Towards Intelligent Web Services for Automating Medical Service Composition

Yugyung Lee, Chintan Patel
School of Computing and Engineering
University of Missouri - Kansas City
{leeyu, copdk4}@umkc.edu

Soon Ae Chun
DS Dept
Seton Hall University
chunsoon@shu.edu

James Geller
CS Dept
New Jersey Institute of Technology
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 software agents to automatically identify these Web resources, integrate them and execute them for achieving the intended goals of the user. Such a composed Web service may be represented as a workflow, called service flow. Current studies of Web services are not sufficient for automatic composition. This paper presents different types of compositional knowledge required for Web service discovery and composition: syntactic, semantic and pragmatic knowledge. As a proof of concept, we have implemented our framework in a cardiovascular domain which requires advanced service discovery and composition across heterogeneous platforms of multiple organizations. Within this framework, we describe (1) How to represent the compositional knowledge, which plays a role in service discovery and composition, in DAML-S; (2) How heterogeneous medical services interoperate in a composed medical service flow. (3) To solve this knowledge level integration, we build on an ontology integration method called SEMIO (Semantic Interoperability).

1. Introduction

In the past few years, the WWW has changed from being nothing more than an indexed repository of documents towards being a repository of interconnected services and documents. Web users are now routinely checking the Web for services such as currency converters, mortgage calculators, shortest driving distance with directions generators, etc. Unfortunately, not every required service is available on the Web, and if it is, it might be hidden at position 1921 of 2000 search engine hits. Therefore Web research has turned to the time-honored approach of its parent discipline and attempts to provide complex services by, in effect, combining simple services in the way of a workflow of services, what we call a *service flow*. However, the problem of creating a service flow for a given specification is difficult, and it is a part of the vision of the Semantic Web [1] to let roaming agents perform this difficult task. For that purpose, (simple) services need to be described in an agent-readable form.

The automatic composition of services requires more than descriptions of service capabilities and input/output parameters. Rather, a service should also indicate in what situations and in what ways it should be used. This is comparable to the manual of an electronic device that provides a service. For example, a cell phone manual describes "use cases" of the services that the cell phone offers: Making phone calls, maintaining a calendar, etc. In case of an emergency, most cell phones allow a 911 call without the payment of a fee. While it is obvious that this kind of knowledge needs to be provided and bundled with the device itself, it is only recently becoming clear that Web services need to have the same kind of knowledge attached to them.

We call this additional level of description of Web services *pragmatic* or contextual knowledge. A service should be described by a pragmatic annotation that represents this pragmatic knowledge, in addition to the semantic and syntactic knowledge that describes the necessary parameters and functionalities of the service. We propose an ontology as a model for representing knowledge to describe services. Specifically, we use ontologies to represent syntactic, semantic and pragmatic knowledge about services.

Clearly, the service composition faces an immediate problem when every service is described using terms from its own underlying domain. The pragmatic and semantic knowledge ontology may contain a collection of these terms

[1]. Therefore, the discovery of correct component Web services will often require additional preliminary steps to integrate the ontologies used to describe these Web services. In many cases it will be necessary to integrate the ontology of an agent, searching for a service, with an ontology describing a service. This will have to be done on the fly and at great speed to decide whether a specific service is a possible candidate for the desired service flow.

Thus, this paper addresses three problems: (1) How do we define, distinguish between, and justify the need for three different kinds of knowledge to be used in service descriptions: syntactic knowledge, semantic knowledge and pragmatic knowledge. We propose a model to represent Web service compositional knowledge; (2) How do heterogeneous medical services interoperate in a medical service flow, composed using these three kinds of knowledge; and (3) How can we perform ontology integration to enable the automatic composition of Web services into a service flow. In this paper, we present an architecture how a service flow can be automatically composed using syntactic, semantic and pragmatic knowledge. We demonstrate how heterogeneous Web services can be made interoperable within our framework. Our motivating examples are drawn from medical services, which should be called, in our case, "Semantic Medical Services (SMS)."

The three different types of compositional knowledge are expressed by compositional rules that a software agent can use for the automatic generation of a service flow. We present an ontology for these compositional rules, applying them to the description of Web services [4]. The paper also illustrates one approach how to integrate terms from several ontologies in an efficient manner, using the framework of Terminological Knowledge Bases [5].

2. Semantic Medical Services

In medical research, there is a need to exchange valuable information between different researchers or research groups. Different research groups rely on disparate technical infrastructures to carry out their work for a particular study of interest. There is an increasing need for information exchange as well as service sharing in a unified manner within and between healthcare organizations.

Information sharing through the Web demands collaboration in a diverse community of healthcare organizations. For example, at the Mid-America Heart Institute (MAHI), there is an on-going effort to share clinical data, services and results among related medical institutes such as Children's Mercy Hospital (CMH), Saint Luke's Hospital (SLH) and Kansas City Area Life Sciences Institute. These activities require exchanging large volumes of cardiovascular clinical data including administrative data, procedural data, various clinical studies and office-based records generated by heterogeneous organizations.

In real world settings, it is difficult to build automated procedures for resource sharing (e.g., patient records or images) and analysis across organizations, which follow different data models and document structures. The nature of today's healthcare enterprises of being distributed boosts the heterogeneity inherently existing in different healthcare organizations. Most existing health care infrastructures are based on legacy components. Thus, interoperability across such services from different organizations becomes more difficult. The integration of real-time data is even more challenging. We need to have some common "data medium" for information interchange between the applications so that heterogeneous data can be easily converted into formats understandable to respective applications.

As practical approaches to resolving these issues, work flow solutions have been successfully implemented in many health care enterprises[1]. Workflow technology offers several advantages, including automation and streamlining of processes and significant cost reductions (administrative and human resources) [16]. Also, ongoing research efforts (Health Level 7 (HL7) and Digital Imaging and Communications in Medicine (DICOM)) provide standards for the exchange, management and integration of medical resources that support clinical patient care and the delivery and evaluation of healthcare services.

Figure 1 shows a view of a medical service flow that is implemented at CMH, University of Missouri (UMKC), and SLH in Kansas City. As depicted in the Figure 1, consider a scenario where a Cardiologists wants to study some patient's catheterization report and heart diagrams using the M-mode technique. The problem is that the required information is distributed across several organizations. Catheterization is performed at MAHI, where the Lab Information Systems stores the relevant information. But, the Master Patient Index is maintained in the Hospital Information System at CMH, where the radiology department also stores the DICOM images. In addition, there are Web services available at UMKC, providing patient record validation and other services like medical code lookup and image retrieval. The challenge is to dynamically combine different services in a pragmatic manner to solve a given high level goal.

We are currently working on developing modularized Semantic Web service wrappers for legacy medical applications. The goal is to abstract the common medical applications and services in the hospital environment, thereby facilitating dynamic composition and flexible management of service flows. Also, we are in the process of modelling services and service parameters using medical ontologies. The idea is to decouple applications and allow seamless integra-

---

[1] http://www.e-workflow.org/case_studies/healthcare/

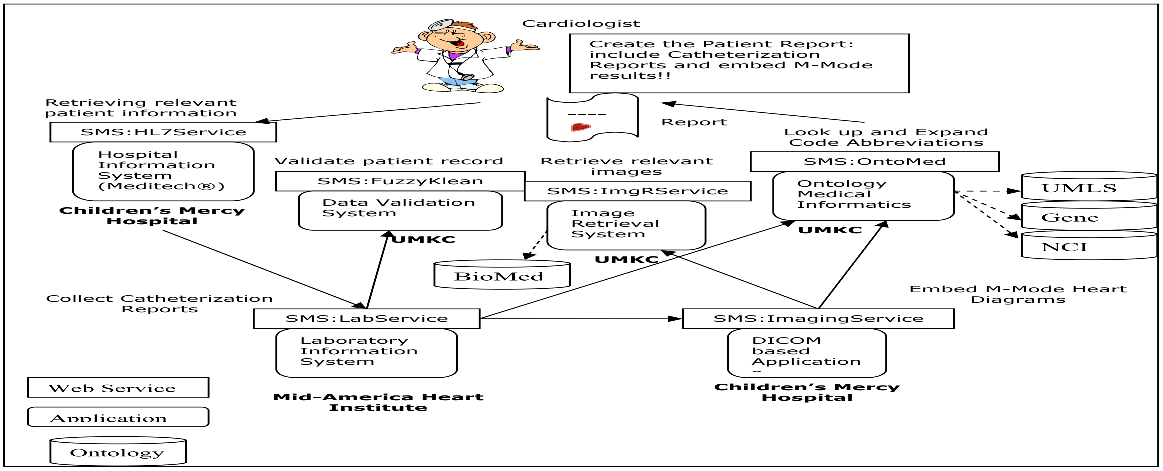


Figure 1. A Semantic Medical Service Flow

tion of applications over the Web. We have observed that there exist several medical vocabularies readily mappable to many of the service parameters we needed to model. We present a sample of modules, service parameters and corresponding mappings to ontologies in Section 5.

3. Service Composition Model

3.1. Service Compositional Knowledge

We will discuss three kinds of compositional knowledge. The first kind of compositional knowledge is based on a syntactic constraint, that is, if the input cannot be supplied, then the service is not composable. Whenever the syntactic constraints on compositionality cannot be met, then a service is not composable and cannot be selected. For example, a clinical service may require as a precondition a valid patient ID, and as input the patient demographic information and insurance information. As output it generates an order, and as effect the relevant patient data are collected and transferred to a subsequent department. The selection of this service is constrained by the fact that the agent must be able to supply the input patient ID. If that is not the case, the service cannot be used by this agent, i.e., it is not composable. The same is true for every precondition. If a precondition of the service cannot be met, the service is not composable from the point of view of this particular agent.

The second kind of compositional knowledge is based on semantic constraints. Web service compositionality often depends on the proper order of services. For instance, the health care service provided by a hospital for a new patient requires an agent to admit the patient first, and then perform scheduling for this patient. If this order is not respected, the two services (admission and scheduling) cannot be composed. This ordering relationship therefore constrains the compositionality of services. Similarly, in other domains, there are rules and regulations to follow. These rules and regulations mandate service compositionality in certain predetermined ways. Another example of semantic compositionality is when a specialized service of a hospital (e.g., heart) mandates the selection of a certain corequisite. 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 extensive domain expert knowledge (e.g., knowledge of health care policies, drug regulations, medical law, health insurance policies). Other examples of the use of semantic rules are abundant, e.g., drug manufacturing services have to follow strict rules in obtaining and processing drug ingredients. Most life insurance application services require one or several verification services for the applicant's health status.

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, which may also be expressed by sets of rules. 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 and limitations.

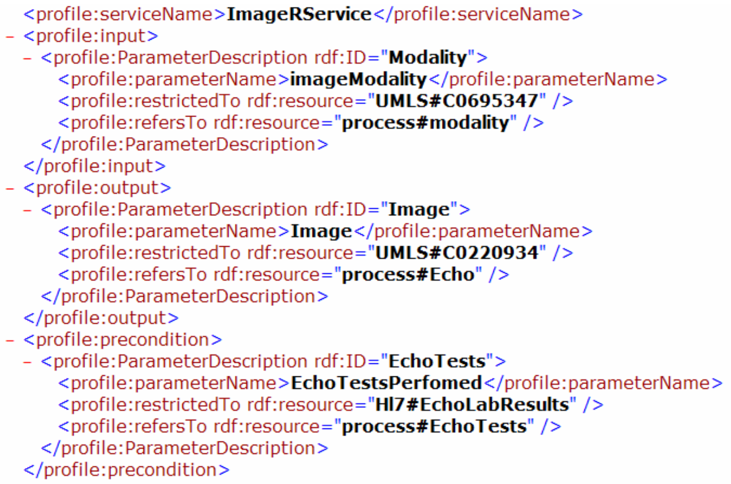
In real life it is often the case that several services possess the same profile and provide the same functionalities. The automated selection of one service among these functionally compatible services may require pragmatic knowledge. When selecting a lab for a blood test, choose one that is close to the patient's work place or home. This pragmatic rule applies when several labs that are in competition with each other provide the same type of services. When otherwise equivalent services differ in response times, a cheaper but slower service may be acceptable. If the information is needed before a specific date, then a provider that is more expensive may be acceptable. This kind of knowledge needs to be modelled in order for agent to automatically select one service over the other, and compose them in a sensible way. In addition, this pragmatic knowledge is highly personalized, as the perception of different brand names and loyalty to them is based on individual choices.

In summary, syntactic composition knowledge allows an agent to consider all compositions with *correct* I/O conditions. Semantic composition knowledge constrains the agent to make *sensible* compositions that conform with the relevant policies and organizational knowledge. Pragmatic composition knowledge allows the agent to make *reasonable* compositions with respect to the needs and preferences of the initiator of the agent. One may also look at the three kinds of compositional knowledge from an outcomes perspective. Without syntactic knowledge one cannot generate a service flow at all. Without semantic knowledge one cannot generate a service flow that conforms to all existing rules and policies and that produces an objectively reasonable result. Without pragmatic knowledge one cannot guarantee a service flow that produces a useful result for the given circumstances.

3.2. Semantic Medical Services Representation

Medical services range from simple patient record retrievals to complex services involving series of lab tests. In this section, we discuss the basic representation mechanisms involved in various types of compositional knowledge in medical settings. To model the syntactic and semantic compositional aspects, we use existing DAML-S based formalisms. The mappings of service parameters to existing medical ontologies and vocabularies must be considered here. The intent of the mappings is to achieve interoperability and smoother composition of medical services. Figure 2 shows a service profile for an Image Retrieval Service.

As depicted in Figure 2, the parameters are mapped to Ontologies like the UMLS (Unified Medical Language System) or HL7 (Health Level 7). When a Medical agent comes to this service to retrieve images for a patient, it needs to first check if EchoTests have been performed and EchoLabResults are available (the HL7 field for the EchoTests in not empty). The agent needs to contain input information related to the modality of image, whether it needs the M-Mode diagram (1D) or T-Mode diagram (2D) for the heart image (represented by MODALITY in the UMLS, by the given Concept Identifier). The output is an EchoCardiogram Image, mapped to the type of Ultrasound.



```
  <profile:serviceName>ImageRService</profile:serviceName>
- <profile:input>
  - <profile:ParameterDescription rdf:ID="Modality">
      <profile:parameterName>imageModality</profile:parameterName>
      <profile:restrictedTo rdf:resource="UMLS#C0695347" />
      <profile:refersTo rdf:resource="process#modality" />
    </profile:ParameterDescription>
  </profile:input>
- <profile:output>
  - <profile:ParameterDescription rdf:ID="Image">
      <profile:parameterName>Image</profile:parameterName>
      <profile:restrictedTo rdf:resource="UMLS#C0220934" />
      <profile:refersTo rdf:resource="process#Echo" />
    </profile:ParameterDescription>
  </profile:output>
- <profile:precondition>
  - <profile:ParameterDescription rdf:ID="EchoTests">
      <profile:parameterName>EchoTestsPerfomed</profile:parameterName>
      <profile:restrictedTo rdf:resource="Hl7#EchoLabResults" />
      <profile:refersTo rdf:resource="process#EchoTests" />
    </profile:ParameterDescription>
  </profile:precondition>
```

Figure 2. ImageRService's Profile (part)

To encode the pragmatic compositional knowledge in service descriptions, we need to model a layer of pragmatic rules on top of the existing ontology-based semantics layer. In the field of medicine some well documented standardized guidelines exist that govern medical procedures and hence the flow of information. Some of the currently used standards are GEM[2] (Guidelines Element Model) and CPGA[3] (Clinical Practice Guideline Architecture). The existence of such well defined and standardized guidelines allows us to formulate the basic pragmatic rules for medical services selection and composition. In fact, some underlying goals for developing the guidelines was to facilitate the workflow production and reduce inconsistencies or ambiguities, which are critical problems.

Currently, no specific rule representation formalism exists for the Semantic Web (though the development has started, in the form of work on OWL-Rules[4]). We are using the RuleML representation format to realize the pragmatics layer. The GML based guidelines were translated into RuleML using XSLT based transformations.

Consider, the Image Retrieval service (Figure 2). If the service user is a Cardiologist, he would generally work with 2D heart images. However, for a CardioVascular surgeon, such images are useless, since he is interested only in M-Mode images showing accurate measurements required to perform surgery. Hence, given a user request, the appropriate service is selected and composed in the service flow. Such pragmatic rules can also be used by the service user (or agent) to discriminate among many semantically matching services and select the best among them.

---

[2] http://ycmi.med.yale.edu/GEM/
[3] http://www.schin.ncl.ac.uk/cpga/
[4] http://www.daml.org/rules/proposal/

4. A Methodology for Semantic Interoperability

When an agent is looking for a service, it will carry with it a description of the kind of service that it is looking for, in terms of its underlying domain. It will encounter service descriptions using the same or different terms from the domain of the service provider. Unfortunately, even if the agent domain and the service provider domain are the same, that does not mean that the agent and the provider can smoothly interact, because there is no global shared ontology of domain terms. The situation is comparable to an Italian tourist in America that tries to order a meal from a Chinese waiter, and both know only subsets of English food language. The waiter and the tourist cannot start talking with each other directly. They need to establish a common language first, by discovering shared terms and finding mappings (hard!) between differing terms.

In ontology research this kind of process is described as a form of ontology integration. The heart of this process is to find mappings between differing terms for the same concept. This integration process has to be performed quickly, as one agent may be visiting many services with service ontologies in its attempt to construct a service flow. For any pair of sizable ontologies it is out of the question to perform a brute force attempt of matching every term in one ontology with every term in the other ontology. To overcome this problem we have developed an extensive method of semantic specification and semantic integration, using two-level ontologies, which is used as a precursor to a the actual integration algorithm [5].

4.1. Two Level Ontologies for Integration

Definition: Terminological Knowledge Base. We call any structure that consists of (1) a semantic network of semantic types; (2) a thesaurus of concepts; and (3) assignments of every concept to at least one semantic type a *Terminological Knowledge Base* (TKB).

$$\text{TKB} = < \hat{\mathcal{C}}, \hat{\mathcal{S}}, \mu >$$

in which $\hat{\mathcal{C}}$ is a set of concepts, $\hat{\mathcal{S}}$ is a set of semantic types (i.e., high-level categories), and μ is a set of assignments of concepts to semantic types. Every concept must be assigned to at least one semantic type. The opposite condition does not hold. We will use capital letters to represent semantic types and small letters to represent concepts.

$$\hat{\mathcal{S}} = \{W, X, Y, ...\}; \qquad \hat{\mathcal{C}} = \{a, b, c, d, e, ...\}$$

Finally, μ consists of pairs (c, S) such that the concept c is assigned to the semantic type S.

$$\mu \subset \{(c, S) |\, c \in \hat{\mathcal{C}} \,\&\, S \in \hat{\mathcal{S}}\}$$

We define that two concepts c, d are similar, $c \simeq d$, if they are assigned to exactly the same set of semantic types of a TKB.

$$c \simeq d :\ \forall S \in \hat{\mathcal{S}}\ [(c, S) \in \mu \ \Leftrightarrow\ (d, S) \in \mu]$$

If two concepts c and d are assigned to the same semantic type X, then these two concepts have similar semantics. On the other hand, if a concept a is assigned to X and a concept b is assigned to Y, then a and b will have semantics that are not similar in the formal sense defined above.

We now advance to integration. By our construction of the Terminological Knowledge Bases, two concepts, q from TKB$'$ and r from TKB$'_2$, can only match if they are both assigned to the same semantic type. There are three cases:

(1) Assume a semantic type S exists in TKB$'$ that has assigned concepts $x, y, z, ...$. Further assume that S does not exist in TKB$'_2$ or, there are no concepts assigned to S in TKB$'_2$. Then, by the similarity assumptions made above, no concepts corresponding to $x, y, z, ...$ exist anywhere in TKB$'_2$. Thus, these concepts don't need to be matched.

(2) The above observation applies in reverse also. If a semantic type S exists in TKB$'_2$ that does not exist in TKB$'$, then the concepts $x, y, z, ...$ assigned to S will not have corresponding concepts anywhere in TKB$'$. Thus, these concepts do not need to be matched at all.

(3) Concepts assigned to the semantic type S in both TKB$'$ and TKB$'_2$ are potentially similar ($\simeq$) and need to be matched. As mentioned above, we allow partial matches between concepts that have been determined to be similar. The exact cut-off is decided by a threshold value.

4.2. Scoring Concept Similarities

4.2.1. Ranking Concepts by their Terms If two concepts have similar names (defined below, based on bigrams) then they are possibly matches. The existence of synonyms and homonyms causes problems for concept matching. The use of synonyms is absolutely necessary, because medical terminologies are full of variant terminologies (e.g., Heart and Coeur, Heart Block and Lev's disease). We include the use of synonyms during the concept matching step itself. If no match is found for a concept, then it is attempted to use its synonyms for matching.

4.2.2. Ranking Candidates by Attributes We assign to every pair of concepts a score as follows. Two concepts that have the same number of attributes, and for every attribute in one concept there is an attribute in the other concept of the same name and same data type, are considered perfectly matched, with a score of 1. If two attributes (of two concepts from two ontologies) have the same name but are of different data types, we assign them a score of k ($k < 1$, $k \gg 0$). Then we compute the ratio of matched attribute scores divided by the number of attributes of the concept that has more attributes. The final decision about similarity is made, based on a minimum threshold for the computed combined score.

4.2.3. Ranking Candidates by Relationships using Propagation Two concepts that point to exactly the same concepts with the same relationships are presumably very similar to each other. We view the relationship targets as data types, and two concepts that point to all the same data types are likely to be quite similar. By this step, we create an initialization for matching up additional concepts by using relationships. Thus, two concepts with different names that point to several target concepts that all have been matched up between two ontologies are presumably themselves a match. We can use a similar ratio criterion as for attributes, however, now the targets carry more semantics than the undifferentiated data types of attributes. Thus, we are willing to assign a pair of relationships a high score if the targets are the same OR if the relationship names are the same.

Thus, the process of finding matches needs to be recomputed until a score change of one concept pair does not result in a score change of any concepts pointing to that pair anymore. Note that this state of equilibrium can be easily achieved, as we are using a threshold. If there are only changes that do not cross the threshold, the update process would terminate.

4.2.4. Combining Matching Scores Two concepts are considered matched if their terms, their attributes and their relationships are (on average) similar. A weight is assigned to each similarity aspect of a concept (term similarity, average attribute similarity, average relationship similarity). Considering these three criteria, we now compute the degree of the similarity of concepts from two distinct ontologies. For this purpose, we use a Multiple Attribute Decision Making (MADM) approach, a simple additive weight-based solution [10]. After a combined score has been computed, we compare the weighted sum with a given threshold α. Some matches may be lacking attributes or relationships. In this case, a weight of zero will be assigned to these aspects of a concept. Details of this algorithm are given in [15].

5. The SMS System

We are in process of developing set of applications, tailored to meet the needs for developing Semantic Medical Services. Currently, we are working with a number of local medical organizations, where we are in the process of integrating various informatics solutions to facilitate collaborative medical research. We are creating wrappers for the legacy solutions using semantics based Web services (Table 1). The central concern is to map the service parameters to existing medical ontologies and vocabularies (Table 2) to facilitate matching and subsequent composition. Table 1 lists some of the services being modeled and the corresponding parameters.

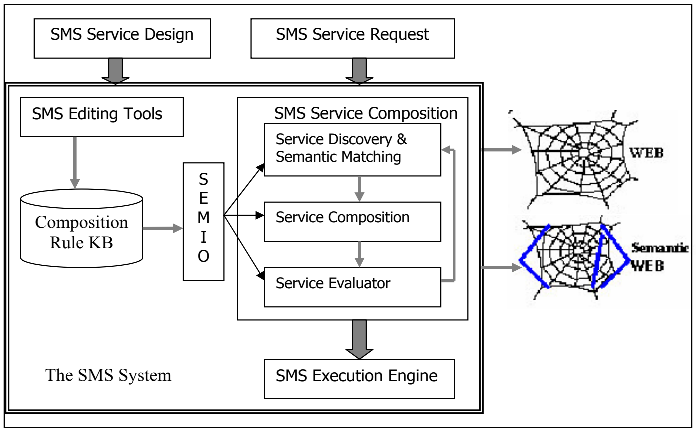


Figure 3. The SMS System Architecture

SMS Editor: A platform to model and create Semantic Medical Web services over existing legacy applications (Figure 4). The editor allows mapping of service parameters (input, output, preconditions, effects) to concepts in predefined medical ontologies. New ontologies could be loaded in the editor. The editor has an ontology search component which performs keyword based search in the ontologies. The user can map resultant concepts to service parameters. The editor parses the WSDL (Web service Definition Language) documents and creates the service grounding descriptions. One interesting feature is plugging of new medical ontologies (required for mapping service parameters). Once stored in the composition rule KB these ontologies are used by the Semantic Matching component.

SMS Evaluator: This component performs evaluation of a service based on the QoS parameters of a particular service. We perform evaluation based on some simple evaluation matrices, which is similar to match algorithm described in [14]. While performing some pragmatic rules evaluation, for example, HL7 ADT (Admit-Discharge-Transfer) messages should be transported to all allied information systems in nearly real time (< 1 sec delay) and to incorporate such pragmatic rules, we need the evaluator to perform a *test* run to satisfy the pragmatics requirements.

SMS Matcher and Composer: Using the semantic integration algorithm (SEMIO) as described in Section 4, this module performs syntactic and semantic matching of concepts. For given two concepts which are basically the service parameters, the component tries to establish a match between them. The service composition aspects was bit

Services	Concepts (used as Parameters)	Description
HL7Services		Retrieve Patient related information using Master Patient Index from Hospital Information System.
1. PatDemoGraphics	MeSH#DemoGraphics UMLS#Patient address (CUI:CO421449)	
2. PatHistory	UMLS#Medical History (CUI: C0262926)	
LabServices		Services related to retrieving various laboratory tests for a patient (could be distributed over number of organizations)
1. CathServices	UMLS#Finding (CUI: C0578829) UMLS#Angioplasty (CUI: C0162577)	
ImagingServices		Services providing Imaging Information for a patient coalescing various images with different modalities
1. CardioImageServ	SNOMED#VentricleEKG (T-32400)	
2. ModalityServ	UMLS#Ultrasonic UMLS#NMR UMLS#PET	
CodeLookupServices		Large amounts of information in medical databases is stored using standard Medical Codes. Lookup services facilitate conversion process
1. ICDServ	UMLS/ICD	
2. CPTServ	UMLS/CPT	
3. ACCServ	ACC	

Table 1. The SMS Services

Ontology Namespace	Ontology Name and Location	Description
UMLS	Unified Medical Language Source http://www.nlm.nih.gov/research/umls/	The UMLSKS is a set of Web based interaction tools and a programmer interface to allow users and developers access to the biomedical terminologies found within the UMLS.
MeSH	Medical Subject Headings http://www.nlm.nih.gov/mesh/MBrowser.html	Vocabulary look-up aid available for locating descriptors of medical publications.
SNOMED	Systematized Nomenclature of Medicine http://www.snomed.org/	Validated terminology that enables clinicians, researchers and patients to share health care knowledge worldwide, across clinical specialties and sites of care.
NCI	Cancer Ontology http://www.mindswap.org/2003/CancerOntology/	Logic-based terminology produced by the National Cancer Institute broad clinical vocabularies, implementing rich semantic interrelationships between the nodes of its taxonomies.
GO	Gene Ontology http://www.geneontology.org/	A controlled gene vocabulary that can be applied to all organisms
ACC	American College of Cardiology http://www.acc.org/	Cardiovascular terminology for nurses, doctors and physician assistants.

Table 2. The Ontologies/vocabularies used in the SMS system

tricky where we followed a cyclic approach: matching of concepts followed by pragmatic evaluation.

SMS Execution Engine: Once the services have been discovered and composed to satisfy the goal, this module actually executes the services. We used some of the existing tools available for this purpose[5], this tool mandates the process specification in specific format. In our case, the process specification are generated as the result of refinement of the composition process.

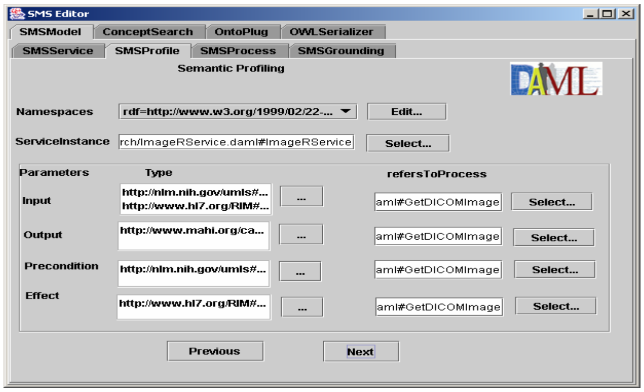
6. Related Work

Current Web services support a certain level of interoperability in using and accessing them. The next level of interoperability cannot be achieved by just making services available, but requires providing automatic mechanisms so that the services can be linked in appropriate and meaningful ways [6]. Semantic interoperability is essential for automated discovery, matching and composition of services. This enhancement depends on the existence of ontologies for the terms used by Web services. The Semantic Web research work, following the DARPA Agent Markup Language (DAML), includes DAML+OIL [9] for the creation of arbitrary domain ontologies [6] for the semantic mediation between services and workflow logic. Some research focuses on compositions of services using workflow management. Automatic composition of Web services [11] has been achieved through automated mapping, composition and interoperation of services, service verification, and execution monitoring. Process modeling languages such as PIF, PSL were designed to support process management.

Several applications require that multiple ontologies are combined into a single coherent ontology [13]. Many lines of research have addressed ontology matching in the context of ontology construction and integration [3] and effective methodologies for automated mappings [12]. Ontolo-

---

[5] http://taverna.sourceforge.net/



Namespaces rdf=http://www.w3.org/1999/02/22-...

ServiceInstance rch/ImageRService.daml#ImageRService

Parameters	Type	refersToProcess
Input	http://nlm.nih.gov/umls#... http://www.hl7.org/RIM#...	aml#GetDICOMImage
Output	http://www.mahi.org/ca...	aml#GetDICOMImage
Precondition	http://nlm.nih.gov/umls#...	aml#GetDICOMImage
Effect	http://www.hl7.org/RIM#...	aml#GetDICOMImage

Figure 4. The SMS Editor

gies are used for constraining the parameters of dynamic service configurations. Reasoning to ensure the semantic validity of compositions is used for automated workflows. Scientific workflow [2] is supposed to support interoperation through semantics. It may have the potential to support Web service descriptions for service discovery, invocation, activation and execution of an identified service by an agent or other service [11]. Unlike these efforts, our approach emphasizes the importance of different kinds of knowledge, especially pragmatic knowledge, and the ontological methodology for heterogeneous semantics for the automatic composition of service flows.

There have been efforts in representing business contracts for service evaluation and negotiations [7] but how to use such pragmatic knowledge for service matching remains still unresolved. In other work [8], workflows in medical settings have been studied, but the efforts are more geared towards QoS (quality of service) and workflow execution aspects. We address the need to consider a broad set of pragmatic rules (including QoS) to compose a service flow of medical services for the Semantic Web.

7. Conclusions and Future Work

In this paper, we have laid out an architecture of the knowledge processing that is necessary for composing individual services into service flows. We have classified service description knowledge into syntactic, semantic and pragmatic composition rules that play a major role in discovery, selection and composition of Web services. These rules incorporate the knowledge that is necessary to select and to compose Web services into a coherent service flow. The description of rules with service concepts allows the system to identify the relevant rules in a certain domain and to identify and select appropriate Web services for composition. Secondly, we have introduced an ontological integration methodology dealing with heterogeneous semantics existing in Web service composition. This method relies on the use of a two-level ontology. Future work includes the extension of compositional knowledge to include negotiation rules. When certain services in the process of service selection do not exactly meet the conditions of a rule, then there should be a possibility to relax the conditions to continue with the selection and integration process.

References

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

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

[3] H. Chalupsky. Ontomorph: A translation system for symbolic knowledge. Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning. 2000.

[4] S. A. Chun, V. Atluri and N. R. Adam, Domain Knowledge-based Automatic Workflow Generation, in proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002), France.

[5] J. Geller, H. Gu, Y. Perl and M. Halper, Semantic refinement and error correction in large terminological knowledge bases, Data & Knowledge Engineering, 45(1), 2003, pp. 1-32.

[6] C. A. Goble. Supporting Web-based biology with ontologies. In The Third IEEE ITAB00, pp. 384-390, 2001.

[7] B. N. Grosof, T. C. Poon, SweetDeal: Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions, In Proc of WWW 2003, May 20-24, 2003, Budapest, Hungary.

[8] J. Cardoso and A. Sheth, Semantic e-Workflow Composition, Journal of Intelligent Information Systems, 2003. (In press)

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

[10] C.L. Hwang and K. Yoon, Multiple Attribute Decision Making, Springer-Verlag, Berlin, 1981.

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

[12] Maedche. A machine learning perspective for the Semantic Web. In Proceedings of Semantic Web Working Symposium (SWWS), 2001.

[13] N. Noy and M. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of the National Conference on Artificial Intelligence, 2000.

[14] Massimo Paolucci, Katia Sycara, and Takahiro Kawamura, ”Delivering Semantic Web Services.” In Proceedings of the Twelves World Wide Web Conference (WWW2003), Budapest, Hungary, May 2003, pp 111- 118 .

[15] K. Supekar, The SEMIO Implementation Report, UMKC, Technical Report, 2002.

[16] W. W. Stead, R. A. Miller, M. A. Musen, and W. R. Hersh, Integration and Beyond: Linking Information from Disparate Sources and into Workflow, Am Med Inform Assoc 2000;7(2):135-145