Ontology-driven Automatic Geospatial-Processing Modeling based on Web-service Chaining

Liping Di, Peisheng Zhao, Wenli Yang, Peng Yue
Center for Spatial Information Science and System, George Mason University
6301 Ivy Lane, Suite 620
Greenbelt, MD 20770

Abstract- Earth System Science (ESS) research and applications often involve in collecting, analyzing and modeling with distributed heterogeneous geospatial data. Those data are processed step-by-step in geospatial analysis systems to extract information and knowledge products for applications and decision makings. Conceptually, such a step-by-step process forms a geospatial processing model that represents the knowledge of geospatial domain experts. This paper presents a study on ontology-driven automatic creation and execution of geospatial processing models in GeoBrain, a Web-service based geospatial knowledge system, to produce user-specific products. Web Services and Service-Oriented Architecture (SOA) provide a framework to support interoperable machine-to-machine interaction over a network. Web service chaining aimed to solve complex application tasks is changing the way of developing and deploying applications. By wrapping data and processes with Web services, it is easy to transform a geospatial processing model into a service chain. From design (knowledge), instantiation (information) to execution (data), this paper illustrates the whole life cycle of the geospatial-processing modeling and relevant implementation in GeoBrain. Ontology is usually used to capture domain knowledge. In this paper, a number of ontologies, including geospatial scientific ontology, geospatial data ontology and geospatial processing ontology, are introduced as the knowledge base to present geospatial domain terms and concepts, linkage between concepts and datasets, relationships among heterogeneous data, and associations between processes and data. By these ontologies, data and processes can be used for more effective discovery, automation, integration, and reuse across multiple diverse applications.

I INTRODUCTION

Earth System Science (ESS) research and applications often involve in collecting, analyzing and modeling with a huge amount of multi-source, multi-scale and multi-discipline geospatial data [1]. Those data are processed step-by-step in geospatial analysis systems to extract information and knowledge products for applications and decision makings. Conceptually, the step-by-step processes from the raw data to a user-specific product form a geospatial processing model. The model represents the knowledge of geospatial domain experts on how to produce an application-specific product from available raw data sources [2]. Currently, only a few trained people have such knowledge. As the result, the use of geospatial data is largely limited by the availability of such professionals. This situation has significantly hampered the wide use of geospatial data for societal benefits.

It is well known that more and more geospatial data and processes are available online as services. This paper presents a study on automatic creation and execution of geospatial processing models based on users' product specifications in GeoBrain [3][4], a Web-service based geospatial knowledge system, to produce the user-specific products. The whole life of the modeling process, including model design (knowledge capture), model instantiation (information fusion) and model execution (data generation), is implemented based on the semantic and syntactic interoperability between data and processes. Ontology as "*specification of a conceptualization*" [5] is often used to capture domain knowledge explicitly to achieve semantic interoperability. By defining geospatial domain terms and concepts, linkage between concepts and datasets, relationships among heterogeneous data, and associations between processes and data in ontologies, data and processes can be used for more effective discovery, automation, integration, and reuse across various applications**.** Thus, the automatic creation of geospatial processing models can be driven by the knowledge represented in geospatial and application-specific ontologies [6] [7]. According to the World-wide Web Consortium (W3C), a Web service is a software system designed to support interoperable machine-to-machine interaction over a network [8]. The Service-Oriented Architecture (SOA) provides a framework to support the service discovery and invocation in a standardized way.

Web service chaining aimed to solve complex application tasks is changing the way of developing and deploying applications. By wrapping data and processes with Web services, it is easy to transform a geospatial processing model into a service chain (instantiation) and execute it regardless their syntactic heterogeneity.

The reminder of this paper is organized as follows. In section 2, we define the geospatial processing model. In section 3, we discuss the ontology-based knowledge base. In section 4, we discuss the catalog service. In section 5, we discuss the life cycle of modeling and relevant implementation. And finally in section 6, we present the conclusions and future work.

II GEOSPATIAL PROCESSING MODEL

A geospatial process transforms geospatial data from one state to another state. Generally, geospatial process can be classified as 1) atomic process, which runs independently, and 2) composite process, which consists of a sequence of processes in a predefined pattern. A geospatial processing model is a composite process.

A geospatial process is defined by its inputs, outputs and operation. The concepts and relationships among these characters are specified in ontologies. Each process has two types of inputs representing the initial state: conditional inputs (specifying the preconditions and the relationship conditions between the inputs and the operation, such as data format and spatial reference system) and data inputs (specifying the actual data required by the operation, such as data URL). The operation uses a certain geo-processing algorithm to transform the state of data inputs in conjunction with conditional inputs. The outputs usually represent a new geospatial data product (i.e., the goal state), including actual data and its metadata. Following is an example of ISODATA image classification process:

```
<process:AtomicProcess rdf:ID="isodata_cls_process">
  <!-- conditional inputs -- >
  <process:hasInput rdf:resource="#clusters"/>
  <process:hasInput rdf:resource="#iteration_number"/>
  <process:hasInput rdf:resource="#target_data_format"/>
  <!-- data inputs -- >
  <process:hasInput rdf:resource="#source_data_url"/>
  <process:hasInput rdf:resource="#source_data_format"/>
    <!-- outputs -- >
  <process:hasOutput rdf:resource="#target_data_url"/>
</process:AtomicProcess>
```

A composite process consists of other (atomic or composite) processes by specifying control constructs such as *Sequence* and *If-Then-Else*. One crucial feature of a composite process is its specification on how the outputs of a particular subprocess can be accepted by the inputs of other particular subprocesses, i.e. data matching. Building a composite process usually takes advantage of backwards reasoning, i.e., from the goal state to the initial state. Generally, a number of inputs for subprocesses are required to derive the composite process. These inputs may or may not physically exist and, in the latter case, need further processes to generate them. This process goes on until all information needed for inputs physically exist. At that point, a tree-like composite process is constructed [9]. The composite process is a geospatial processing model in which the components and control structures contain the knowledge of a specific application domain. Following is an example of composite process for landslide susceptibility, which is composed of slope, aspect and landslide atomic processes with sequence and split control structures. And the inputs of landslide process come from the outputs of slope and aspect processes.

```
<process:CompositeProcess rdf:ID="landslide_susceptibility_proc">
   <process:composedOf>
     <process:Sequence>
      <process:components>
       <process:Perform>
         <process:Split>
          <process:Perform rdf:ID=”proc_1”>
            <process:process rdf:resource="#slope_proc"/>
          </process:Perform>
          <process:Perform Perform rdf:ID=”proc_2>
            <process:process rdf:resource="#aspect_proc"/>
          </process:Perform>
         </process:Split>
       </process:Perform>
        <process:Perform>
         <process:process rdf:resource="#landslide_proc"/>
          <process:hasDataFrom>
           <process:Binding>
            <process:theParam rdf:resource="#slope_data"/>
             <process:valueSource>
              <process:ValueOf>
               <process:theParam rdf:resource="#O11"/>
                 <process:fromProcess rdf:resource="#proc_1"/>
               </process:ValueOf>
              </process:valueSource>
             </process:Binding>
            </process:hasDataFrom>
            <process:hasDataFrom>
             <process:Binding>
```

```
             <process:theParam rdf:resource="#aspect_data"/>
              <process:valueSource>
               <process:ValueOf>
                <process:theParam rdf:resource="#O11"/>
                 <process:fromProcess rdf:resource="#proc_2"/>
              </process:ValueOf>
             </process:valueSource>
            </process:Binding>
           </process:hasDataFrom>
         </process:Perform>
       </process:Components>
     </process:Sequence>
   </process:ComposedOf>
</process:CompositeProcess>
```

III ONTOLOGY-BASED KNOWLEDGE BASE

Knowledge base provides the overall knowledge of the geospatial-processing modeling. In this project, a set of ontologies are used to capture geospatial domain knowledge, i.e. domain terms and concepts, linkage between concepts and datasets and associations between processes and data. The use of ontologies gives well-defined semantic meaning for the diverse data sources and geo-processing services. Thus, the ontology-based knowledge base can help users to efficiently find the best solution and the most appropriate data. Figure 1 shows the ontologies for geospatial- processing modeling.

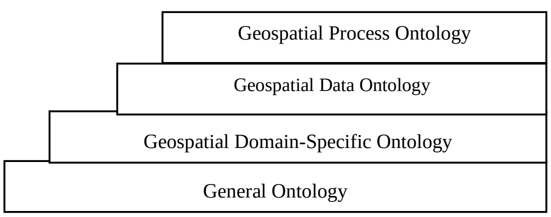

Fig. 1. Ontologies for Geospatial Processing Modeling

The general ontology is the core upper level vocabulary for describing general concepts independent of domain. It is a common language that all other ontologies must reference. The Dublin Core Metadata and OpenCyc are used as the basis to define upper level concepts and assertions about these concepts.

Geospatial domain ontology provides the core conceptualization and knowledge structure of geospatial domain. It describes the problem space which geospatial processing models represent. For example, “Erosion Sedimentation” belongs to “Land Surface”, and “Landslide” is a kind of “Erosion Sedimentation”. Other ontologies directly or indirectly incorporate geospatial domain ontology. We use the SWEET ontology [10], which provides an upper-level semantic description of Earth system science, as a starting point to reorganize and expand the geospatial knowledge. By incorporating the terms in the Global Change Master Directory (GCMD), the Earth Science Modeling Framework (ESMF) and the he Alexandria Digital Library (ADL) Feature Type Thesaurus, geospatial domain ontology covers the knowledge about 1) spatial-temporal factors, e.g. location, time and unit, 2) physical facts, e.g. physical phenomena, physical properties and physical substances, 3) disciplines, e.g. scientific domains and projects, and 4) data platforms, e.g. instruments and sensors.

Geospatial data ontology conceptualizes geospatial data into “DataType” by providing the scientific meanings to the distributed heterogeneous data resources. It links the data with scientific disciplines directly through incorporation of the domain ontology. NASA has used ECS metadata to describe data in NASA data centers. There are also some other metadata standards widely used, such as ISO 19115 and FGCDC. Geospatial data ontology also adds the semantics into the metadata that allows a user to locate data without knowing the exact metadata keywords used by NASA because the terms in query have an equivalent definition in the geospatial domain ontology. To provide a unified view of metadata, the semantic relationships among terms in different metadata standards are defined, such as “disjoint” and “sameAs”. Thus, there is no distinct boundary across various metadata standards. User can use any term from any one of the metadata standards to query the data described in any one of the ontology-supported metadata standards.

Geospatial process ontology provides a reference model to conceptualize different kinds of geospatial processes. It directly incorporates the geospatial domain ontology with geospatial data ontology to associate the processes with scientific problems and relevant data sources. A geospatial process ontology oriented toward the research themes in NASA Earth-Sun system is being developed in this project. This ontology represents the features of the available geospatial processes, classify their internal structure, and document the relationships and the constraints among them, by incorporating following important concepts:

- Scientific discipline: the domain that a geospatial process can be applied, such as solar irradiance and land surface. The definition of the domain is from the geospatial domain ontology.
- Methodology/Algorithm: the type of methodologies and algorithms used in the data mining process, such as ISODATA and MINDISTANCE image classification.
- Data Input: the type of data and its sources the process can work on. The data type is defined in the data ontology.
- Data Output: the type and properties of output of a process, such as running time and accuracy. The data type is from the data ontology.

To some extent, a geospatial process is similar as an abstract Web service without liking to a specific service implementation. Therefore, a geospatial process is treated as a "*service type*" in our approach. We adopt OWL-S specification to describe geospatial processes. Therefore, automatic discovery and composition of all registered processes conforming to the OWL-S specification become possible.

Since all of the ontologies are represented by OWL, the inference engine in the knowledgebase is an OWL reasoner built on Prolog. Ontological information written in OWL or OWL-S is converted into RDF triples and loaded into the knowledgebase. The engine has built-in axioms for OWL inference rules. These axioms are applied to facts in the knowledgebase to find all relevant entailments such as the inheritance relation between classes that may be not directly in the subclass relationships.

IV CATALOG SERVICE

A catalog service plays a 'directory' role in helping providers to describe and advertise the resources availability by using meta-information, and requestors to discover the right resources by querying meta-information. From the design phase to the execution phase, catalog service is very important for each step of the geospatial-processing modeling to discover the types of service and data and their relevant instances.

Currently, there are two prominent general models for registry services: the Electronic Business Registry Information Model (ebRIM) [11] and the Universal Discovery Description and Integration (UDDI) model [12]. For the geospatial domain, the ebRIM is more general and extensible by providing comprehensive facilities to manage metadata based on the ISO 11179 set of standards. To classify, register, describe, discover and access geospatial information, we implement an Open Geospatial Consortium (OGC) Catalog Service for Web (CSW) [13], an ebRIM profile for Web-based geospatial catalog service. Figure 2 shows its information model.

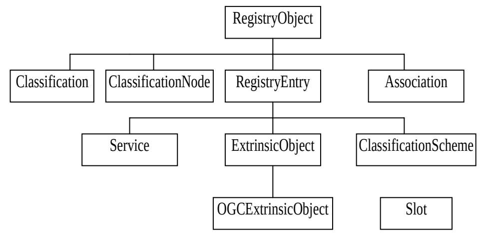

Fig. 2. OGCRIM Class Hierarchy.

The "*ClassificationScheme*" class defines a tree structure made up of "*ClassificationNode*"s to describe a structured way for classifying or categorizing "*RegistryObject*"s: service type and data type. The "*Association*" class uses an "*associationType*" attribute to identify the relationship between service and data. To enable the semantic search capability in CSW, "Slot" and "Association" are used to record corresponding OWL classes, properties and related axioms such as "*subClassOf*". By incorporating the ontologies in knowledgebase, our CSW supports flexible semantic matching.

Let's assume that the output of a geospatial service type is G_o and the request is R, the "flexible semantic matching" can deal with all of the following cases: 1) exact, if $G_o = R$, then G_o and R are exactly equivalent; 2) plug in, if G_o subsumes R than G_o could be plugged in place of R; 3) subsume, if R subsumes G_o, then G_o just completes part of R and R needs other G_o to implement the other part of R or whole R; 4) fail, there is no relationship between G_o and R. This matching algorithm requires the supports from process ontology and domain ontology. The process ontology modulates the structure of processes and indicates the relationships between process and data, while domain ontology plays the role of meta-ontology which indicates the relationships between the terms used in process ontology. By using the ontology, the matching process can perform inference on the subsumption hierarchy to get the recognition of semantic matches regardless of syntactic differences.

V. GEOSPATIAL- PROCESSING MODELING

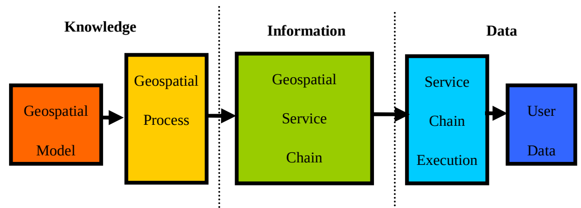

Fig. 3. Life cycle of geospatial processing modeling

As shown in figure 3, the life cycle of geospatial-processing modeling includes 3 phases: 1) knowledge phase for building a geospatial-processing model by composing a composite geospatial process, 2) information phase for instantiating a geospatial process into a geospatial service chain, and 3) data phase for executing a geospatial service chain to generate the geospatial data. A set of tools, including ontology-based knowledge base, model designer, catalog service, virtual data service and data fusion service, and BPELPower (service chain engine), are used for the automation of geospatial processing modeling. The "automation" means automatic design, instantiation and execution without or with least user intervention. Figure 4 shows the relationship of toolkit components.

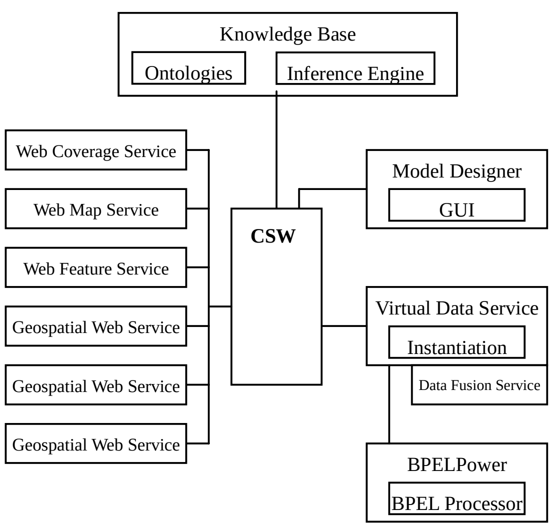

Fig. 4. A toolkit for geospatial processing modeling

A. Knowledge phase

We aim at realizing three levels of automation of model construction similar as of service chain defined in [14].

- **User-defined (transparent):** the user queries the catalog service with more specific details on the different geospatial service types to define and manage the model.
- **Workflow-managed (translucent):** the user queries the catalog service for a given problem, and then the knowledgebase assists the user to select and configure the most suitable geospatial service types in each step of model construction.
- **Aggregate (opaque):** the user presents a problem, and then the knowledgebase incorporates the catalog service to build a geospatial model with the best geospatial service type without user's intervention.

Currently we have implemented the "transparent" and the "translucent" in the GeoBrain model designer. The "opaque" is being developed. The model designer provides a graphic user interface allowing user to drag and drop "data type" and "service type" to build the model. Figure 5 shows using the model designer to build landslide susceptibility model. The left column shows the "data type" and "service type" registered in the catalog service, and the right column is the model graphic representation. The whole process is as following:

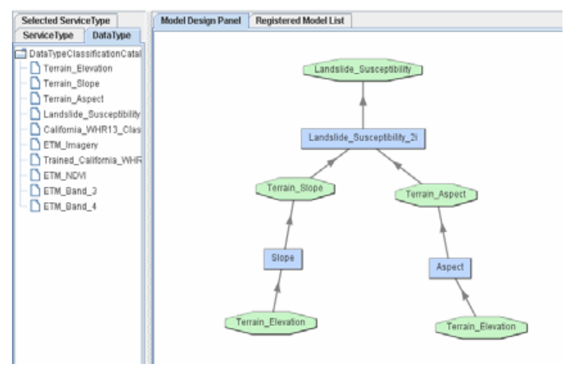

Fig. 5. Model designer

1. Select "Landslide_Susceptibility" data type, i.e. the goal state.
2. Find a service type whose output type is "Landslide_Susceptibility". This is done automatically by the designer. Only the satisfied service types are listed and selectable so that match errors between service type and data type are impossible. The designer finds the "Landslide_Susceptibility_2i" service type that can generate the output of "Landslide_Susceptibility."
3. Select "Landslide_Susceptibility _2i" service type whose input data types are "Terrain_Slope" and "Terrain_Aspect". If there are more than

one satisfied services, the designer allows user to view their metadata to assist selection.

4. Find service types whose output data types are “Terrain_Slope” and “Terrain_Aspect” as step 2. Select “Slope” and “Aspect’’ service type whose input data type is “Terrain_Elevation”.

Note the selection loop can be terminated manually by user at any step or automatically if there is no service type that can output the required data types.

Once the model is created, it can be registered in the catalog service as a service type for future use. It has its own inputs, outputs, spatial and temporal scopes like other service type. The distinction is it has its own composite process description in XML just like the example given in Section 2.

Ontology plays a very important role in the model building process. It suggests what to do and what to use on the basis of semantic matching, e.g. locating data types on a specific topic and finding service types for a desired data type, a particular method, or a specified geospatial scientific task.

B. Information phase

In this phase, a geospatial model is instantiated into a geospatial Web service chain with registered service instance information. Such a service chain represents the information of how to derive the exact data product. A virtual data service is implemented to fulfill this phase:

1. Service discovery. Since every service instance registered in the catalog service has an association to service types, it is easy to find a service instance for each service type in the geospatial model. If there are more than one service instance available, the selection depends on the quality of service (QoS). Of course, the matching level of services and data should be considered first in the following sequence of relevance: exact > plug in > subsume. The other functional factors and conditions also should be considered, such as accuracy, time, data format and data projection. If no service instance is discovered, this phase will be failed and stop here.

2. Data discovery and fusion. In a geospatial model, there is no indication of who provides the inputs to the services in the model leaves. With the help of catalog service, the virtual data service automatically adds a relevant data service instance, which provides such input data, at the beginning of service chain. If the outputs and inputs of adjacent services are heterogeneous on data format and data projection, some data fusion services are also added into the service chain automatically to deal with these heterogeneities. Examples of such data fusion services include Web Coordinate Transformation Service (WCTS) and Data Format Translation Service.

3. Representation of service chains. The representation of service chain is critical to its materialization and reuse. Some industry initiatives have been developed to address the needs that coordinate the sequencing and execution of services. We adopt the widely used Business Process Execution Language for Web Services (BPEL4WS) [15], a language for the formal specification of business processes and business interaction protocols, to represent service chain. Although the BPEL4WS is initiated for business process, our experiment shows it still can cover all the requirements of scientific processes.

C. Data phase

In this phase, a geospatial service chain is executed to derive the desired data products. For this purpose, we have developed the BPELPower, a service chain engine based on the mainstream standards, such as BPEL, WSDL, WSIF, UDDI, SOAP, JNDI and J2EE. It can run on the top of popular application servers, such as Tomcat, JBoss, Weblogic and WebSphere. Figure 6 shows its user interface. WSDL-based web services and BPLE-based web services chain can be deployed and executed dynamically in BPELPower, where their validations are checked. Different invocations (e.g., HTTP POST/GET, SOAP document/rpc, etc.) are well supported.

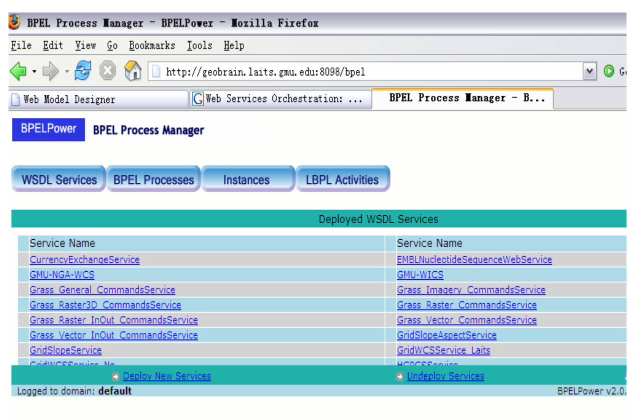

Fig. 6. BPELPower – service chain engine

VI Conclusion

This paper presents a three-phase approach, i.e. knowledge (design), information (instantiation) and data (execution), to the automation of geospatial-processing modeling. The basic idea of this approach is to use Web service chain to represent geospatial-processing models. The Web service chaining technology provides interoperable framework on discovery, orchestration and invocation. The most significant distinction of this approach is the use of ontologies (including upper level ontology, geospatial domain ontology, geospatial data ontology and geospatial process ontology) to capture and represent geospatial domain knowledge and classify geospatial data and services (processes) in order to achieve the automation through semantic inference and matching. To facilitate the automation in each phase of modeling, we develop a toolkit, including OGC CSW for information discovery, model designer for model design, virtual data service for model instantiation, and BPELPower for model execution.

"Service Type" and "Data Type" are the core concepts in this approach. It is critical for us to have a full and correct classification schema for geospatial data and processes. The standardization efforts from the GCMD, ISO19115 and OGC seem not enough yet. In the next step, we will investigate more existing geospatial ontologies and standards to sketch geospatial domain precisely and elaborate the relationships inherent in the nature of geospatial data and processes.

The "opaque" modeling is being developed, which builds the model without user intervention by using artificial intelligence (AI) technologies and ontology reasoning. Many efforts on automating the Web service composition problem using AI planning have been reported in [16] [17]. We will investigate AI planning to enhance the capability of automatic modeling.

REFERENCES

[1] L. Di, 2004. Distributed Geospatial Information Services-Architectures, Standards, and Research Issues. *The International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, Vol. XXXV, Part 2, Commission II, 7p. (Invited).

[2] L. Di, 2005. "Customizable Virtual Geospatial Products at Web/Grid Service Environment." Proceedings of IGARSS 2005. July 25-29, Seoul, South Korea. 4p. (Invited).

[3] L. Di, 2004. GeoBrain-A Web Services based Geospatial Knowledge Building System. Proceedings of NASA Earth Science Technology Conference 2004. June 22-24, 2004. Palo Alto, CA, USA. (8 pages. CD-ROM).

[4] L. Di, 2005. "The Implementation of Geospatial Web Services at GeoBrain." Proceedings of 2005 NASA Earth Science Technology Conference. June 28-30, 2005. College Park, MD. 7p. (CD-ROM).

[5] T. Gruber, "A translation approach to portable ontologies", in *Knowledge Acquisition*, Vol. 5, Issue 2, 1993, pp. 199-220.

[6] L. Di, 2005. "A Framework for Developing Web-Service-Based Intelligent Geospatial Knowledge Systems," Journal of Geographic Information Sciences. Vol. 11, no 1, pp. 24-28.

[7] L. Di, P. Zhao, W. Yang, G. Yu, and P.Yue, 2005. "Intelligent Geospatial Web Services." .Proceedings of IGARSS 2005. July 25-29, Seoul, South Korea. 4p. (Invited).

[8] W3C, "Web Services Architecture", 2004. http://www.w3.org/TR/ws-arch/

[9] L. Di, 2005. "Customizable Virtual Geospatial Products at Web/Grid Service Environment." Proceedings of IGARSS 2005. July 25-29, Seoul, South Korea. 4p. (Invited).

[10] R. Raskin, "Enabling Semantic Interoperability for Earth Science Data". http://sweet.jpl.nasa.gov/EnablingFinal.doc

[11] OASIS, "OASIS/ebXML registry information model v2.0", 2002. http://www.oasisopen.org/committees/regrep/2.0/specs/ebRIM.pdf.

[12] OASIS, "The UDDI technical white paper", 2000. http://uddi.org/pubs/uddi-tech-wp.pdf

[13] R. Martell, ed., 2004. "OGC™ Catalogue Services–ebRIM(ISO/TS 15000-3) profile of CSW", 2004.

[14] ISO/TC 211, "Geographic Information – Services, ISO/DIS 19119", 2001.

[15] T. Andrews, ed., "Business process execution language for Web services version 1.1", 2003.

[16] J. Rao, X. Su, "A Survey of Automated Web Service Composition Methods". In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004*", San Diego, California, USA, 2004.

[17] J. Peer, "Web Service Composition as AI Planning - a Survey". University of St.Gallen, Switzerland, 2005.