agent services. Any agent service to be exposed for consumption by Web services should be invocable using basic request-response operations as this is the communication mode of typical Web services. The WSIG is available as a component of the JADE agent development framework (v3.3).

Integrating Web services and software agents brings about the immediate benefits of connecting application domains hosting one or the other technology: A Web service should be able to invoke an agent service and vice versa. Once this interconnection is established software agent concepts and technologies should help enable new operational and usage modalities of Web services. Due to the evident technology mismatches between Web services and software agents, including strong vs. loose coupling and representational encodings, we have identified an approach that introduces an intermediary service entity designed to encapsulate the functionality required to connect the two domains, whilst ensuring minimal human intervention and service interruption.

Three key features of the WSIG are: • *Transparency*. The WSIG is designed to be operationally transparent to invoking entities, whether they be agents or Web services. • *Automation*. Once operational, the WSIG requires no manual intervention or configuration. • *Integration*. The WSIG encapsulates and integrates the functionality required to deliver the above features without requiring additional external resources.

2 WSIG Architecture

The Gateway is designed as a collection of functional modules housed within a JADE agent platform. The Gateway offers automatic, bidirectional operation allowing both FIPA compliant agent services and Web services to be registered with it. Agent services and Web services can thereby publish

their service descriptions to consumers outside their normal operational domain. The Gateway can then intercept calls to these registered services allowing agents to invoke Web services and vice versa by transforming message encodings and creating service access endpoints.

The following assumptions were made when designing the Gateway architecture:

- All agents are assumed to be FIPA compliant.
- All Web services use the standard WSU stack.
- Gateway is registered as an agent service in FIPA DF.
- All invocation related interactions between the Gateway and agents use ACL encoded FIPA-Request and FIPA-Inform performatives.

As shown in the figure the Gateway consists of several components, each linked either directly or indirectly to two registries - a Directory Facilitator (DF) for storing agent service descriptions and a UDDI service for storing Web service descriptions. The DF is the same instantiation as that used by all agents within the agent platform and is not visible outside the platform. The UDDI belongs solely to the Gateway, is visible internally to the Gateway and externally to Web services and Web service clients, but not directly to agents. Each registry exposes the standard operations: Registration, deregistration, modification and discovery.

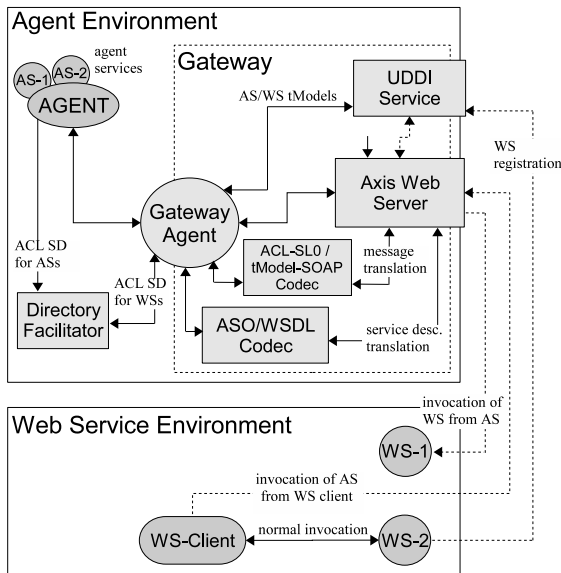
Any service description that is registered with either the DF or UDDI registry is automatically translated into an entry for the other. This duplication ensures that any registered Web service is visible to agents via the DF and any registered agent service visible to Web service clients via the UDDI.

The Gateway Agent

The Gateway Agent which is implemented as a standard JADE agent, with behaviours controlling access to a number of local components (codecs, Web server, etc.), operating as independent threads. The Gateway Agent is responsible for the following tasks:

- Receiving notifications from the agent platform DF that an agent has registered an ACL Service Description (SD) that it wishes to expose for invocation as an external Web service.
- Administering the mapping of newly registered ACL SDs into tModels for publication via the Gateway UDDI.
- Receiving notifications from the Gateway UDDI that a Web

¹JADE is the Java Agent Development Environment. Additional information can be found at <http://jade.tilab.com>



service has registered a WSDL SD to be exposed for invocation by agents.

- Administering the mapping of newly registered WSDL SDs into agentservice ontologies and ACL SDs for publication via the agent platform DF.
- Receiving requests from platform agents to invoke an external Web service. These requests are expected strictly in the form of FIPA Request messages. Responses from a Web service are returned to the agent in the form of FIPA Inform messages.
- Forwarding requests for service invocation received from external Web services, in the form of FIPA Request messages. Again, responses are expected to be in the form of FIPA Inform messages.

The ACL-SL0/tModel-SOAP Codec

This module is a complex codec that bidirectionally translates FIPA SL0 service descriptions into UDDI tModels and FIPA ACL messages into SOAP messages. These two codecs are unified because ACL/SL0 can be translated into either tModels or SOAP messages according to the usage case. The module is responsible for the following tasks:

- Parsing ACL messages received from the platform DF to extract the SL0 encoded service descriptions (SD) held within their content.
- Translating the SL0 SD into a UDDI tModel and returning the results to the Gateway agent for registration with the UDDI.
- Parsing ACL-SL0 messages sent to the Gateway to invoke a Web service into a corresponding SOAP message.
- Operating bidirectionally to translate SOAP and tModels into correctly encoded ACL-SLO messages and DF entries.

The Axis Web Server

This module is used to both interface with external Web services and their clients. It is responsible for the following

tasks:

- Externally (outside the agent platform) publishing agent services as endpoints that can be called by Web service clients.
- Handling invocation requests for published agent services, issued by Web services.

3 Gateway Operation

The Gateway has four principal modes of operation: (1) Mapping a newly registered agent service into its counterpart Web service counterpart, (2) The inverse mapping of (1), (3) Invoking a Web Service from an agent and (4) Invoking an agent service from a Web service client.

When initiated the Gateway agent starts it first registers with the local agent platform DF.

Registering an Agent Service

Agent Service registrations are made onto the JADE platform DF as normal. These are automatically picked up by the WSIG Gateway agent and translated into an equivalent WSDL/tModel that is cross-registered with the UDDI.

Web service registrations are received by the Gateway via the Gateway UDDI and are subsequently processed into FIPA ACL service descriptions which are stored in the JADE platform DF.

De-registration, modification and discovery are unaffected and operate as would be expected for both the JADE DF and the WSIG UDDI.

Invocation of a Web service by an Agent

An agent sends a standard ACL FIPA Request message to the Gateway containing the identity of the Web service to be invoked and any parameters as properties of the request. Received ACL messages are parsed by the Gateway and a SOAP message constructed using the WSDL service description of the Web service to be invoked. If a response is expected from the Web service a temporary endpoint is established on the Gateway Web server to receive responses. On reception of the response the incoming SOAP message is parsed into an ACL FIPA Inform message and sent to the invoking agent.

Invocation of an agent service by a Web service client

An incoming SOAP message from a Web service client is parsed into an ACL FIPA Request message containing the SL0 encoded service request and sent to the agent containing the service to be invoked. Once the receiving agent processes the service invocation, if required, it will return a response to the Gateway as a FIPA Inform message. This response is then parsed by the Gateway into a SOAP message and returned to the invoking Web service client.