



Semantic annotation of image processing tools

Bacem Wali, Bernard Gibaud

► To cite this version:

Bacem Wali, Bernard Gibaud. Semantic annotation of image processing tools. International Conference on Web Intelligence, Mining and Semantics, Jun 2012, Craiova, Romania. pp.article n°29. hal-00690926

HAL Id: hal-00690926

<https://hal.archives-ouvertes.fr/hal-00690926>

Submitted on 24 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semantic annotation of image processing tools

Bacem Wali

INSERM/INRIA/Univ. Rennes 1, VISAGES U746

Faculté de médecine Rennes, France

bacem.wali@irisa.fr

Bernard Gibaud

INSERM/INRIA/Univ. Rennes 1, VISAGES U746

Faculté de médecine Rennes, France

bernard.gibaud@irisa.fr

ABSTRACT

Collaborative biomedical imaging research raises the issue of coherently sharing data and processing tools involved in multi-centric studies. Federative approaches are gaining increasing credibility and success to build distributed collaborative platforms. In the context of the NeuroLOG project, we designed the OntoNeuroLOG ontology as a cornerstone of our mediation layer. This contribution focuses on processing tools and is two-fold. We propose an extension of the OntoNeuroLOG ontology to conceptualize shared processing tools and enable their semantic annotation. Leveraging this modeling, we propose a set of semantic treatments aimed at easing their sharing, their reuse and their invocation in the context of neuro-data processing workflows.

Categories and Subject Descriptors

D.3.1 [Software]: Formal Definitions and Theory—*Semantics, Syntax*; I.2.4 [Computing Methodologies]: Knowledge Representation Formalisms and Methods—*Representation Language*

General Terms

Algorithms, Design, Languages, Verification.

Keywords

Web Services, Semantic annotation, Web Services composition.

1. INTRODUCTION

Biomedical imaging research increasingly involves multi-centre studies, raising the issue of sharing (i) the data produced by the cooperating centers, and (ii) the processing tools to be used in those studies. Although more complex than centralized ones, federative approaches are gaining increased credibility and success, as can be seen with initiatives like the Biomedical Informatics Research Network (BIRN) and the Cancer Biomedical Informatics Grid (caBIG) in the US, and e.g., projects like @NeurIST in Europe. Especially, data integration in federated systems becomes feasible, thanks to ontology-based mediation [1]. The issue of sharing and reusing processing tools received less attention, yet. We present here our work focusing on the sharing of processing tools for neuroimaging research, carried out in the context of the NeuroLOG project. This project aims at developing a middleware to federate data repositories and to facilitate the sharing and reuse of processing tools to analyze the

shared images. A key aspect of the NeuroLOG approach is the definition of an application ontology (called OntoNeuroLOG) to annotate the heterogeneous resources to be shared [2].

The NeuroLOG project¹ is a framework developed to share resources in the neuroimaging domain. By resources we mean brain images obtained using various imaging modalities (MRI, CT, etc. ...) and image processing tools deployed as Web Services or composed as workflows. OntoNeuroLOG [3] is an application ontology designed specifically to provide common semantics of shared information throughout the system. The deployment of image processing tools as web services and their execution are ensured by the jGASW [4] software, which purpose is to create XML descriptions of their inputs, outputs and parameters. Besides, service composition is maintained by the MOTEUR [5] software which is an engine for defining strategies and policies through XML descriptions to match and invoke web services as well as orchestrate workflows. Both address specific requirements of image processing tools, such as coping with image formats and headers or more complex parameters.

Web Services are self-contained, modular applications, accessible via the Web [6], that provide a set of functionalities to businesses or individuals. As loosely coupled, reusable software they are designed to be manipulated remotely over a network and they have capabilities to invoke each other [7]. Web service composition [8] is a new trend and has been given a lot of attention in research, thus leading to manifold technological advances in the integration of web services, especially with Service Oriented Architecture SOA [9]. Increasingly, SOA is becoming among the most used technologies for integrating software resources by assembling autonomous web services, unfortunately there is a lack on common understanding due to the poor semantic expressivity of used documents for web services descriptions [10]. Indeed, Web services rely on XML descriptions like WSDL², UDDI³ or SOAP⁴ that represent an abstract interface of services and contain the basic properties in term of inputs outputs and capabilities.

SOA environment has emerged in the workflow domain and mainly addresses the issue of orchestration of web services according to the BPM (Business Process Management) [11]. From a process point of view, a workflow is composed of procedural steps, each of which is fulfilled by an activity performed by one or more of the services. From an SOA point of view, a workflow is a set of services, each of which interchanging data and information in a logical controlled way. Semantic Web and SOA provide solutions to handle web services reusability, composition, flexibility and interoperability [12] [13] [14]. The Semantic Web is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. WIMS'12, June 13-15, 2012 Craiova, Romania Copyright © 2012 ACM 978-1-4503-0915-8/12/06... \$10.00.

¹ <http://neurolog.i3s.unice.fr/neurolog>

² <http://www.w3.org/TR/wsd1>

³ <http://uddi.xml.org/uddi-org>

⁴ www.w3.org/TR/soap/

a set of technologies designed to make the resources accessible and understandable by programs and software agents. Semantic annotation based on ontologies is the most popular way of sharing information semantics. It is increasingly used and has become a most praised way to enhance resource sharing and workflow composition within SOA architecture in the biomedical area [7] [22] [25]. Indeed, by providing a common specification of shared entities, web services can be more easily linked to business processes. Beyond, ontologies help describing the information content of inputs, outputs as well as the capabilities of services, so that automated processing can be applied to them.

Eventually, we have at our disposal a well-defined ontology representing many entities relevant to the neuroimaging domain. We have also at our disposal two powerful tools for the advertisement and execution of images processing tools within the NeuroLOG framework. However, this is not sufficient to enable an effective reuse, especially due to insufficient information about the expected context of use of those tools. Indeed, the OntoNeuroLOG ontology encompasses the share of scientific studies and it is helpful for users to track the improving of those scientific studies over the time, but sharing concepts and conceptualizations is not enough to support the sharing and reuse of image processing tools. They require some specific semantic verification according to the neuroimaging domain expectations, thus, we should add semantics to better control that the execution and orchestration of the services meet essential constraints related to underlying business processes. Besides, actual reuse of the datasets produced along the execution of the web services requires that they be properly annotated with semantic information pertaining to the business domain. Indeed, this is not provided by the processing tools themselves, whose function is limited to creating the processed image files. Some complementary semantic processing must be achieved to provide such annotation based on available knowledge about the processing tools being used and existing context of the processing.

In other words, image processing tools are shared technically but, though they are based on quite poor XML descriptions, and so they remain difficult to manage. Also, processed information is frequently ignored because it is generally not well understood. More broadly speaking, we need to add metadata associated to services as well as to their execution context in order to ensure the appropriate design and use (i.e. invocation, composition) of the services and the appropriate use of the data resulting from their execution.

This paper highlights semantic requirements of image processing tools within the NeuroLOG framework and the proposed approach tackles 3 aspects of semantic annotation of image processing tools:

1°) Ensures semantic annotation of image processing tools (simple and composite) according to the OntoNeuroLOG ontology and allows the user checking whether such annotation makes sense and meets neuro-imaging expectations by implementing some specific verification algorithms.

2°) Verifies whether the composability is possible.

3°) Makes possible to infer new knowledge along the platform exploitation. This last point is achieved by adding rich semantic rules according to the nature of the image processing tool and the execution expectations. The new metadata are generated automatically from rules, and enriches the experimental platform and provides new valuable expert information.

OntoneuroLOG relies on DOLCE (Descriptive Ontology for Language and Cognitive Engineering), a foundational ontology that provides both the basic entities (at the top of the entities' taxonomy) and a common philosophical framework underlying the whole conceptualization [3].

In this paper we present our ontology of the shared services and its usage. The ontology was indeed built in a bottom-up approach to facilitate the sharing, invocation and reuse of services in new image processing pipelines [15]. The following of the paper is organized as follows. Section 2 discusses the background of the work, the section 3 describes the proposed ontology, as well as the semantic processing operations that produce and exploit semantic annotations based on this ontology. Section 4 provides further details on the implementation of semantic services and their integration in the NeuroLOG middleware. Section 5 highlights the added value of this approach, briefly situating this work with respect to similar modeling work, e.g. in the context of W3C, and draws some perspectives.

2. RELATED WORK

2.1 Semantic description and composition of web services

Over the last decade, the shift in emphasis from abstract specification based on XML language toward semantic description of various data processing in the field of life sciences has increasingly invaded the thoughts of all researchers and scientists. Today, we are all well aware of the need to add more semantics in the mechanisms of invocation, composition and orchestration of web services.

Among the first tools that have emerged, WS-BPEL [16] is the most popular and the most widely used language in the industry for defining the execution logic of the business process. It extends WS Description (WSDL) with an XML description and models behavioral aspects of services like business protocols. Some works interested in BPEL underline the different aspects of structural matching within it due to the lack of formal semantics and the heterogeneity of syntactic construct uses [17] [35]. Through the semantic web community, many standards have been proposed DAML+OIL, DAML-S [18], OWL⁵ and OWL-S⁶ which is an initiative which aims to facilitate automatic discovery, invocation, composition interoperation and monitoring of web services through their semantic descriptions. OWL-S supports a rich semantic description, namely (1) a profile that describes what the service actually does, (2) a process model that specifies how the service works in terms of inputs outputs preconditions and effects, and (3) a grounding that defines how the service is accessed. OWL-S adds precondition and effects to specify workflow coordination and service discovery. WS Description is based on the OWL language which is an extension of the RDF language by adding more semantic vocabulary (based on Description Logics) to describe concept properties and to allow more reasoning upon web resource content and capabilities. Such capabilities allow automatic selection, composition and orchestration of web services. However, reusability in the context of DOLCE is more complex challenge. WSDL-S⁷ defines a mechanism to add semantics to WSDL, it was initially developed by the LSDIS lab of the University of Georgia, then refined by the W3C [19]. It consists in

⁵ <http://www.w3.org/TR/owl-guide/>

⁶ <http://www.w3.org/Submission/OWL-S/>

⁷ <http://www.w3.org/Submission/WSDL-S/>

enriching the WSDL with *ModelReference* property by adding pointers to the application domain. The approach keeps annotation mechanisms separate from the semantic description to offer flexibility to select multiple semantic descriptions and frameworks. However, the approach presents some limits regarding capability of orchestration and composition of services mainly because there is no control construct mechanism implemented, and there is no implementation within other frameworks that use workflows to address this issue, therefore, the reasoning side is really neglected. Similarly, SA-WSDL⁸ allows selecting the most appropriate domain ontology and then mapping WSDL elements to respective ontological concepts [20]. WSMO [21] is a W3C submission that aims at offering a mechanism for automatic service mediation, discovery, invocation, composition and interoperation of services. It supports four basic concepts called top-level elements, which are (1) Ontologies (2) Goals (3) Web Services and (4) Mediators. Through the mediation layer WSMO addresses interoperability issue between loosely coupled web services, goals and protocols via the ontology layer. In [21], they use the WSMX which is an execution engine based on the WSMO specification to build a hybrid matchmaker by combining both syntactical and logical constraint matching. However, in the METEOR-S project, efforts are concentrated on the matchmaking between services based on multiple abstract specifications (WSDL, WSDL-S, SA-WSDL ...). Thus, WSMO enhances the capability of systems to discover, select and rank web services described using various frameworks.

2.2 Some existing solutions

Data integration in the biomedical domain is becoming more ubiquitous. Goble et al. emphasize the need for common, shared identities and names, the need of shared semantics and stable access mechanisms, and the role played by standards to alleviate heterogeneity and data sharing problems [10]. The problem is spreading up the interoperability, composition and discovery of resources (i.e. data or services) and the need of shared semantics to enable reasoning [8].

In [22], the authors alleviate the problem of heterogeneity by fostering an additional level of description of Web services called *pragmatic* or *contextual* knowledge, which is an intermediary layer to share and underline conceptualizations coming from various domain ontologies. The contribution is focused on enabling automatic composition of service in medical domain. The approach emphasizes the need of this kind of layer to bridge the gap of heterogeneity of ontological frameworks. Our case is approximately the same, indeed, by using shared domain specific terms from the OntoNeuroLOG.

By the way, ontologies are gaining increasingly a confidence of use in order to enhance data mediation. For example, the BIRN project [23] funded by the National Center for Research Resources (NCRR), provides a data sharing structure for sharing data and services in the biomedical domain. The data mediation within the BIRN project is based on ontologies to structure the domain vocabulary and on rules to enrich the knowledge base. However, the BIRN architecture enables participant groups to deploy their own workflow engines (such as the LONI pipeline). Some tools deployed at the partner sites of BIRN can be invoked with local or distributed data [24].

Due to its efficiency, interoperability, scalability and simplicity of integration, SOA gains more and more attention in the biomedical domain and various disciplines. The Living Human Digital Library [36] built a SOA-based biomedical digital library infrastructure to share multiscale data resources and data processing. SOA solves various problems like heterogeneity by decoupling real execution and interoperability among services, and enables reuse of processing algorithms. In SOA environments the workflow is represented as a set of web services. The SOA approach enables federating data access and analysis across different institutional and disciplinary sources and leverages a standard interface with which users can use, compose, and invoke services. Web services are software components that can be accessed by other software. Many efforts are focused on the automation of service composition; mostly they speak about semantic models. The BioMOBY is an open source framework aiming at exploring web services deployed on shared registries. There is a lot of works in the semantic domain that rely on BioMOBY framework, for example in [25] the authors use a semi-automatic approach to facilitate service selection, composition and execution within the BioMOBY framework. Their approach is based on an interaction model that defines the types and roles of inputs and outputs and how services interoperate. It (1) provides composition and re-composition of workflows that already exist, (2) enhances the selection algorithm by hiding not interesting bindings (3) visualizes results in a timely way. To fulfill users' requirements a lot of algorithms are discussed like type compatibility, graph construction and ranking.

In [26] the authors explain that the TAVERNA [27] workflow enactment of BioMOBY together with the MyGrid, allow semantic discovery and composition within grid technologies. Taverna is a software created in the context of the MyGRID project aiming at providing an infrastructure to design and execute scientific workflows to help *in silico* experimentation. MyGrid is a project aimed at sharing grid skills through resource sharing technologies, using therefore higher throughput data coming from the bioinformatics domains [28]. All those skills are gathered in the MyExperiment environment [29] to share new workflows and experiments. Similarly, @NeurIST project supports integration of heterogeneous resources for the treatment of cerebral aneurysms. Like most similar frameworks, system mediation is based on ontologies [30] and mapping schemas. The middleware layer represents the cornerstone of the application. It is composed of infrastructures dealing with security layer [31] via SOA, WSDL, SOAP and web service architecture [32].

The next section presents our contribution to overcome heterogeneity and lack of semantic interoperability issues in the context of the NeuroLOG framework.

3. METHOD

Sharing processing tools in a federated system requires overcoming the heterogeneity of their implementation. This issue has two facets. The first is of a syntactic nature, and deals with the practical way of selecting the corresponding software (e.g. from a repository) and of invoking it. The second is of a semantic nature and concerns the homogeneous and consistent definition of the classes of processing in an application domain and of the related data being processed. In NeuroLOG, the syntactic problem was addressed through encapsulation in Web Services (WS) that can then be invoked in a homogeneous way throughout the federated system. This was achieved thanks to a software component called jGASW that wraps legacy application codes with a standard WS

⁸ <http://www.w3.org/TR/sawSDL/>

invocation interface and instruments them for integration. The semantic issue is discussed below in this section.

3.1 Ontology of Web Services processing tools

(In the following, the classes and properties of the ontology are represented in courier font and the properties denoted in italics).

The definition of this ontology was made using the common modeling framework used throughout the NeuroLOG project [3]. It relies on the foundational ontology DOLCE and on a set of core ontologies modeling key entities that are involved in different domains. Our model of WS highlights a few notions that are classically involved in WS specification such as the notions of interface (*ws-interface*), operation (*ws-operation*), input and output variables (*input-variable*, *output-variable*). Besides, the model introduces relationships with the classes of data processing that a particular *ws-operation* implements (such as *rigid-registration* or *segmentation*), as well as with the classes of entity that the input and output variable actually represent.

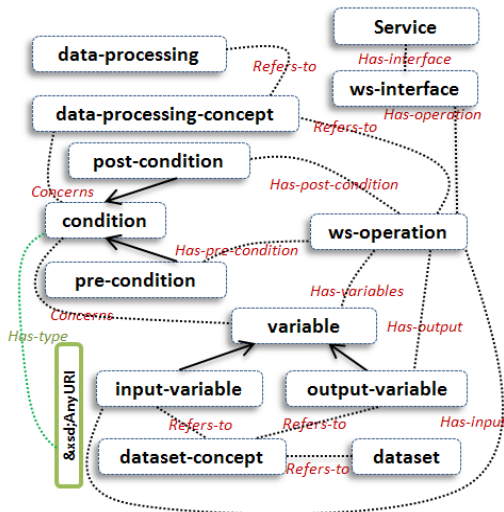


Figure 1. Semantic model for describing image processing tools

Indeed, OntoNeuroLOG defines within our application domain taxonomies for (i) the classes of data processing (denoting the actions performed by image processing tools), and (ii) datasets, a common abstraction for all kinds of images and roles of data processing tools [3]. For image processing tools those taxonomies represent inputs and outputs data types and tools functionalities. The major problem here, is the semantic that those taxonomies express according to the DOLCE, in fact, they cannot be instantiated (this will be discussed in discussion section). Therefore, according to this expectation, we define *dataset* concept and *data processing concept* taxonomies; concepts replace concepts of aforementioned taxonomies and can be instantiated according to the foundational ontology DOLCE.

Indeed, based on DOLCE and on the core ontologies used in OntoNeuroLOG, we choose to specifically use the *refers-to* relation that basically links representational entities (e.g. Propositions) to the actual entities that they represent.

Through *pre/post-conditions* our aim is to represent explicitly the conditions for a relevant use and understanding of the service as well as the data that the service allows to produce.

For example we add the precondition ‘input1 and input2 must refer-to the same dataset class’ or the postcondition ‘output1 must refer-to the same subject or group of subjects to which the input1 refers-to’. We have conceived an annotation method used at the tool design time and set up a mechanism to automatically generating such rule. This annotation method consists in adding links between the condition and the classes of concepts with which it interacts: for example via the *concerns* data property we link inputs and outputs of the concerned service and the data processing that we intend to carry out.

We hypothesized that human experts use their own domain knowledge to define some kinds of rules. Those kinds have a common sense over processing and tools. The *Has-type* data property defines the kind of the pre/post-condition (*SameClassOf*, *refersTo* ...). Finally, the model includes a notion of orchestration (*orchestration*), which models how a *ws-operation* can rely on other *ws-operation* to execute, thus composing a workflow (Fig. 2.). This involves a mapping entity (*mapping*), and three specializations (called *i2i-mapping*, *o2i-mapping* and *o2o-mapping*) that express how the « global » input and output variables (i.e. referring to the orchestrated operation) are connected to the « internal » input and output variables.

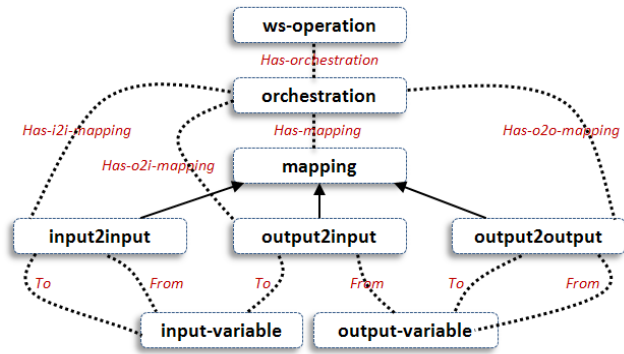


Figure 2. Semantic model for describing the composition of image processing tools as workflow

A second aspect of this ontology deals with the execution of operations (*ws-operation-execution*). This entity allows « provenance » information to be represented (Fig. 3.), e.g. to relate a particular *ws-operation-execution* to the specific values assigned to the input and output-variables, as well as the relationships to the entities that they represent.

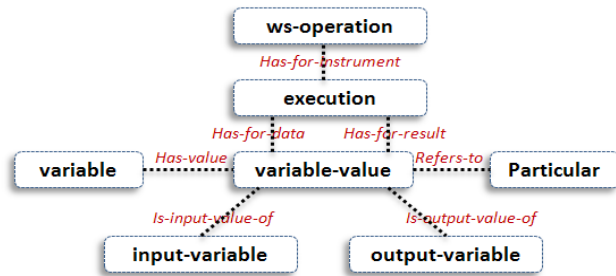


Figure 3. Model of semantic execution of image processing tool

For, example variable-value *refers-to* some particular allows to express that a particular variable-value instance represents in fact a particular image, e.g. a T1-weighted-MR-dataset instance.

A third aspect is the grounding between a concrete realization of services and flows and the semantic descriptions;

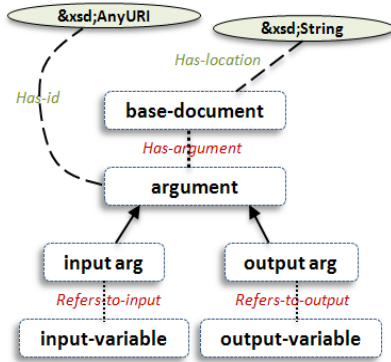


Figure 4. Model for the generic grounding of simple or composite image processing tool

Neither jGASW (for jGASW tool) nor GWENDIA (for MOTEUR tool) descriptors actually contain the notion of operation found in WSDLs; they have only inputs, outputs and mappings. For this reason we add *base-document* representing the descriptor and *argument* which represents the inputs or outputs of the service. At execution time they are known using the *Has-id* data property and they are linked to the inputs and outputs variables defined within the *ws-operation* of the service via the *Refers-to-input* and *Refers-to-output* object properties.

The ontology is represented in OWL, and uses the OWL-Lite subset in order to cope with the limitations of the reasoners used in our implementation.

3.2 Semantic services

In this section we describe the set of semantic services implemented within the NeuroLOG framework.

- Semantic annotation of simple and composite services, according to the model described above. This module is executed when a user annotates a service. It consists on the one hand in specifying the class of processing realized by this service and on the other hand the classes of entities that are involved in the processing, as inputs, outputs or parameters. The operation itself consists in checking the consistency of these specifications. Service grounding is also involved. Basically, annotations rely on the raw XML files (i.e. jGASW descriptor or GWENDIA descriptor)

Validation of an orchestration is executed when a user builds a new workflow involving existing WS operations. The processing consists in ensuring for each of the mappings of this workflow that the classes of entities referred-to-by the input and output variables are consistent (i.e. source subsumed by target)

N.B. Consistency is valid if orchestration is valid and type compatibilities between all mappings are valid.

Annotation is valid if compatibility between operation and data processing is valid.

- Semantic invocation of simple and composite services: This operation is executed when a service is invoked. It ensures that the real instances selected by the user (e.g. a Dataset) and assigned as variable values actually meet the constraints specified in the semantic annotations of the service. In practice, the semantic service checks whether the class of this instance is subsumed by the class of value (Dataset-concept) that is-referred-to-by the corresponding Input-variable. As a result real produced images get the data type from the Output annotations.

- Generating and applying rules to produce metadata: we can add rules to services according to the model described above. At the execution time, rules will be generated automatically from their descriptions. Thus, concerned variables will be substituted by real values used or produced. When the rule is applied metadata are added to the knowledge base.

- Check Compatibility between data processing class and operation (*Roles and Types*): This service allows users to ensure that the operation is compatible with data processing selected by the user at annotation time. The algorithm is the following: first we create a temporary class *new_data_processing* class relatively to the current operation, then we will try to convert relations between operation, inputs and outputs to axioms and add them to the *new_data_processing class*. Thus, for every relation *has-input/has-output* we count the number of inputs grouped by dataset concept nature to extract the *cardinality* of the axiom that we try to add.

- The definition of data processing differs from the definition of operation and uses different object properties thus requiring the third step, explained hereafter. It consists in *selecting the appropriate object property* for the construction of the axiom according to the nature of the dataset referred by dataset-concept. For example, the (1) (*has-input i1 refers-to Mr-dataset-concept refers-to Mr-dataset*) is substituted by *has-for-data-at Mr-dataset*, and (2) (*has-input i1 refers-to floating-dataset-concept refers-to floating-dataset*) and (*has-input i1 refers-to Mr-dataset-concept refers-to Mr-dataset*) both are substituted by *has-for-floating Mr-dataset*. Otherwise, if the class referred by the concept corresponds to a role in OntoNeuroLOG, then we replace the *has-for-data-at* object property by the adequate relation (i.e. bearing the semantics of the corresponding role) and add them as axioms to the temporary class already created. For example, *floating-dataset* is a role, then should be replaced by the *has-for-floating-dataset* object property. Thus, we have a new data processing class having the adequate relations and datasets according to the operation. The last step is to add the new *new_data_processing* class with axioms added above as subclass of the class referred by the data processing selected by the user, and then, classify and check consistency. If the ontology is consistent then the annotation is valid. Semantically, the functionality of the tool is agreed (i.e. *has-for-data-at/has-for-result-at* object property describe outputs in the data processing) Fig. 5. Shows an illustrative example.

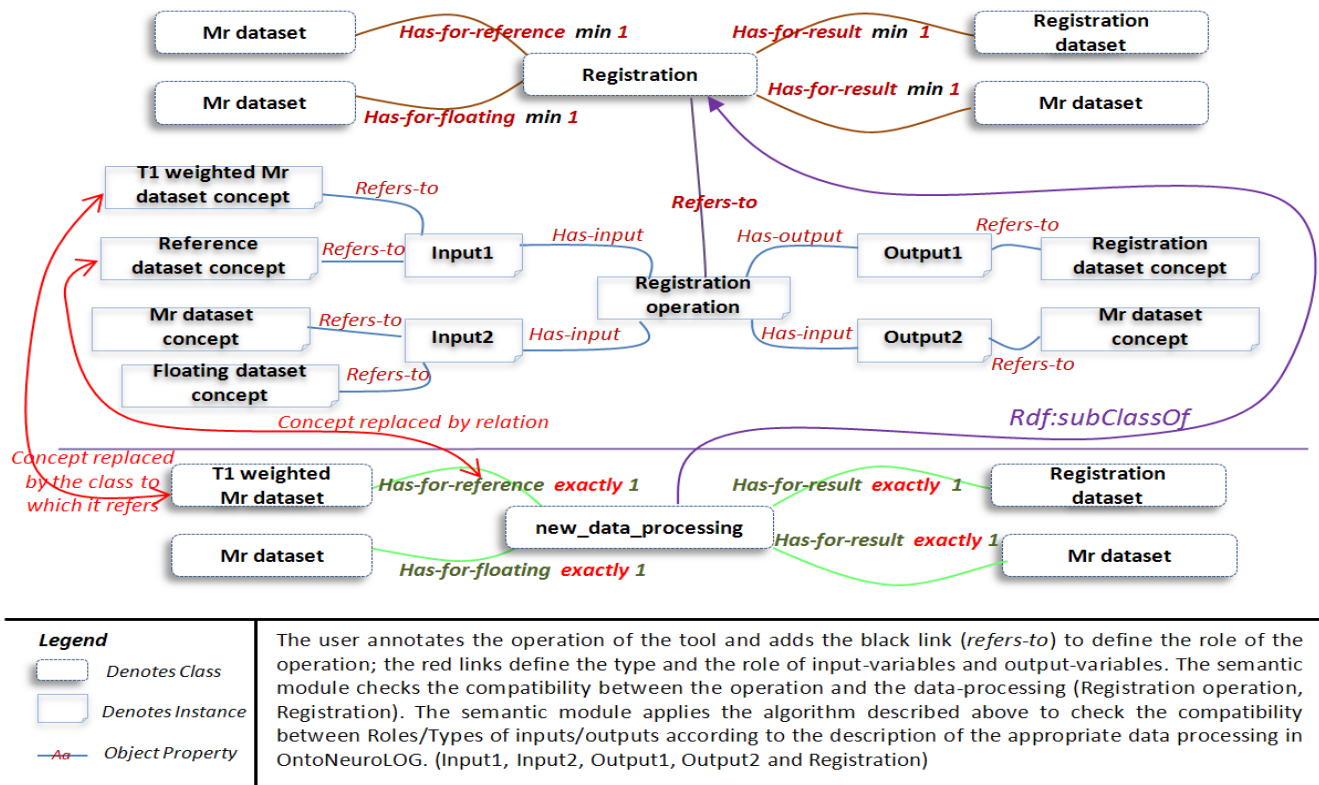


Figure 5: Semantic verification between operation and data processing annotation

4. IMPLEMENTATION AND RESULTS

Integration in the NeuroLOG middleware: Figure 6 depicts the NeuroLOG platform architecture with a particular focus on the sharing and invocation of neuroimaging processing tools. This deployment shows three collaborating sites A, B, C and an end-user interacting with his/her proper gateway (Site A) through the client application. Processing tools are syntactically described and instrumented as relocatable bundles through jGASW [4] to enable their deployment and invocation on various computing infrastructure. The MOTEUR2 [5] component enables the design of new experiments as scientific workflows and is responsible for their enactment.

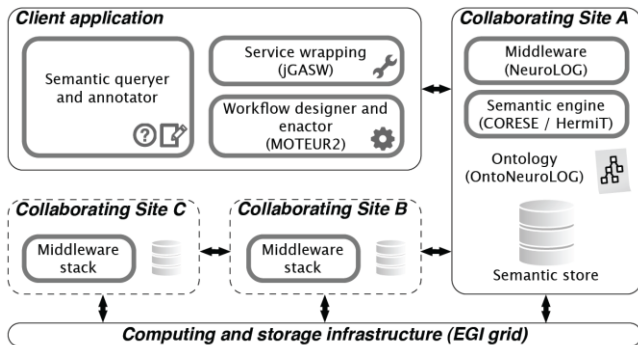


Figure 6: NeuroLOG Architecture

Semantic treatments proposed in section (B) are called either by the user interface (tab dedicated to the semantic annotation of processing tools), or by the workflow enactor (MOTEUR2). Semantic annotations are managed through local RDF triple stores implemented with the Jena API.

Implementation of semantic services: The « Semantic annotation of a processing tool » is accessible through the GUI and enables a user to load the jGASW or GWENDIA descriptor and present the taxonomy of the dataset and data processing concepts, so that the user can select them and associate them to the web service operation being annotated.

This operation uses the HermiT reasoner in order to check the satisfiability of the class of data processing defined by the user.

Here some RDF triples representing the semantic description with orchestration of two jGASW services (enrichment of wfEx12V01.gwendia file);

```

<!-- &wf;baseDocument-wfEx12V01.gwendia_416 -->
<ws:BaseDocument rdf:about="&wf;baseDocument-wfEx12V01.gwendia_416">
  <ws:has-location
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">/home/.../wfEx12V01.gwendia</ws:has-location>
  <ws:has-argument rdf:resource="&wf;inputArgument-Ex1input1_417"/>
  <ws:has-argument rdf:resource="&wf;inputArgument-Ex2input2_419"/>
  <ws:has-argument rdf:resource="&wf;outputArgument-output1Ex2_421"/>
  <ws:has-argument rdf:resource="&wf;outputArgument-output2Ex2_423"/>
</ws:BaseDocument>
<!-- &wf;input-variable_Ex1input1_418 -->

```

```

<ws:input-variable rdf:about="&wf;input-
variable_Ex1input1_418">
<iec:refers-to rdf:resource="&ws;#T1-weighted-MR-
template-dataset-concept_ind"/>
<ws:is-involved-as-input
rdf:resource="&wf;I2Imapping_428"/>
<ws:refers-to-argument rdf:resource="&wf;inputArgument-
Ex1input1_417"/>
<ws:is-input-of
rdf:resource="&wf;operation_wfEx123V01_427"/>
</ws:input-variable>
  <!-- &wf;inputArgument-Ex1input1_417 -->
<ws:InputArgument rdf:about="&wf;inputArgument-
Ex1input1_417">
<ws:has-id
rdf:datatype="http://www.w3.org/2001/XMLSchema#stri
ng">Ex1input1</ws:has-id>
<ws:is-argument-of rdf:resource="&wf;baseDocument-
wfEx12V01.gwendia_416"/>
<ws:refers-to-variable rdf:resource="&wf;input-
variable_Ex1input1_418"/>
</ws:InputArgument>
  <!-- &wf;operation_wfEx123V01_427 -->
<ws:ws-operation
rdf:about="&wf;operation_wfEx123V01_427">
<rdf:type rdf:resource="&ws;#ws-operation"/>
<iec:refers-to rdf:resource="&ws;#de-noising-
concept_ind"/>
<ws:has-input rdf:resource="&wf;input-
variable_Ex1input1_418"/>
<ws:has-input rdf:resource="&wf;input-
variable_Ex2input2_420"/>
<ws:has-orchestration
rdf:resource="&wf;orchestration_wfEx123V01_433"/>
<ws:has-output rdf:resource="&wf;output-
variable_output1Ex2_422"/>
<ws:has-output rdf:resource="&wf;output-
variable_output2Ex2_424"/>
</ws:operation>
  <!-- &wf;orchestration_wfEx123V01_433 -->
<ws:orchestration
rdf:about="&wf;orchestration_wfEx123V01_433">
<ws:uses rdf:resource="&ex1-1;operation-ex1-
1.0.0_395"/>
<ws:uses rdf:resource="&ex2-1;operation-ex2-
1.0.0_1239"/>
<ws:has-mapping rdf:resource="&wf;I2Imapping_428"/>
<ws:has-mapping rdf:resource="&wf;I2Imapping_430"/>
<ws:has-mapping rdf:resource="&wf;O2Imapping_429"/>
<ws:has-mapping rdf:resource="&wf;O2Omapping_431"/>
<ws:has-mapping rdf:resource="&wf;O2Omapping_432"/>

```

```

<ws:is-orchestration-of
rdf:resource="&wf;operation_wfEx123V01_427"/>
</ws:orchestration>

```

This annotations block represents semantic annotations of a GWENDIA file describing a composite service; this service uses two elementary services (the first one is the De-noising service and the second is Segmentation service, their description is not presented here). We use CORESE to retrieve the semantic annotations of internal services from the semantic repository. We remark here that there is no grounding of operation but there are groundings of inputs, outputs and necessarily the GWENDIA file (basedocument). This grounding is sufficient to build an orchestration. In fact, mappings are detected from the GWENDIA file. The « Validation of an orchestration » operation also proceeds using the HerMiT reasoner. The validation of the consistency of the operation and the data-processing referred by this operation is done using the OWL API and the HerMiT reasoner.

The « Invocation of a processing tool » operation uses the semantic search engine CORESE [33] to retrieve the semantic annotations associated to the datasets (i.e. images) selected by the user. Then it uses HerMiT to check whether the classes to which the selected datasets belong are subsumed by those specified in the processing tool annotations, concerning the related input variable. We query the semantic repository using CORESE to select the semantic annotations of inputs.

```

<?xml version="1.0"?>
<rdf:RDF
xmlns="http://www.irisa.fr/wfEx12V01.gwendia"
  xml:base="http://www.irisa.fr/wfEx12V01.gwendia"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#"
  xmlns:DBfile="http://www.irisa.fr/DBfile.rdf#"
  xmlns:wf="http://www.irisa.fr/wfEx12V01.gwendia#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:ws="http://www.irisa.fr/web-service-owl-
lite.owl#"
  xmlns:iec="http://www.irisa.fr/iec-owl-lite.owl#"
<owl:Ontology rdf:about="&wf;"/>
  <!-- &wf;execution_wfEx123V01_427 -->
<ws:execution
rdf:about="&wf;execution_wfEx123V01_427">
<ws:is-instrument-of
rdf:resource="&wf;operation_wfEx123V01_427"/>
<ws:has-for-data-at rdf:resource="&wf;input-
vv_Ex1input1_418"/>
<ws:has-for-data-at rdf:resource="&wf;input-
vv_Ex2input2_420"/>
<ws:has-for-result-at rdf:resource="&wf;output-
vv_output1Ex2_422"/>

```



```

<ws:has-for-result-at rdf:resource="&wf;output-
vv_output2Ex2_424"/>
</ws:excution>
  <!-- &wf;input-vv_Ex1input1_418 -->
<ws:variable-value rdf:about="&wf;input-
vv_Ex1input1_418">
<iec:refers-to rdf:resource="&DBfile;#Image1.nii"/>
<ws:is-input-value-of rdf:resource="&wf;input-
variable_Ex1input1_418"/>
</ws:variable-value>
  <!-- &wf;input-vv_Ex2input2_420 -->
<ws:variable-value rdf:about="&wf;input-
vv_Ex2input2_420">
<iec:refers-to rdf:resource="&DBfile;#Image2.nii"/>
<ws:is-input-value-of rdf:resource="&wf;input-
variable_Ex2input2_420"/>
</ws:variable-value>
  <!-- &wf;output-vv_output1Ex2_422 -->
<ws:variable-value rdf:about="&wf;output-
vv_output1Ex2_422">
<iec:refers-to
rdf:resource="&DBfile;#registeredimage1.nii"/>
<ws:is-ouptut-value-of rdf:resource="&wf;output-
variable_output1Ex2_422"/>
</ws:variable-value>
  <!-- &wf;output-vv_output2Ex2_424 -->
<ws:variable-value rdf:about="&wf;output-
vv_output2Ex2_424">
<iec:refers-to rdf:resource="&DBfile;#outputImage2.nii"/>
<ws:is-output-value-of rdf:resource="&wf;output-
variable_output2Ex2_424"/>
</ws:variable-value>
</rdf:RDF>

```

The precondition and effects or postconditions are in CORESE format: Example of postcondition:

Use case: After the invocation of the registration tool we need to save in our semantic repository that the resulting registered image concerns the same subject or group of subjects as the floating image used in the registration process. The rule aims at providing more semantics about generated data that registration tools cannot provide, since they focus on the geometrical problem of registration only.

Semantically the rule has as type “refersTo” (discussed section 3.1 pre/postcondition annotations) types are (predefined in the knowledgebase by the specialist) and is described as: The *Output1* must refer to (refers to data property) the same subject or group of subjects as *Input1* does.

tool: <http://www.irisa.fr/registration.owl> (description of the tool);
ws: <http://www.irisa.fr/web-service-owl-lite.owl> (Ontology of web services);
dp:<http://www.irisa.fr/data-processing-owl-lite.owl>

(Ontology of data processing) rs: <http://www.irisa.fr/resource.owl> (temporary resources extracted from semantic database (instances of dataprocessing concepts...))

tool:postC1 ws:has-type “refersTo”

tool:postC1 ws:concerns dp:Registration

tool:postC1 ws:concerns tool:input1

tool:postC1 ws:concerns tool:output1

Rules are expressed according to the CORESE format. They allow adding metadata to the newly created data, through a three-step process: (1) adding concerned RDF triples, (2) generating the rule in CORESE format and (3) applying the rule. The resulting metadata can be queried by means of a specific query. CORESE rules are made of simply two blocks (if-then). The first contains a SPARQL query to select the concerned RDF triples and the second generates the annotations.

For example, to generate the if-block of the rule described above we create the RDF triple for selecting the value used for such variable-value (ex.: *tool:postC1 ws:concerns tool:input1* → *tool:input1 ws:has-value ?inputvalue1*), to select the dataset affected at execution time to this value we add the triple (*?inputvalue1 iec:refers-to ?dsinputvalue1*) and to select the subject or the group of subjects we add (*?dsinputvalue1 iec:refers-to ?particular1*)

Those RDF triples are generated automatically from the description of the rule.

The next block describes the rule in CORESE format:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY cos "http://www.inria.fr/acacia/corese#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY ws "http://www.irisa.fr/web-service-owl-lite.owl#">
<!ENTITY tool "http://www.irisa.fr/registration.owl#">
<!ENTITY iec "http://www.irisa.fr/iec-owl-lite.owl#"> ]>
<rdf:RDF
xmlns:rdfs="&rdfs;"xmlns:rdf="&rdf;"xmlns:cos="&cos;"
xmlns:owl="&owl;"xmlns:xsd="&xsd;"xmlns:ws="&ws;"xmlns:
:tool="&tool;"xmlns:iec="&iec;"><cos:rule cos:name =
'&tool;#postC1'><cos:if>
<!-- According to the input concerned (&tool;#postC1
ws:concern tool:input1) we select the variable value created
while execution -->
tool:input1 ws:has-value ?inputvalue1
<!-- For inputvalue1 we select the image (dataset) selected by
the user (images are already annotated and stored in the
knowledgebase) -->
?inputvalue1 iec:refers-to ?dsinputvalue1

```

```

<!-- For dsinputvalue1 we select the particular (subject or
group of subject) to which it refers to-->
?dsinputvalue1 iec:refers-to ?particular1

<!--      http://www.irisa.fr/web-service_owl-lite#concern
tool:output1 According to output concerned (output1) we
select the variable value created while execution -->
tool:output1 ws:has-value ?outputvalue1

<!-- For outputvalue1 we select the image created and saved
in the knowledge base after execution of the tool Registration
-->
?outputvalue1 iec:refers-to ?dsoutputvalue1 }
</cos:if><cos:then>
{ <!-- According to selected values we add meta data that
consist in (the image generated must refers to the same
subject or group of subject (particular1) that the input1 refers
to -->
?dsoutputvalue1 iec:refers-to ?particular1 }
</cos:then></cos:rule>
</rdf:RDF>

```

Figure 7: How to generate rules from semantic annotations

Every pre/postcondition contains its own SPARQL query which is automatically generated from its annotations, for example:

```

PREFIX reg: <http://www.irisa.fr/registration.owl #>
PREFIX ws: <http://www.irisa.fr/web-service-owl-lite.owl #>
PREFIX iec: < http://www.irisa.fr/iec-owl-lite.owl #>
Select * where {reg:postC1 ws:concerns ?outputvariables
?outputvariables rdf:type ws:output-variable
?outputvariables ws:has-value ?outputvariablevalues
?outputvariablevalues iec:refers-to ?datasets?datasets
iec:refers-to ?particular }

```

If in the semantic repository there are RDF triples that associate the input image to the subject subject1, for example:

At annotation time:

```

tool:registration-operation ws:has-input tool:input1
tool:registration-operation ws:has-output tool:output1

```

At execution time:

```

tool:input1 ws:has-value rs:valuesinput1
rs:valuesinput1 ws:refers-to rs:dataset1 (given by the user)

```

Triples generated after execution:

```

tool:output1 ws:has-value rs:valuesoutput1
rs:valuesoutput1 rs:refers-to rs:registered-dataset1 (created)

```

In the knowledge base we have the fact:

```
rs:dataset1 iec:refers-to rs:subject1
```

then, the metadata created should be:

```
rs:registered-dataset1 iec:refers-to rs:subject1
```

5. DISCUSSION

The reuse of processing tools in federated systems is hampered by the heterogeneity of their implementation. Encapsulation into WS allows invocation aspects to be homogenized. Semantic heterogeneity is a more complex issue. NeuroLOG is addressing it through the definition of an application ontology that provides explicit definition of both data processing classes and of the dataset classes to which such processing apply or from which they result. In addition, NeuroLOG uses this notion of service to achieve a clearer exposition of the function of processing resources. For example, the various capabilities of image registration software may lead to defining several services, each of them exploiting a particular set of capabilities toward a single function, e.g. anatomical normalization to a template, or rigid registration of images belonging to a single subject. This approach allows the service interface to be simplified and the application context to be more clearly understood. The explicit reference (via the semantic annotations) to a data processing class, together with the specification of allowed values for input data provide additional capabilities to ensure that the essential constraints for an appropriate use are actually fulfilled. This should avoid misuse and then facilitate relevant reuse by non-specialist users.

Our model of WS shares an underlying conceptualization with models submitted to W3C (OWL-S, and WSMO), however its modularization is different as a consequence of a different modeling strategy. In line with e.g. [34], our aim is to define rigorously the concepts at stake by specializing a set of already defined concepts in an ontological framework structured by the foundational ontology DOLCE. Moreover, this model is consistently integrated in a larger one including a model of processed data (datasets) to account for the latter's provenance.

OWL-S and WSMO are still a W3C member submission designed to clarify the semantics of Web services. OWL-S aims to improve the expressiveness of web services and add semantic reasoning capability in order to discover, invoke, compose and manage Web services. WSMO tries to overcome the problems of heterogeneous data and protocols with mediators. The main objective of WSMO is to automate most of the discovery process, selection, composition, mediation and execution of web services. WSMO adds thereto the following objective: A strong decoupling between components and a central role in mediation. One of the fundamental principles of WSMO consists of the total separation between the different elements involved in the composition of Web services.

WSMO and OWL-S use sequence and control construct to manage workflow execution over the time and the order of execution. Here the jGASW and the MOTEUR software actually do that. Thus we do not need control construct and sequences in our model. Eventually, semantic invocation is triggered by MOTEUR, indeed, it has a pointer to the IRIs of the ws-operation instance for every execution of a service or workflow that will be executed over a workflow. W3C submissions are complicated and oblige us to use some technical detail that we do not need like lists and sequences. Similarly, WSMO is generally used for the mediation and is not adequate with our work. However, our model is very sampler and can be reused. It does not oblige someone to use mandatory the reasoning mechanisms also it lets the possibility to be enriched by other ontologies over the interface class.

WSMO defines orchestration to: (a) ensure behavioral aspect while composing web services, by matching them using their

declared choreography, (b) facilitate the reuse of service combinations, and (c) enable client constraints to be checked [39]. Therefore, a choreography in WSMO describes the behavior of the Web service from a client's point of view. The client may be either a human user or an agent acting on behalf of a user.

The SOPHIE solution [40] (Semantic web services choreography service) is also a choreography description that follows WSMO. It tries to overcome heterogeneity by adding some solutions to overcome the lack of technological independence, the lack of clear structural, behavioral and operational models, and heterogeneity of message exchange. For example, messages can be sent/received in a different order than expected, or can be non-compliant with respect to the expected behavior of the other party (sequence and cardinality mismatches), the structure or the format of exchanged messages can be incompatible (structural incompatibility), or different terminological conventions have been used for representing encoded concepts (semantic incompatibility) [41]. SOPHIE introduces business alliance. The common goal of such semantic approaches for business alliances is to automate interoperability processes between heterogeneous businesses which are providing various information by referring to their own knowledge structures. This framework acts without paying attention to the control flow or to nature of processed service. In contrast, our work deals with this aspect through the validation algorithms especially the verification between operation and data-processing class. However, the improvements for choreography in SOPHIE can be adopted and used for the automatic selection and discovery in the context of our framework, essentially, cardinality and format of exchanged messages.

In the neuroimaging area, web services involve specific input or output data; for example to detect brain tumors, de-noise and realign MRI images, thus, WSDL based on poorly formatted XML documents are not expressive enough for the building of biomedical workflows chaining multiple web services seamlessly and in an interoperable way. On the other hand, SOA-based web services are still a means to integrate web services; however, it still lacks semantics and composition to ensure their interoperability. Furthermore, our proposition avoids enriching jGASW WSDL because they have a specific format and are understandable only by jGASW Engine. Thus specifications like OWL-S and WSMO could not use this kind of WSDL.

As seen earlier, most standard specifications have grounding based on WSDL documents, generally enriched by XML or other kinds of documents. Such an approach was not sufficient to meet the neuro-imaging domain expectations. To overcome this problem, we enhanced the semantically poor standard WSDL files created dynamically by jGASW (for single tools), and similarly we enhanced the GWENDIA files generated by MOTEUR (for composite tool) thus adding the necessary semantics based on our web services ontology, in order to enable the reasoning that was needed to safely invoke and compose our services.

By rules we have extended the expressivity in two levels: first, the annotations second the execution; contrary to others works which use rules for the semantic behavioral aspect we tried simply to add more semantics to neuroimaging knowledge base. New rules added to the semantic annotation of services to supplement the knowledge base, may also lead unpredictable results for example add new roles or nature of some images saved in the relational databases.

In the neuroimaging area web services have sometimes complex input or output data structure and a large computation time; for example to (detect brain tumors, de-noise and realign MRI images...), thus, WSDL based on poorly formatted XML document are not expressive enough for the building of biomedical workflows by chaining multiple web services seamlessly and make them interoperable. Similarly, SOA-based web services are still a means to integrate web services and make them interoperable; however, they still lack semantics and composition is out of their focus. Furthermore, our proposition avoids enriching jGASW WSDL because they have a specific format and are understandable only by jGASW Engine. Thus specifications like OWL-S and WSMO does not support this kind of WSDL. We then choose to deal with jGASW descriptors and MOTEUR) designed specifically for the biomedical background.

Taverna [27], seahawk [37], MOWserv [38] and many other tools enable building of workflows by connecting BioMoby [26] services in the bioinformatics area, and apply them in the neuroimaging domain. Although, the degree of complexity and heterogeneity of data and workflow constructions in the both domains are not the same. For us we do not want to miss the features, capability and flexibility of jGASW and MOTEUR software. They are a key consideration in the construction of image processing web services and workflows. For this reason we have chosen this strategy of annotating semantically their generic descriptors.

Web service composition has been the center of interest of manifold projects. W3C standards are ubiquitous and sometimes considered as a panacea for addressing many interoperability and standardization problems over the web services sharing and composition. However, they lack many notions which should be present in biomedical domain. Compared to our work, the OWL-S specification acts as a composer of workflow, rather than a monitoring tool, and this has a profound effect on how the specification is designed, the features it offers, and what neuroimaging expectations might be. Our approach does not meet OWL-S solutions.

DAML-S and WSMF do not address the issue of services composition and they do not consider purpose, parameter unit and business roles. Our approach address the issue of composition by giving a semi-automatic composition of neuroimaging web services and consider business roles and parameter unit throw data processing definition and verifications algorithms.

Our approach, from a conceptual corner of view, consist in proposing semantic web services ontology model whose main classes are grounded to foundational ontology DOLCE. It would allow to smartly articulate the model and domain ontologies OntoNeuroLOG based on foundational ontologies, and thus exploit these ontologies at design and execution runtime. And we benefit here from rich and well-documented axiomatization as to location in space and time, dependence and parthood, and to the fact that it relies on explicit structuration principles in the DOLCE foundational ontology firstly and in OntoNeuroLOG secondly. We address also the possibility to reason about entities and to map ontologies in the future. This ontology integration task could also cover the semantic mapping between input-variable and variable-value in the model and OntoNeuroLOG Datasets. Indeed considering our approach from an ontology design perspective, a significant effort.

Much more tests are still needed to assess the added value of this approach; like the use of the NeuroLOG semantic module to

query semantic data repository, or like the manual visualization by specialists of provenance information accorded to the descriptions of output images processed by image processing tools.

Semantically, according to the proposed model, we can compose workflows by considering them as single service having composite operation. Unfortunately, the MOTEUR tool does not take in account the possibility to do that because when a workflow is edited it is flattened to its atomic components (jGASW services). However, tool like OWL-S enable this kind of use.

6. CONCLUSION

In conclusion, we are convinced that the semantic annotation of processing tools shared as web services is a key factor to stimulate and facilitate their appropriate re-use and interoperation in federated systems. This paper presents a new model for the semantic annotation of image processing tools and some semantic processing based on these annotations for services validation at design and execution time. The motivation for building a new model is the need to rely on domain ontology and the related foundational ontology. The second major contribution of this work is the implementation of some semantic services for checking consistency of workflows at design time and rule-based creation of semantic metadata associated to results of processing tools execution.

Future work should deal with improving the reasoning mechanisms to ensure automatic selection and discovery of jGASW services. This step should rely on the aforementioned algorithms; take in account the aforementioned crucial points like similarity detection, choreography and semantic patterns.

7. ACKNOWLEDGMENTS

The NeuroLOG project was funded by the French National Agency for Research (ANR-06-TLOG-024). The authors gratefully acknowledge all contributors, including our clinical colleagues providing the image data exploited in the NeuroLOG testbed. I thank very much Especially, Alban Gaignard, Franck Michel, Javier Rojas Balderrama and Johan Montagnat from I3S laboratory and Gilles Kassel from MIS laboratory, who have sacrificed time, efforts and experiences for my support during the various stages of design and implementation of this work.

8. REFERENCES

- [1] Martone, M.E. Gupta A. and Ellisman, M.H. E-neuroscience: challenges and triumphs in integrating distributed data from molecules to brains, *Nat Neurosci.* 7, 5 (2004), 467-72.
- [2] Michel, F. Gaignard, A. Ahmad, F. Barillot, C. et al. Grid-wide neuroimaging data federation in the context of the NeuroLOG project, *Studies in Health Technology and Informatics*, 159 (2010), 112-23.
- [3] Temal, L. Dojat, M. Kassel, G. Gibaud. B. 2008. Towards an ontology for sharing medical images and regions of interest in neuroimaging. *J. of Biomedical Informatics* 41, 5 (October 2008), 766-778. DOI=[10.1016/j.jbi.2008.03.002](https://doi.org/10.1016/j.jbi.2008.03.002)
- [4] Balderrama, J. R. Montagnat, J. Lingrand, D. 2010. jGASW: A Service-Oriented Framework Supporting HTC and Non-functional Concerns. In *Proceedings of the 2010 IEEE International Conference on Web Services (ICWS '10)*. IEEE Computer Society, Washington, DC, USA, 691-694. DOI=[10.1109/ICWS.2010.59](https://doi.org/10.1109/ICWS.2010.59)
- [5] Glatard T. Montagnat, J. Lingrand, D. and Pennec, X. 2008. Flexible and Efficient Workflow Deployment of Data-Intensive Applications On Grids With MOTEUR. *Int. J. High Perform. Comput. Appl.* 22, 3 (August 2008), 347-360. DOI=[10.1177/1094342008096067](https://doi.org/10.1177/1094342008096067)
- [6] Tsalgatidou, A., and Pilioura, T. 2002. An Overview of Standards and Related Technology in Web Services. *Distrib. Parallel Databases* 12, 2-3 (September 2002), 135-162. DOI=<http://dx.doi.org/10.1023/A:1016599017660>
- [7] Dang, J. Hedayati, A. Hampel, K. and Toklu, C. 2008. An ontological knowledge framework for adaptive medical workflow. *J. of Biomedical Informatics* 41, 5 (October 2008), 829-836. DOI=[10.1016/j.jbi.2008.05.012](https://doi.org/10.1016/j.jbi.2008.05.012)
- [8] Dustdar, S. and Schreiner, W. 2005. A survey on web services composition. *Int. J. Web Grid Serv.* 1, 1 (August 2005), 1-30. DOI=[10.1504/IJWGS.2005.007545](https://doi.org/10.1504/IJWGS.2005.007545)
- [9] Papazoglou, M-P. and Heuvel, W-J. 2007. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal* 16, 3 (July 2007), 389-415. DOI=[10.1007/s00778-007-0044-3](https://doi.org/10.1007/s00778-007-0044-3)
- [10] Goble, C. and Stevens, R. 2008. State of the nation in data integration for bioinformatics. *J. of Biomedical Informatics* 41, 5 (October 2008), 687-693. DOI=[10.1016/j.jbi.2008.01.008](https://doi.org/10.1016/j.jbi.2008.01.008)
- [11] Khusidman, V. 2006. *SOA Enabled Workflow Modernization [EBOL]*, BP Trends (Oct.2006), <http://www.bptrends.com/publicationfiles/10-06-ART-SOA-Enabledworkflow-Khusidman2.pdf>.
- [12] Navas-Delgado, I. Del Mar Rojano-Muñoz, M. Ramírez, S. Pérez, A-J. León, E.A. Aldana-Montes, J.F and Trelles, O. 2006. Intelligent client for integrating bioinformatics services. *Bioinformatics* 22, 1 (January 2006), 106-111. DOI=[10.1093/bioinformatics/bti740](https://doi.org/10.1093/bioinformatics/bti740)
- [13] Korotkiy, M. Towards an ontology-enabled service-oriented architecture. Technical report, Vrije Universiteit Amsterdam. URL: <http://www.cs.vu.nl/~maksym/pap/Onto-SOA-ICSOC.pdf>.
- [14] Korotkiy, M. and Top, J. 2006. Onto-SOA: From Ontology-enabled SOA to Service-enabled Ontologies. In *Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW '06)*. IEEE Computer Society, Washington, DC, USA, 124-.
- [15] Vitvar, T. Kopecky, V. Viskova, J. and Fensel, D. 2008. WSMO-lite annotations for web services. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications (ESWC'08)*, Berlin, Heidelberg, 674-689
- [16] Anders, T. Curbera, F. Dholakia, H. Golland, Y. Klein, J. Leymann, F. Roller, D. Smith, D. Thatte, S. Trickovic, I. and Weerawarana, S. *Business Process Execution Language for Web Services*, Version 1.1. <http://www.ibm.com/developerworks/library/ws-bpel/>, 2002.
- [17] Beerli, C. Eyal, A. Kamenkovich, S. and Milo, T. 2005. Querying business processes with BP-QL. In *Proceedings of the 31st international conference on Very large data bases (VLDB '05)*. VLDB Endowment 1255-1258.

- [18] Burstein, M.H. Hobbs, J.R. Lassila, O. Martin, D. McDermott, D.V. McIlraith, S.A. Narayanan, S Paolucci, M. Payne, T.R. and Sycara, K.P. 2002. DAML-S: Web Service Description for the Semantic Web. In *Proceedings of the First International Semantic Web Conference on The Semantic Web* (ISWC '02), Ian Horrocks and James A. Hendler (Eds.). Springer-Verlag, London, UK, UK, 348-363.
- [19] Miller, J., Verma, K., Rajasekaran, P. Sheth, A. Aggarwal, R. Sivashanmugam, K. *WSDL-S: Adding semantics to WSDL - White paper*, Technical Report, LSDIS Lab, University of Georgia.
- [20] Mokarizadeh, S. Küngas, P. and Matskin, M. 2010. Ontology learning for cost-effective large-scale semantic annotation of web service interfaces. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses* (EKAW'10), Philipp Cimiano and H. Sofia Pinto (Eds.). Springer-Verlag, Berlin, Heidelberg, 401-410.
- [21] Klusch, M. and Kaufer, F. 2009. WSMO-MX: A hybrid Semantic Web service matchmaker. *Web Intelli. and Agent Sys.* 7, 1 (January 2009), 23-42.
DOI=[10.3233/WIA-2009-0153](https://doi.org/10.3233/WIA-2009-0153)
- [22] Lee, Y. Patel, C. Chun, S. A. and Geller J. 2004. Towards Intelligent Web Services for Automating Medical Service Composition. In *Proceedings of the IEEE International Conference on Web Services* (ICWS '04). IEEE Computer Society, Washington, DC, USA, 384-.
DOI=[10.1109/ICWS.2004.115](https://doi.org/10.1109/ICWS.2004.115)
- [23] Astakhov, V. Gupta, A. Santini s. and Grethe J.S. 2005. Data Integration in the Biomedical Informatics Research Network (BIRN), In: (B.Ludäscher, and L Raschid eds.) In *proceedings of the Second International Workshop, Data Integration in Life Sciences* (San Diego, CA, USA, July 20-22, 2005), Lecture Notes in Computer Science, 3615-317.
- [24] Santini, S. Gupta, A. 2003. The role of Internet images in the biomedical informatics research network. In *Proceedings of SPIE* (San Jose, CA 5018, 2003). Internet Imaging.
- [25] DiBernardo, M. Pottinger, R. and Wilkinson, M. 2008. Semi-automatic web service composition for the life sciences using the BioMoby semantic web framework. *J. of Biomedical Informatics* 41, 5 (October 2008), 837-847.
DOI=[10.1016/j.jbi.2008.02.005](https://doi.org/10.1016/j.jbi.2008.02.005)
- [26] Kawas, E.A. Senger, M. and Wilkinson, M.D. 2006. BioMoby extensions to the Taverna workflow management and enactment software. *BMC Bioinformatics*, 7 (2006), p. 523+.
- [27] Hull, D. Wolstencroft, K. Stevens, R. Goble, C. Pocock, M. Li, P. Oinn, T. Zhang, N. Yao, L. Nenadic, A. Chin, J. Rector, A. Chadwick, D. Otenko, S. and Shi, Q. 2006. "Taverna: a tool for building and running workflows of services Achieving Fine-grained Access Control in Virtual Organisations," *Nucleic Acids Research*, 2006.
- [28] Stevens, R. D. Robinson, A. J. and Goble, C. A. 2003. myGrid: personalised bioinformatics on the information grid. *Bioinformatics*, 19:i302-i304, 2003.
- [29] De Roure, D. Goble, C. and Stevens, R. 2009. The design and realisation of the Experimentmy Virtual Research Environment for social sharing of workflows. *Future Gener. Comput. Syst.* 25, 5 (May 2009), 561-567.
DOI=[10.1016/j.future.2008.06.010](https://doi.org/10.1016/j.future.2008.06.010)
- [30] Boeker, M. Stenzhorn, H. Kumpf, K. Bijlenga, P. Schulz, S. Hanser, S. 2007. The @neurIST Ontology of Intracranial Aneurysms: Providing Terminological Services for an Integrated IT Infrastructure. In *Proceedings of AMIA 2007 Annual Symposium* (Chicago, USA, November 2007), AMIA 2007.
- [31] Iacono, L. L. Rajasekaran, H. 2008. Security Architecture for Distributed Medical Information Systems. In *Proceedings of GI Jahrestagung* (1)2008. pp.110-116
- [32] Benkner, S. Arbona, A. Berti, G. Chiarini, A. Dunlop, R. Engelbrecht, G. Frangi, A.F. Friedrich, C.M. Hanser, S. Hasselmeyer, P. Hose, R.D. Iavindrasana, Köhler, J.M. Lo Iacono, L. Lonsdale, G. Meyer, R. Moore, B. Rajasekaran, H. Summers, P.E. Wöhrer, A. and Wood, S. 2010. @neurIST: infrastructure for advanced disease management through integration of heterogeneous data, computing, and complex processing services. *Trans. Info. Tech. Biomed.* 14, 6 (November 2010), 1365-1377.
DOI=[10.1109/TITB.2010.2049268](https://doi.org/10.1109/TITB.2010.2049268)
- [33] Corby, O. Dieng-Kuntz, R. Faron-Zucker, C. and Gandon, F. 2006. Searching the Semantic Web: Approximate Query Processing Based on Ontologies. *IEEE Intelligent Systems* 21, 1 (January 2006), 20-27.
DOI=[10.1109/MIS.2006.16](https://doi.org/10.1109/MIS.2006.16)
- [34] Oberle, D. Lamparter, S. Grimm, S. Vrandêlê, D. Staab, S. and Gangemi. 2006. Towards ontologies for formalizing modularization and communication in large software systems. *Appl. Ontol.* 1, 2 (April 2006), 163-202
- [35] Eshuis, R. and Grefen, P. 2007. Structural Matching of BPEL Processes. In *Proceedings of the Fifth European Conference on Web Services* (ECOWS '07). IEEE Computer Society, Washington, DC, USA, 171-180.
DOI=[10.1109/ECOWS.2007.26](https://doi.org/10.1109/ECOWS.2007.26)
- [36] Zhao, X. Liu, E. Clapworthy, G. J. Viceconti, M. and Testi, D. 2010. SOA-based digital library services and composition in biomedical applications.
DOI: [10.1016/j.cmpb.2010.08.009](https://doi.org/10.1016/j.cmpb.2010.08.009)
- [37] Gordon, P.M.K. and Sensen, C.W., Seahawk: moving beyond HTML in web-based bioinformatics analysis. *BMC Bioinformatics*. v8. 208+
- [38] Ramirez, S., Muñoz-Merida, A., Karlsson, J., Garca, M., Pérez-Pulido, A. J., Claros, M. G., and Trelles, O. MOWServ: a web client for integration of bioinformatic resources *Nucleic Acids Research*, 2010, 38, 671-676.
- [39] C. Pedrinaci, J. Domingue, and A. Sheth, "Semantic Web Services," Handbook of Semantic Web Technologies, J. Domingue, D. Fensel, and J. Hendler, eds. Springer, preprint, 29 Dec. 2010
- [40] Arroyo, S., Duke, A. Manuel, J. Cobo, L., and Sicilia, M. A. 2007. A model-driven choreography conceptual framework. *Comput. Stand. Interfaces* 29, 3 (March 2007), 325-334.
DOI=[10.1016/j.csi.2006.05.004](https://doi.org/10.1016/j.csi.2006.05.004)
- [41] Arroyo, S., and Sicilia, M. A. 2008. SOPHIE: Use case and evaluation. *Inf. Softw. Technol.* 50, (12 November 2008), 1266-1280.
DOI=[10.1016/j.infsof.2008.01.001](https://doi.org/10.1016/j.infsof.2008.01.001)