AN IMPLEMENTATION OF EXTERNAL INTERFACE FOR PROVIDING THE KOMPSAT-2 CATALOGUE SERVICE

Jaehong Park, Sungu Lee, Younsoo Kim

Korea Aerospace Research Institute
45 Eoeun-dong, Yuseong-Gu, Daejeon 305-333, Korea
jhpark75@kari.re.kr, leesg@kari.re.kr, younsoo@kari.re.kr

Korea Aerospace Research Institute (KARI) has been trying for utilizing the polar station in Svalbard, Norway to receive the KOMPSAT-2 images, which are distributed and applied directly for Europe Space Agency (ESA) users. Now, KARI is operating the SpaceCapture-2 system that provides the KOMPSAT-2 catalogue service via the internet, whereas ESA is operating the EOLI-SA tool that provides access to the online ESA catalogues of Earth Observation (EO) products. This paper describes external interface based on service-oriented architecture in order to provide the KOMPSAT-2 catalogue service from the SpaceCapture-2 system to the EOLI-SA tool. As the KOMPSAT-2 catalogue service is available in the EOLI-SA tool through an implementation of external interface, the extend use of KOMPSAT-2 imageries is expected.

KEY WORDS: KOMPSAT-2 Catalogue Service, EOLI-SA, Service-Oriented Architecture

1. INTRODUCTION

The KOMPSAT-2 was launched successfully on 28th July 2006 and the imaging data has been archived and distributed since then. Korea Aerospace Research Institute (KARI) has been trying for utilizing the polar station in Svalbard, Norway to receive the KOMPSAT-2 images, which are distributed and applied directly for European Space Agency (ESA) users. Now, KARI is operating the SpaceCapture-2 system (http://www.spacecapture.kr) that provides the KOMPSAT-2 catalogue service via the internet. ESA is an international organization which is established to promote, for exclusively peaceful purpose, cooperation among European States in space research and technology and space application. ESA is operating the EOLI-SA tool that allows users to access catalogue and order services of various satellites such as ENVISAT, LANDSAT, NOAA, and so on.

This paper describes service-oriented architecture and web services to support interoperable machine-to-machine interaction over a network in a decentralized, distributed environment. It also describes external interface based on service-oriented architecture in order to provide the KOMPSAT-2 catalogue service from the SpaceCapture-2 system to the EOLI-SA tool.

2. SERVICE-ORITENTED ARCHITECTURE AND WEB SERVICES

2.1 Service-Oriented Architecture (SOA)

SOA is component model that inter-relates the different functional units of an application, called services, through well-defined interfaces and contracts between these services. It is an architecture pattern on service point of view that IT system immediately copes with change from business model and strategy.

Each operation mentioned in SOA definition consists of service provider, service broker, and service consumer. The component and role of each operation has interaction as shown in Figure 1. The service provider creates a web service and possibly publishes its interface and access information to the service broker. In some cases, the service provider could be the service consumer to invoke service of other service providers. The service consumer locates entries in the service broker using various find operations and then binds to the service provider in order to invoke one of its web services. The service broker is a registry, an intermediary between a service provider that publish their availability and a service consumer that needs to find services.

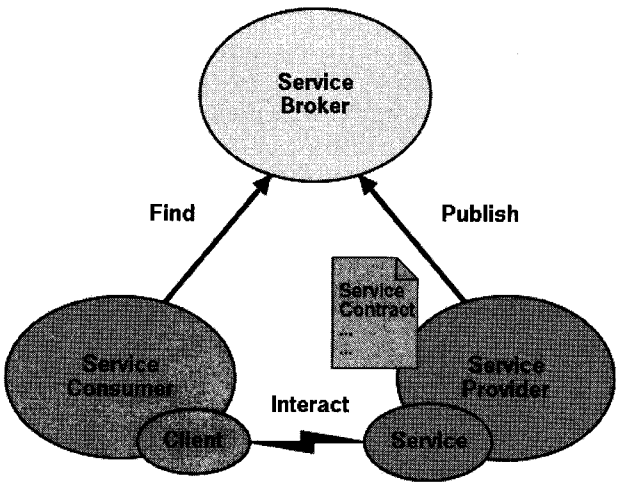

Figure 1. Overview diagram of SOA

2.2 Web Services

Web services are software systems designed to support interoperable machine-to-machine interaction over a network. This interoperability is gained through a set of XML-based open standards, such as SOAP, WSDL, and UDDI. These standards provide a common approach for defining, publishing, and using web services. Web services are the preferred standards-based way to realize SOA.

2.2.1 SOAP

SOAP (Simple Object Access Protocol) is an XML-based protocol to exchange the information in a decentralized, distributed environment. SOAP messages are fundamentally one-way transmissions from a sender to a receiver and often combined to implement patterns such as request/response. All SOAP messages are an XML document with a schema and include proper namespace on all elements and attributes. A SOAP message consists of SOAP envelope, SOAP header, and SOAP body as shown in Figure 2. The SOAP envelop is the top element of the XML document representing the message and is used in a protocol context to describe the complete message frame. The SOAP header element provides a flexible mechanism for extending a message in a decentralized and modular way without prior knowledge between the communicating parties. The SOAP body element provides a simple mechanism for exchanging mandatory information intended for the ultimate recipient of the message. Typical uses of the SOAP body element include marshalling RPC (Remote Procedure Call) calls and error reporting. The SOAP Fault element is used to indicate error messages.

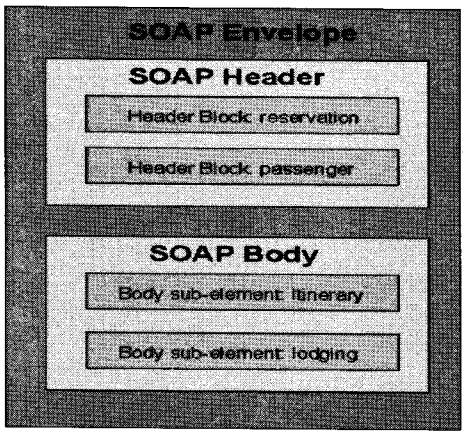

Figure 2. SOAP structure

2.2.2 WSDL

WSDL is web services standards description language to provide the service interface of service provider to service consumer using xml. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

WSDL uses the following elements in the definition of network services:

- Types: a container for data type definitions using some type system.
- Message: an abstract, typed definition of the data being communicated.
- Operation: an abstract description of an action supported by the service.
- Port Type: an abstract set of operations supported by one or more endpoints.
- Binding: a concrete protocol and data format specification for a particular port type.
- Port: a single endpoint defined as a combination of a binding and a network address.
- Service: a collection of related endpoints.

2.2.3 UDDI

UDDI (Universal Description, Discovery and Integration) defines a standard interface for publishing the availability of a service and for finding a required service. A UDDI registry is accessible through web browser or SOAP. The logical structure of UUDI consists of three components: White Pages, Yellow Pages, and Green Pages. White Pages give information about the business supplying the service. This includes the name and description of the business. Yellow Pages provide a classification of the service or business, based on standard taxonomies. Green Pages are used to describe how to access a web service, with information on the service bindings such as the address of the service and the parameters, and references to specifications of interfaces.

A UDDI registry is divided into public UDDI registry and private UDDI registry. Public UDDI registry provided by IBM, Microsoft, and SAP was closed in 2006 due to the lack of governance over the entries. Private UDDI registry is used by either internal, or between close business partners so far.

3. EXTERNAL INTERFACE

3.1 Service Architecture

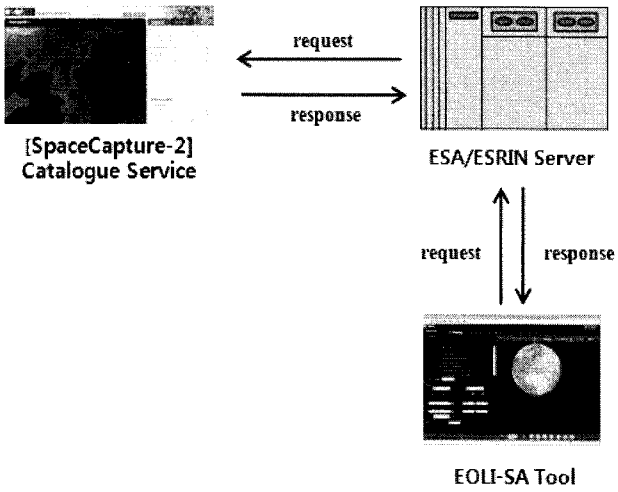

Figure 3. Service architecture

ESA provides the EOLI-SA tool based on service-oriented architecture as shown in Figure 3. The tool allows users to access catalogue and order services of various satellites such as ENVISAT, LANDSAT, NOAA, and so on.

The SpaceCapture-2 system should perform the role of service provider in order to provide the KOMPSAT-2 catalogue service to the EOLI-SA tool and define external interface to describe how to access the catalogue service.

The ESA/ESRIN server includes metadata information for service access to linked satellites. It performs information exchange with service providers in substance to process service request from the EOLI-SA tool. The information exchange for service request and response is performed using SOAP.

3.2 SOAP Messages

Table 1 shows the input and output message sequence to invoke the KOMPSAT-2 catalogue service. The name and types of the element and attributes contained in the message are defined in a schema.

Table 1. Payload Message Sequence

Input Payload Message	Output Payload Message
searchRequest	response
presentRequest	response

The searchRequest message allows to identify products inside a single collection via a search condition and to request the number of hits or product metadata in different formats as Response. A detailed element to compose message is as shown in Figure 4.

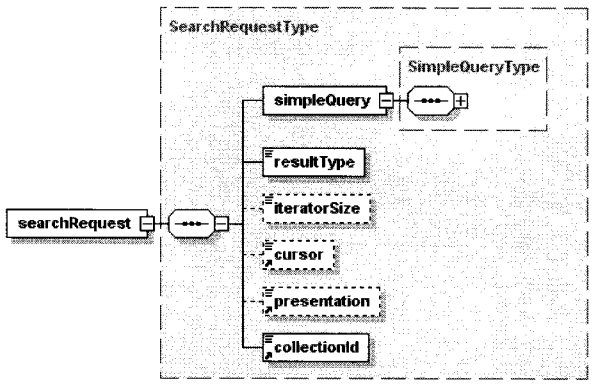

Figure 4. searchRequest diagram

The presentRequest message allows to retrieve (multiple) product metadata within a single collection by providing the product identification. A detailed element to composite message is as shown in Figure 5.

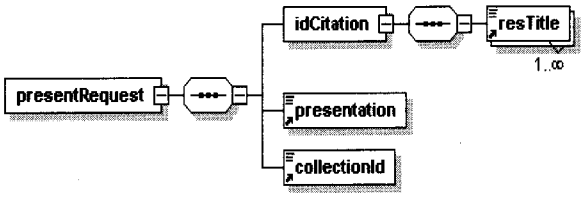

Figure 5. presentRequest diagram

The response message contains the result in form of number of hits or returned product metadata. A detailed element to composite message is as shown in Figure 6. The "presentation" column specifies the use of the element in the different type of descriptors: F(full), B(brief), b(browse), and s(summary). The specific element should be provided if it is available within the various presentations.

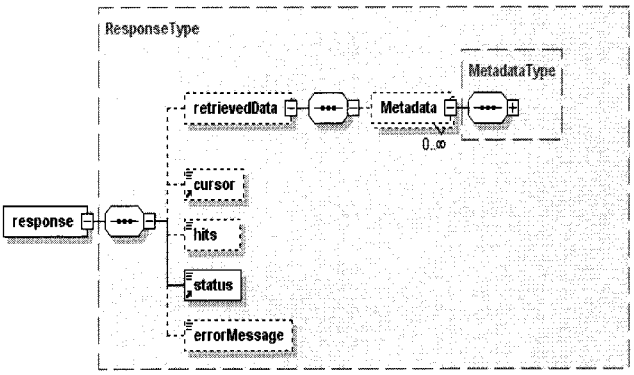

Figure 6. response diagram

3.3 WSDL

WSDL for the KOMPSAT-2 catalogue service from the SpaceCapture-2 system is as shown in Figure 7.

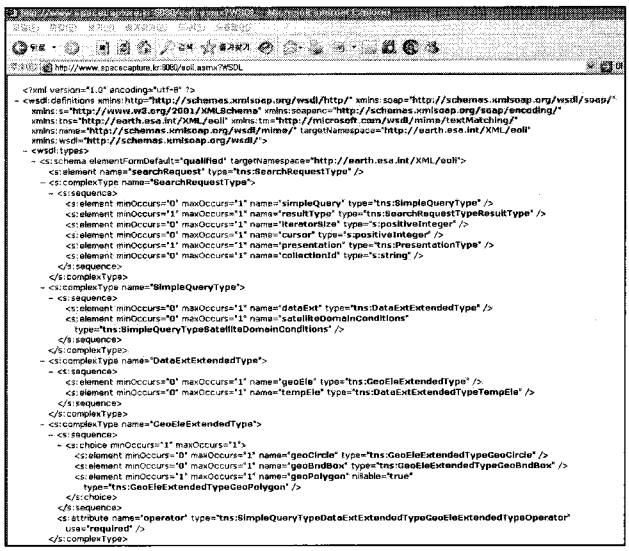

Figure 7. WSDL for KOMPSAT-2 Catalogue Service

3.4 Catalogue service linkage

We implemented external interface in VB.NET based on IIS (Internet Information Services) to provide the KOMPSAT-2 catalogue service from the SpaceCapture-2 system to the EOLI-SA tool. The KOMPSAT-2 catalogue service is available in the EOLI-SA tool as shown in Figure 8.

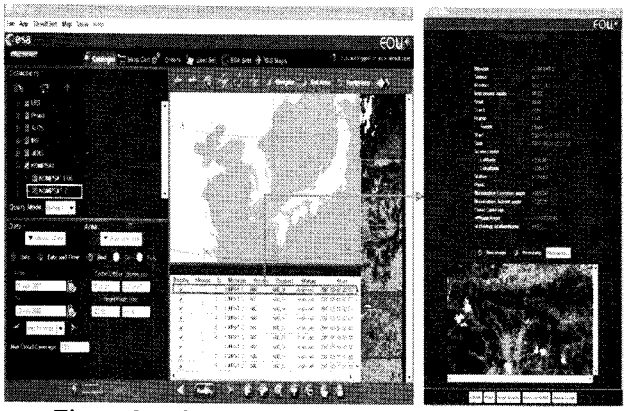

Figure 8. KOMPSAT-2 catalogue service linkage through the EOLI-SA tool

4. CONCLUSION

This paper described external interface based on service-oriented architecture and web services in order to provide the KOMPSAT-2 catalogue service from the SpaceCapture-2 system to the EOLI-SA tool. As the KOMPSAT-2 catalogue service is available in the EOLI-SA tool through an implementation of external interface, the extend use of KOMPSAT-2 imageries is expected.

5. REFERENCES

[1] Xiaolin Lu, "An investigation on service-oriented architecture for constructing distributed Web GIS application", IEE int. Conf. Services Computing vol. 3, pp. 191-197, 2005

[2] Seong-Kyu Lee, "System Modeling for Web Service based on Service-Oriented Architecture", Journal of the Korea Society for Simulation Vol. 16. No. 1, pp. 49-57. March 2007

[3] David Booth, "Web Services Architecture", http://www.w3.org/TR/ws-arch/, W3C Working Group Note 11 February 2004

[4] Don Box, "Simple Object Access Protocol (SOAP) 1.1", http://www.w3.org/TR/2000/NOTE-SOAP-20000508, W3C Note 08 May 2000

[5] Erik Christensen, "Web Services Description Language (WSDL) 1.1", http://www.w3.org/TR/2001/NOTE-wsdl-20010315, W3C Note 15 March 2001

[6] UDDI, http://uddi.xml.org