Ontological and Pragmatic Knowledge Management for Web service Composition

Soon Ae Chun[1], Yugyung Lee[2], James Geller[3]

[1] CDS Dept, Seton Hall University, NJ
chunsoon@shu.edu
[2] CS Dept, University of Missouri, Kansas City
leeyu@umkc.edu
[3] CS Dept, New Jersey Institute of Technology,Newark, NJ
geller@njit.edu

Abstract The vision of the Semantic Web is to reduce manual discovery and usage of Web resources (documents and services) and to allow intelligent agents to automatically identify these Web resources, integrate them and execute them for achieving the intended goals of the user. The composed Web service is represented as a workflow, called *service flow.* This paper presents different types of *compositional knowledge* required for automatic Web service flow generation: *operational (syntactic), semantic* and *pragmatic* knowledge. Operational knowledge deals with the right output and input types for possible service composition, while semantic knowledge is domain-specific expert knowledge that governs the Web service compositionality. In addition, pragmatic knowledge allows the contextual as well as common sense knowledge to constrain the Web service composition. We show how to extend the current XML-based standards to represent the compositional knowledge or rules that play a role in service discovery and composition.

1. Introduction

When a Web user is performing a complex task, he/she typically needs to identify a set of documents and services, using key word searches. She then needs to extract relevant information from each document and submit service requests that may be entered into Web forms. All these steps have to be performed manually and in the right order. The vision of the Semantic Web is to reduce these manual activities and to allow intelligent agents to automatically identify tasks, integrate them and execute them for achieving the goals of the user [Berners-Lee et al., 2001]. This vision is achievable when Web services and documents are described in a machine-interpretable format. Towards that end, several standards such as WSDL, UDDI, and SOAP have been created. SOAP (Simple Object Access Protocol) is a standard for applications to exchange XML-formatted service descriptions over HTTP. WSDL (Web Service Description Language) describes what a Web service does, where it resides, and how to invoke it. UDDI (Universal Description, Discovery and Integration) is a standard for publishing information about Web services in a global registry. A DAML-based Web Service Ontology (DAML-S) supplies Web service providers with a set of markup language constructs for describing the properties of their Web services [Ankolekar et al., 2001].

Unfortunately, the automatic service composition requires not only these descriptions of service capabilities and properties, but also *compositional knowledge* of how to select and in what order to perform Web services. An autonomous agent should be equipped with this meta-knowledge about composition in order to discover a set of Web services to achieve a user's goal, composing the services in a proper order, and executing them, without or with minimal human intervention. Thus, this paper addresses the problem of what kinds of knowledge are necessary for Web service composition and how we can perform the automatic composition of Web services into a work flow of services, which we call a *service flow*. Depending on the domain, a service flow can be referred to as a commerce flow (com-flow), a government flow (gov-flow) or an education flow (edu-flow). In this paper we do not discuss edu-flow.

In the networked economy, organizations often form virtual enterprises, where business processes cross boundaries of several organizations, including partners, subsidiaries, and even competitors. An inter-organizational com-flow differs from other kinds of service flows, because it involves partners with trade secrets that might be competitors in other arenas. Generating a com-flow requires knowledge about business rules and the commerce environment. This includes common knowledge about products and services typically provided by different companies. For example, Dell provides computers, while eBay provides auction services. It is also necessary to augment this knowledge with an expert's knowledge of government regulations, company policies and common practices, occasionally constraining the composition of certain services. In order to compose individual Web services into service flows, the service capability descriptions need to include not only the semantic descriptions, but also knowledge telling the automatic agents "in what situations a Web service should be used." We call this additional level of description of Web services *pragmatic or contextual knowledge.* A service should be described by *pragmatic annotations* that represent this pragmatic knowledge, in addition to the *semantic annotations* and *syntactic annotations* of the network and service descriptions.

Contributions of this paper: In this paper, we propose a knowledge architecture for the automatic composition of Web services often provided by independent organizations anywhere on the Web. First, the paper presents different types of compositional knowledge expressed as rules that a software agent can use for the automatic generation of a service flow. Second, we present semantic and pragmatic rules ontology for service selection and filtering, extending the current practice of Web service description. The use of the Semantic Web language OWL is proposed as a format for rule description ontology.

The paper is organized as follows. In Section 2, we discuss different types of compositional knowledge, followed in Section 3 by our representations for rules and semantic and context rules ontology. In Section 4, we present an overall system architecture. Relevant work and conclusions are presented in Sections 5 and 6, respectively.

2. Web Service Composition Knowledge

To create a service flow, individual Web services are composed into a workflow, which is greatly facilitated if they are represented with Web service integration languages such as WSCL, WSFL, XLANG, and BPEL4WS. These technologies help to represent the composite service as a workflow, but they do not address the issue of how the service flow can be automatically composed and generated. The standards of WSDL provide syntactic annotations for Web services that include tags mostly limited to network connections, input/output specifications, data types and how they bind to concrete ports. DAML-S provides semantic descriptions of Web services, using an ontology to support users and software agents to discover, invoke, compose and monitor Web services with particular properties. These semantic annotations for Web services equip service providers with a set of markup language constructs for describing the capabilities of their services in computer-interpretable form.

Web services in DAML-S are described with profiles, models, and groundings. The service profile includes contact information of the provider, input and output of the service, and features such as category, quality rating, estimated maximum response time, geographic radius, etc. The service profile provides the type of information about "what the service does." The profile is therefore used for advertising, registry, discovery, and matchmaking for the agent to find the service that meets its purpose.

The service model tells "how the service works," guiding a service-seeking agent to compose a service as a process, using inputs, outputs, preconditions, effects, and subprocesses. This information is used to perform a specific task, to coordinate the activities of the different participants, and to monitor the execution of the service. A service grounding specifies the concrete details of how an agent can access it. Typically, a grounding specifies some communications protocol (e.g., RPC, HTTP-FORM, SOAP), and service-specific details such as port numbers used.

An agent uses these services, described using DAML-S, to achieve the service composition. It looks up the service profile, and then it selects the candidate services that match its goals. Among these candidate services, it then selects a service that matches preconditions, input and output. For example, a virtual store service may require as a precondition a valid credit card, and as input the credit card number and expiration date. As output it generates a receipt, and as effect the card is charged. The selection of this service is constrained by the fact that the agent must be able supply the input credit card number. If that is not the case, the service is not *composable*. The same is also true for the precondition. If the precondition of the service cannot be met, the service even though it has a valid profile for selling, is not composable from the point of view of this particular agent. This knowledge of composition is based on a syntactic constraint, that is, if the input cannot be supplied, then the service is not composable.

Another kind of compositional knowledge is based on semantic constraints. Web service compositionality often depends on the proper ordering of services. For instance, the business registration service provided by the New Jersey State government for a new corporation requires an agent to establish a business entity first, and then perform tax registration for this entity. If this order is not respected, then the two services cannot be composed. This ordering relationship therefore constrains the compositionality of services.

Similarly, in other domains, there are rules and regulations to follow. For instance, according to New Jersey law, any land development close to the coastal area forces the agent to select *the coastal permit application service*, rather than the usual land development service. The selected service is required to be part of the service flow. Thus, the application of semantic constraints (rules) plays a significant role in service compositionality. The semantic composition rules often require domain expert knowledge, such as business law, trade laws, State legislations, federal environmental regulations, formal company policies, etc.

The third type of knowledge necessary for the composition of services, which has been given little attention in the literature, is *contextual or pragmatic knowledge*. Pragmatic knowledge is about the situations in which a service should be used. As such it is not about the service itself, but about the way the service relates to the satisfaction of the goals of the consumer of the service flow. Pragmatic knowledge is therefore intimately connected to the context in which a service flow is executed, which makes it more difficult to formalize. Potentially, pragmatic knowledge touches on a wide range of human endeavors, as the following examples will show.

In real life it is often the case that several services possess the same profile and provide the same functionalities. The automated selection of one service among these functionally compatible services may require pragmatic knowledge. For example, when buying a book offered by several vendors, get it where it is cheapest, unless otherwise specified. This pragmatic rule applies when the companies that are in competition with each other provide the same type of products. However, when otherwise equivalent products carry different brand names, they tend to have higher price tags, which may be acceptable. If the goods need to be shipped before a specific date, then a supplier that is more expensive but offers overnight delivery may be acceptable. This kind of pragmatic knowledge needs to be modeled in order for the agent to automatically select one service over the other. Some of this knowledge is highly personalized, as the perception of brand names is based on individual tastes.

In summary, syntactic knowledge allows an agent to consider all *possible* compositions. Semantic knowledge constrains the agent to make *sensible* compositions that conform to the relevant laws and organizational knowledge. Pragmatic knowledge allows the agent to make *reasonable* compositions with respect to the needs and preferences of the initiator of the agent. In the following section we discuss how to model the three types of compositional knowledge for service flows.

3. Service Composition Model

Web Services and Web Service Flows Web services are described using standard DAML-S ontology notation. Examples of Web services in DAML-S notation can be found in [DAML-S]. A Web service flow consists of individual Web services, serving as component tasks in a workflow. The formal workflow model developed in [Chun, 2003] can be represented in BPEL4WS [BPEL]. Due to space limitations we omit the BPEL4WS representation.

Composition Rules Ontology A computational agent needs to automatically identify component Web services and compose them, using syntactic, semantic and pragmatic knowledge, described in the previous section, given a user's goals, constraints and preferences. For this, we can use Web Ontology Language OWL for a rule representation with conditions and actions, but we organize the rules by using a topic ontology. For example, the knowledge about regulations is modeled as regulatory topic ontology (Figure 1). This topic-subtopic hierarchy allows the identification of necessary information (e.g. type, location and kind of a new business) and regulation information. Each leaf node is associated with a regulatory rule. The rule nodes are modeled as Condition-Action pairs, where a Condition is a logical expression and an Action is an operation on services (e.g., insert into flow, parallel, order). Each rule node may also have a link to services that implement it.

Just as the WSDL allows for an easier identification of services, the XML-based rules are linked to a topic ontology. This hierarchical representation allows rules to be easily identified according to a relevant topic. Once the applicable rules have been identified, these compositional rules allow the selection of proper Web services and the ordering of the services. During service flow generation, the user requirements and preferences are matched against the concepts in the topic ontology. In our domain, the problem is to get all the necessary permits for opening a new business. These permits are offered via Web services of independent government agencies. When a user wants to register a new autobody shop, he might have special needs. If he plans to use over 1/2 gallon of spray-paint per hour, this affects the air quality. In this case, a rule for inserting an air quality permit requirement into the service flow would be activated. The algorithm processing the rules therefore has the information what services to select and what the ordering of the services should be. Figure 2 shows an extension of OWL (Web Ontology Language) with proposed tags to describe the rules and rule ontology[1].

Pragmatic and Contextual Rules We model pragmatic rules used to further select a candidate service among similar services with a *contextual rules ontology*. Unlike in the government domain, in the E-commerce domain, a service may be provided by many companies. A service can be identified, by using an interest-related rule hierarchy, as shown in Figure 1. But there should be more information to select one among many service providers. For this purpose, contextual and pragmatic rules are used. The intelligent agent uses this pragmatic knowledge (rules) to select one of several competing services.

---

[1] RuleML (Rule Markup language) [RuleML] or SWRL (Semantic Web Rule Language that combines OWL and RuleML) can be used.

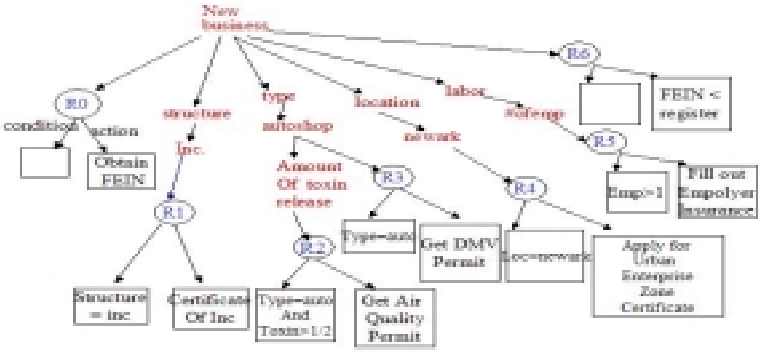

Figure 1: **Topic ontology for government regulations for business registrations**

```
<owl:Class rdf:ID=”AutoBodyShop”>
<refs:subClassOf ref:resource=”BusinessType” />
<refs:subClassOf>
  <owl:RestrictRule ref:ID=”airqualityRule”>
  <owl:Condition ref:datatype=”expr”> SprayPaint > 1/2  gallon per hour </owl:Condidtion>
   <owl:Action refs:resource =”Air quality permit”>
          <owl:Action refs:action =”insert”>
           </owl:Action>
 </owl:Class>
```

Figure 2: Proposed OWL extension for Rules Ontology

These pragmatic rules are organized in a pragmatic rules ontology, as shown in Figure 3 for the domain of travel arrangement services. As shown, the airline reservation service can be provided by several service providers, i.e., Delta or American services. Both of these providers are subject to various regulatory travel-related rules (e.g. if the destination is a country that requires a visa, then check the passport and visa). In addition, these services are subject to the pragmatic (contextual) rules that are specific to airline services, such as that a Saturday night stay implies lower cost, or that people tend to prefer first class seating if there is no price difference, etc. These rules can be overridden by user-specific rules. If a user does not care about the Saturday night discount, then that rule is not used. These kinds of rules are needed for an intelligent agent to select among compatible services for automatic composition. Unlike the regulatory or topic-related rules, these rules order services in common situations.

Similar to the OWL representation of semantic rules, pragmatic knowledge can be represented with OWL, which is not presented due to the space limitation. The difference lies in that the action is in preference form, e.g. action ="choose (or prefer) the service ***s*** for which the cost is lower," rather than in assertive form as in semantic/regulatory rules (e.g. action=”insert ***s***”).

4. System Architecture

The proposed system architecture for a Web service portal is shown in Figure 4. The service flow composition component has three modules. First, the user service request for a composite service is processed by the composition agent. This discovery agent consults the rules topic ontology to identify which component services are needed. The component services identified by semantic rules often can be several by various service providers with similar functionalities.

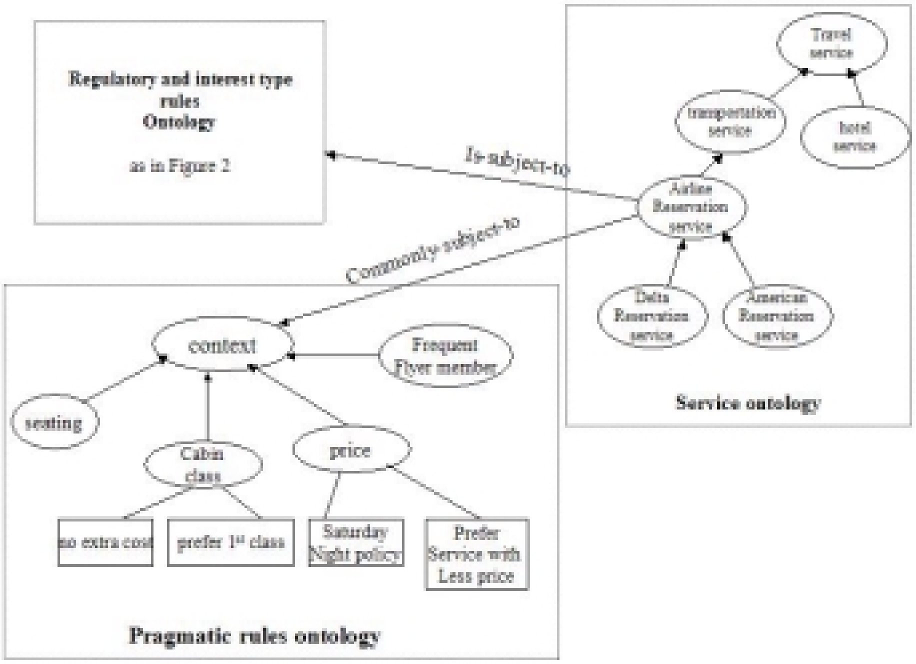

Figure 3: **Pragmatic rules ontology**

These candidate services are returned to the service selection agent, which further consults the pragmatic rules to filter, compare and eventually select the most appropriate service. The selected service is returned to the composition agent. This process is repeated until the composed service flow meets the user request. The composed Web services can be executed and monitored by the service flow execution engine. Prototype systems for the e-government and medical domains are currently under development.

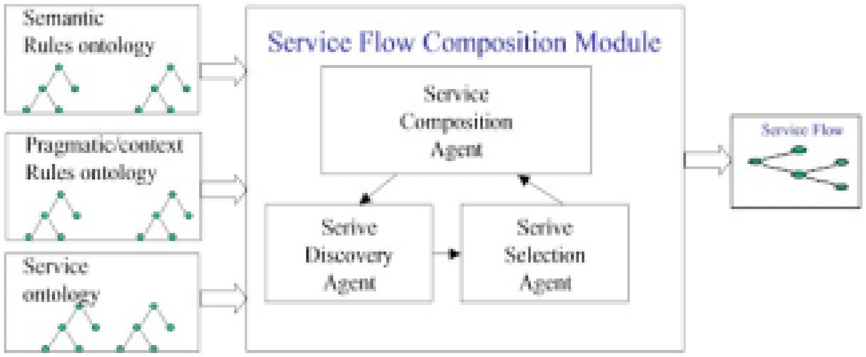

Figure 4: System Architecture for Service Flow Composition

5. Related Work

The Semantic Web research work, following the DARPA Agent Markup Language (DAML), includes DAML+OIL [Hendler and McGuinness, 2001] for the creation of arbitrary domain ontologies and DAML+OIL/RDF(S) for the semantic mediation between services and workflow logic. Automatic composition of Web services [McIlraith et al., 2001] has been achieved through automated mapping, composition and interoperation of services, service verification, and execution monitoring. There are other emerging relevant approaches such as indexing services based on process models [Klein and Bernstein, 2001] and reasoning about service descriptions in choosing computational resources [Raman et al., 1999]. Scientific workflow [Cardoso and Sheth, 2002] is supposed to support interoperation through semantics. It may have the potential to support Web service descriptions for service discovery, invocation and execution of an identified service by an agent or other service [McIlraith et al., 2001].

Unlike these efforts, our approach emphasizes the importance of different kinds of knowledge, especially pragmatic/context knowledge, for the automatic composition of service workflows. [Chun et al 2002a; Chun et al., 2002b; Chun, 2003] have developed an automatic workflow composition algorithm and prototype in the E-government domain, using a government regulatory rules ontology for government E-service workflow composition. These have not addressed the pragmatic and contextual knowledge often used in the selection of alternative services.

6. Conclusions and Future Work

Many real-life business processes have to live by various rules and constraints. In this paper, we have classified these rules into *syntactic, semantic and pragmatic (context) composition rules* that play a major role in the Web service composition. These rules incorporate the knowledge that is necessary to discover, select and compose Web services into a coherent service flow. We have proposed an approach to modeling the composition knowledge with rules ontology using OWL, DAML-S standards. The model of commonly assumed knowledge in a domain as pragmatic rules with ontology concepts is proposed and used for the service filtering and selection. Future work includes the extension of compositional knowledge to include negotiation rules. When a topic in the process of service selection does not exactly meet the conditions of a rule, then there should be a possibility to relax it. This is best modeled as a form of agent negotiation. We are also planning to work on a user service request model that better supports personalization for the targeted selection of services and composition rules.

References

[Ankolekar et al 2001] A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S: Semantic markup for Web services. In Proceedings of the First International Semantic Web Working Symposium (SWWS), pp. 411-430, Stanford, 2001.

[Berners-Lee et al., 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, 284(5), pp. 34-43, May 2001.

[BPEL4WS] BPEL4WS Website: http://www-106.ibm.com/developerworks/library/ws-bpel/

[Cardoso and Sheth, 2003] J. Cardoso and A. Sheth. Semantic e-workflow composition. Technical report, LSDIS Lab, Computer Science, University of Georgia, July 2002.

[Chun et al., 2000a] S. A. Chun, V. Atluri and N. R. Adam, Domain Knowledge-based Automatic Workflow Generation, in the proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002), September 2-6, 2002, Aix en Provence, France.

[Chun et al., 2000b] S. A. Chun, V. Atluri, and N. R. Adam, Dynamic Composition of Workflows for Customized eGovernment Service Delivery, in the proceedings of The Second National Conference on Digital Government (dg.o 2002), May 19-22, 2002, LA, CA.

[Chun, 2003] Soon Ae Chun, Decentralized Management of Dynamic and Customized Workflows, Ph.D. Dissertation, Rutgers University, 2003.

[DAML-S] DAML-S: http://www.daml.org/services/daml-s/0.7/CongoService.daml

[Hendler and McGuinness, 2001] J. Hendler and D. L. McGuinness. DARPA Agent Markup Language. IEEE Intelligent Systems, 15(6):72-73, 2001.

[Klein and Bernstein, 2001] M. Klein and A. Bernstein. Searching for services on the semantic web using process ontologies. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, July 2001.

[McIlraith et al., 2001] S. McIlraith, T. Son, and H. Zeng. Semantic Web services. *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, 16(2):46-53, 2001.

[Raman et al., 1999] R. Raman, M. Livny, and M. Solomon. Matchmaking: An extensible framework for distributed resource management. Cluster Computing: The Journal of Networks, Software Tools and Applications, 2:129-138, 1999.

[RuleML] RuleML Website: http://userpages.umbc.edu/~mgandh1/2002/06/DamlRuleML/