A Flexible Framework for Adaptive Knowledge Retrieval and Fusion for Kiosk Systems and Mobile Clients

Simon Bergweiler

German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Email: simon.bergweiler@dfki.de

***Abstract*—This approach describes a centralized framework that offers a flexible integration and access to different external knowledge sources via a single query interface. The advantage of this approach is to use the potential of a combination of different services to focus information and knowledge. Different SOAP or REST-based Web services can be requested in parallel and their results are integrated, analyzed and harmonized to one result structure that is sent to the user's client application. Various knowledge sources, can be integrated in the framework, without the need to know a query language of the Semantic Web like SPARQL. A special point in this approach is the discovery of services and the harmonization of the results of involved services using matching and mapping rules.**

Keywords–Semantic Web Services; OWL-S; WADL.

I. INTRODUCTION

The current paradigm of service-oriented architectures, that are highly modular, adaptable, distributed, and thus scalable leads to an increasing distribution and multi-disciplinarity of systems in today's Internet [1]. Nowadays, factual knowledge, such as pictures and videos are made accessible from anywhere through so-called cloud services. However, the distributed nature and the wide range of these services prove to be a problem. The variety of services and their different technical configurations through non-standardized custom Web interfaces (APIs) cannot be used easily from the consumer's point of view. Industry and manufacturers also recognized this and provide client applications, but only in a very static form and closely tight to their own published database, and usually in the form of mobile client apps for tablets and smartphones. However, this is precisely the problem: these client applications are customized and adapted to only fit to one specific requirement of a central source of knowledge with its interfaces. Unfortunately, with such a solution, it is not possible to formulate a comprehensive knowledge query to multiple knowledge bases across the specific domain. This is exactly the key point where the solution presented in the discussed approach will play a major role.

This paper gives an overview of the established service framework approach published by Bergweiler [2] and describes an extension and further development of the planning process and the advanced approach of the service discovery process of information sources. The objective of the approach is the design and development of a middleware that integrates information derived from various sources of knowledge. All obtained information is adapted for the particular requesting client. With this solution, new sources can be integrated, removed or modified without stringent dependencies on specific providers of information and their interfaces. Multiple heterogeneous Web services are composed and extracted data sets are analyzed and harmonized to a result set, that is returned to the client. The here presented framework combines the two classes of services: it closes the gap between classical Remote Procedure Calls (RPC) and pure Representational State Transfer (REST)-ful services calls, that binds factual knowledge extracted from Semantic Web Services like Freebase [3] or DBpedia [4], by mapping results and their respective annotations syntactically and semantically well-defined to a domain ontology.

A system prototype was developed, that answers combined requests, such as *"Give me the bordering countries of Germany and the population and some images of their capitals."*. The system decomposes the combined input and maps the respective query parts to connected matching services. After the execution of the service, the results of each query part are integrated and harmonized by means of an internal domain ontology and returned as a combined result to the respective client. In the result structure it is still indicated, from which source the relevant information comes.

The article is structured as follows: Section II gives an overview on related work. Section III describes the approach for the development of this framework for the discovery of services, the information extraction and the harmonization and fusion of all extracted information. Section IV shows how the approach can be adapted to a special domain and Section V gives a conclusion and an outlook on future work.

II. RELATED WORK

For the sake of a better understanding of Semantic Web technologies and their integration in the functioning of the presented framework, these technologies will be described below, along with references to their prototypical implementations.

Web services are defined as Web-based and self-contained software components that allow an interoperable machine-to-machine interaction and communication over networks. There exist two classes of Web services RPC and REST(ful) services. Most of the RPC-style Web services use the Simple Object Access Protocol (SOAP) [5] and the Web Service Description Language (WSDL) to describe the interface in a machine processable format. The data transfer model to which REST is based on, is present in its basic design in the Hypertext Transfer Protocol (HTTP) since the early days of the Internet. REST is used as a lightweight approach for resource-oriented loosely coupled self-contained software components - RESTful services. REST is an architectural style for distributed systems and has a syntactic description called Web Application Description Language (WADL) [6][7].

 ISBN: 978-1-61208-353-7

A. Web Service Description Language

The XML-based Web Service Description Language (WSDL) was originally designed and developed by the companies IBM, Microsoft, and Ariba to provide a standard mechanism for describing Web services in their SOAP toolkits. After the conception a proposed language draft was submitted to the World Wide Web Consortium (W3C) to define a standardized interface definition language (IDL) that defines the communicational aspect of Web services. A specific Web service endpoint with its operations and methods can be described abstractly and how the communication has to be achieved [8]. Here, WSDL strongly binds to SOAP [9] or plain HTTP [10]. Unfortunately, WSDL addresses the technical mechanisms and aspects of Web services, but does not reflect the functionality of a service and it has some strict limitations with its fixed orientation on SOAP. Furthermore, WSDL is used to generate modular source code automatically, such as stubs and skeletons for the service call. Thus, minimal interface changes mean that all parts of the program have to be regenerated until a part of the program can be used again.

The framework discussed here uses WSDL to describe classical Web services, but the new lightweight service architectures rely on another architecture model. For RESTful services this approach needs an additional service description language, the Web Application Description Language.

B. Web Application Description Language

For the description of resource-oriented loosely coupled Web services that follow the REST paradigm, first mentioned by Fielding [11], an XML-based language called WADL was developed by Sun Microsystems [7]. The aim of the development of WADL is the unified description of REST-based services in machine-readable form, in order to make processing easier and simplify the development of tools in the context of Web 2.0. Thus, WADL is the syntactic description of RESTful Web services as WSDL for SOAP-based RPC-services.

There are also efforts to attach annotations to RESTful Web services to provide automatic mediation or composition of services to so-called Web 2.0 mashups. The most common description format is SA-REST [12], Semantic annotation of Web Resources. However, this approach does not rely on WADL, but follows the basic ideas of SAWSDL. REST-based services are described in this approach by (X)HTML. A specific (X)HTML service description can be added to different metadata models, such as taxonomies or ontologies in order to describe a service semantically.

In this approach, we use WADL as service description language for the grounding of the connected RESTful sources. This is done according to the work described in [13], whereas the concrete service description for the integrated sources of knowledge of the addressed domain must be extended accordingly. See a detailed description in Section III-D *Semantic Service Repository*.

C. Ontologies and Semantic Web Technologies

A prominent and leading role in the development of Semantic Web technologies plays the W3C with the technical evaluation and specifications of its recommendations, the so-called standards. In line with the extension of the current Web to the "Semantic Web", in which information is given well-defined meaning, to enable a better work and cooperation of people and computers [14], one of the most important data formats, the Resource Description Framework (RDF) [15], has been developed. RDF formalizes the syntax for the description of statements in the specified order of subject, predicate and object. Such triples define resources, which use a prefixed uniform resource identifier (URI) to unique characterize the type. This allows a combination of data and resources from various sources. Unfortunately, it is very difficult to model complex interrelationships in these triples. For the modeling of complex contexts and the representation of concrete domain knowledge, ontologies are the appropriate method of choice.

An ontology is often defined as a *formal, explicit specification of a shared conceptualization, with the intended purpose of enabling knowledge sharing and reuse*, whereas the conceptualization is an abstract simplified view of the world that we are trying to represent [16].

In knowledge organization the term "domain" refers to the scope for which facts of knowledge are formalized. Ontology defines a formal system, which represents knowledge by means of a fixed relevant descriptive terminology or vocabulary, relations, hierarchies and logical attributes. With this logical representation and the taxonomic structure, and the specific evaluation of these structures and interrelations, knowledge can be derived by inference mechanisms. The W3C recommends the Web Ontology Language (OWL) [17], a standardized representation language. It is currently the most advanced ontological approach, which also finds global popularity and use. The design of OWL also relies on RDF. Because of its complexity OWL currently exists in three expansion stages or versions: OWL-Lite, OWL-DL and OWL-Full. The so-called lightweight or "lite" version offers the simplest variant on the use of OWL ontologies, whereby the user of OWL-Full can exploit the full expressive power [18].

Ontologies are modeled for a specific application domain and in this approach the created ontologies define terminologies for relevant context-adaptive properties and related entities. In detail, the ontology contains a variety of customizable parameters, which can be adapted according to a specific domain, described in Section III-A *Representation structures*.

D. Semantic Web Services

The Web service ontology language OWL-S [19] is based on OWL and extends this base to a set of constructs that relate to properties, specificities and dependencies of the Web service level and is also machine readable and processable. A concrete service description in OWL-S is divided into three parts: service profile, service model and service grounding. Primarily the service profile is used for service discovery and describes what the service does. It contains information about the organization that provides the service, the preconditions, input and output values, preconditions and effects, as well as the features and benefits of the service. Once a service has been selected for use, the service profile is no longer used. For the concrete process of involving and execution of a service the description defined in the service model is used. Figure 1

 ISBN: 978-1-61208-353-7

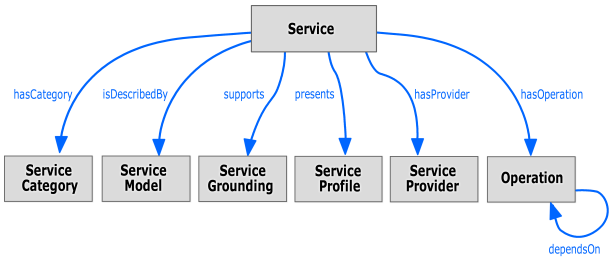

Figure 1. Main concepts of the Web service ontology language OWL-S.

shows the main concepts and relations of a service description in OWL-S.

The *Service Model* describes the actual execution process of a service. Here, this process description consists of simple atomic processes or complex composite processes that are sometimes abstract and not executable. The process describes the individual use of the service by clients by specifying input and output data, preconditions and effects.

The *Service Grounding* provides detailed technical communication information on protocols and formats as well as addressing details. Furthermore the grounding model provides a direct link or mapping between the service model and the technical service execution level. For example implementation details like input and output messages of the service model are translated into corresponding elements of the service description language. The W3C recommends and specifically describes in its member submissions WSDL, but other groundings are also possible. For a better understanding of this recommendation, it is important to know that W3C member submissions serve as input to the standards process. These descriptions contain concrete information for the service implementation and realization by enabling a direct link between the grounding and the WSDL elements. In their research articles Sirin et al. describe their prototypical implementation to directly combine OWL-S with actual executable invocations of WSDL [20][21].

This approach is based on the preliminary work in the area of semantic Web services modeling with grounding in WSDL and expands the approach to lightweight REST-based interfaces with their service descriptions in WADL [13][22].

III. Concept of the Framework

With this framework, users easily get access to Semantic Web Services without the need of special skills of RDF(S) or specific database query languages like SPARQL Protocol and RDF Query Language (SPARQL) [23]. For non-specialists the entry barrier in the world of Semantic Web technologies, to handle RDF triples, or to formulate a SPARQL query, is very high. These query languages are primarily designed to exploit the full power of the Semantic Web and make it possible to navigate in big semantic annotated data bases and specifically define restrictions to extract the specific information. As mentioned at the beginning of the article this framework uses a combination of Semantic Web technologies to create fine-grained matching answers to a given combined user query.

The composition of heterogeneous Web services and Semantic Web Services - which are playing the role of knowledge sources - poses a special challenge, because Web service interfaces or APIs might change over time. The functions are defined according to the principle of input-processing and output (IPO), as depicted in Figure 2. The interaction system (multimodal dialog system with its tablet or smartphone clients) formulates a query and the service framework generates a corresponding output after involving adequate services.

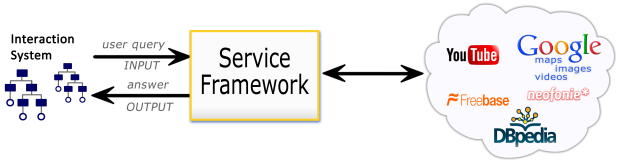

Figure 2. Process workflow of the service framework.

The focus of this paper is to provide an overview of the concept of the developed framework, to describe the discovery of services and the corresponding retrieval of semantic content. The processing chain contains many open questions in the field of:

- Query analysis and evaluation with associated mapping and matching of data structures from one format to the next. (*Query Processing*)
- Service discovery according to a complex combined query structure and the matchmaking process. (*Semantic Service Repository*)
- Composition of services to interoperable complex services. (*Planning and Execution*)
- The output presentation for each client application and the detection of duplicates. (*Output Presentation*)

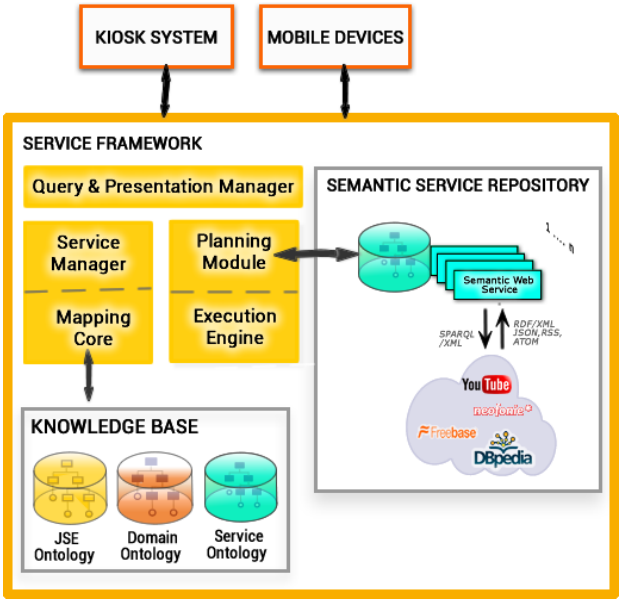

Figure 3. Architecture of the *Service Framework*.

A specific coarse-grained architecture design of the service framework is shown in Figure 3. The specific core components the (*Query Module*), the (*Planning Module and Execution Engine*) in combination with the (*Semantic Service Repository*), and finally the (*Output Presentation Manager*) form the

 ISBN: 978-1-61208-353-7

processing chain to retrieve factual knowledge out of connected knowledge sources based on semantic technologies.

A. Representation structures

The key feature of this component is the fusion of information, but with the ulterior motive or intention to integrate information and keep information interpretable and assessable. Therefore, the component harmonizes the contents on the basis of an ontology that reflects the vocabulary and domain knowledge. This solution is used in distributed application architectures as backend query point, which is the direct link in a heterogeneous world of services. The necessary knowledge contents (images, facts and videos) are introduced into the system in matching input and output formats.

The used ontologies represent and define all found metadata semantically:

- Discourse Ontology (eTFS)[24]
- Service-Framework Ontology (OWL)
- Service Ontology (OWL-S)

The *Discourse Ontology* (for modeling details see Section III-B *Query processing*) defines the vocabulary of the connected client system and characterizes the request and expected response structures. These structures must be analyzed and aligned to concepts with the similar meaning out of the *Service-Framework Ontology*. These alignments are done by using mapping data structures, defined in the *Mapping Core*. The *Service-Framework Ontology* is modeled in OWL [18]. The *Service Ontology* based on OWL-S [19] is used to describe the services in the *Semantic Service Repository* semantically.

B. Query Processing

The processing chain starts with the request of a client application to the service framework. The interface of the *Query module* distinguishes between different input formats. As mentioned at the beginning of this paper, complex natural language formulations require complex modeling languages to express and represent all dependencies and other linguistic nuances. The developed multimodal dialogue system [25] uses extended Typed Feature Structures (eTFS) [24] to formulate *complex natural language queries*. These structures comprise a hierarchy of types and typed semantic objects, which are useful to formulate semantic queries, such as *"Give me the bordering countries of Germany and the population and some images of their capitals"*. In a special case, concrete instances of pre-defined concepts of the discourse ontology of the dialog system can be adapted to formulate a specific query. However, the query is intentionally abstract and does not specify which services in particular should be used and how the result should be obtained. Figure 4 shows how to define such a composed query with focus on bordering countries and main capitals. The abstract eTFS query specifies two attributes ("slot" types):

- `CONTENT`: This slot contains a semantic object and represents input data.
- `FOCUS`: This slot points to the part what is requested and may appear more than once to address and define the abstract search options.

```
<object type="http://dfki.de/dm#AttributeRetrievalTask">
 <slot name="http://dfki.de/dm#hasTrigger">
  <object type="http://dfki.de/ont/odp#InfoRequest">
   <slot name="http://dfki.de/ont/odp#hasContent">
    <object type="http://dfki.de/dm#Country">
     <slot name="http://dfki.de/dm#countryCode">
      <value type="String"><![CDATA[DE]]></value>
     </slot>
    </object>
   </slot>
   <slot name="http://dfki.de/ont/odp#hasFocus">
    <object type="http://dfki.de/ont/odp#Focus">
     <slot name="http://dfki.de/ont/odp#hasContent">
      <object type="http://dfki.de/dm#Country">
       <slot name="http://dfki.de/dm#hasBorderingPlace"/>
      </object>
     </slot>
    </object>
   </slot>
   <slot name="http://dfki.de/ont/odp#hasFocus">
    <object type="http://dfki.de/ont/odp#Focus">
     <slot name="http://dfki.de/ont/odp#hasContent">
      <object type="http://dfki.de/dm#Capital" />
     </slot>
    </object>
   </slot>
  </object>
 </slot>
</object>
```

Figure 4. Complex combined query formulated in eTFS-Format including search topics and properties as input parameters.

In parallel to complex formulated queries, in formats like eTFS, a second simple format, where queries are assembled in XML [26], can also be used. The interpretation and analysis of simple XML queries to align the requested content with the framework's ontology is carried out in accordance to an underlying schema and taxonomy. There exist predefined mapping rules that bundle the concepts or objects of decomposed query parts of the input structures to domain concepts. For this procedure element-based matching techniques are used, according to each decomposed query part, concepts or objects are mapped to a local meta-representation, the internal framework's ontology. The query component extracts knowledge concepts and adds them to predefined ontological structures. The outcome of this are individuals, basic ontological components, and relations to internal concepts that represent the query in a processable, framework-readable form. It specifies the input type, such as a city or a country, specified by properties (complete name, keywords, etc.), implicit relations and search topics (area, population, etc.). A detailed description on query processing and decomposition of this framework in combination with a multimodal dialog system is discussed in [2].

C. Planning and Execution

Due to the high amount of services, planning has become a major task. Moreover, the goal of the system is also to carry out the planning and composition of these services automatically. Pre- and post-conditions must clearly characterize a problem in the planning process, to allow a compositional process. Due to the interoperability of services, it is easier to transfer functionality, without having to integrate the services completely. The task is to find Web services and to either decide to compose them or call them directly. Here, the composition workflow can be quite complex, under consideration and evaluation of input and output parameters. In most cases the

 ISBN: 978-1-61208-353-7

solution to a complex query can be solved by a combination of services or the sequential invocation of multiple services. This consideration in turn raises the issue of how services are interconnected and to find out if a static execution sequence is needed. An execution sequence defines the composition or concatenation of services, whereby a service at its execution time ($S1_{(t)}$) might need for its execution the result structures of another service ($S2_{(t-1)}$). This management of the supply chain opens up new challenges - services need to work together to compose complex services [27].

This paper does not set the focus on planning and composition: to discuss the algorithm in detail is beyond the scope of this paper. It is intended to give an overview of the developed framework and the discovery of services. The here presented framework uses Business Process Model and Notation (BPMN) a high-level notation that specifies semantics for composition, to orchestrate services and provide aggregated results [28]. The advantage of this approach is to use the potential of a combination of different services to focus information and knowledge. The planning component works with predefined plans, that define preconditions that must be matched by adequate modeled input query structures. A plan contains the description on how to proceed in the discovery and execution process, which abstract types of services are needed, the domain that is addressed, in which order the services have to be executed, and all essential parameters for the matchmaking process in the connected *Semantic Service Repository*, that opens up access to services that encapsulate knowledge sources of different domains.

The *Execution Engine* provides connectors and encapsulates the calls to the REST or API interfaces, by reformulating and using specific query formats like XML or languages like SPARQL and Metaweb Query Language (MQL) [3]. Once all results of different called services are received by the *Execution Engine* an internal mapping process starts a review and reasoning process and maps them with the help of additional semantic mapping rules and classifies them according to the internal framework's domain ontology.

D. Semantic Service Repository

The discovery and matchmaking of services is a major part in this processing chain. The *Semantic Service Repository* provides and manages semantic descriptions of various pre-annotated information sources. One of the advantages of this component is the functionality to add, modify, replace, and delete these stored descriptions of sources without hard programmatic dependencies and without stringent dependencies on specific providers of information and their interfaces (APIs). Certainly the component benefits from the interoperability of services. The *Semantic Service Repository* provides access to different types of information sources like Semantic Web Services that cover information stored in external database management systems or *Semantic Repositories*, such as DBpedia [4] and Freebase. These systems store information in a structured and manageable form, but can be only accessed by special query languages like SPARQL for DBpedia or MQL for Freebase. The main difference of this approach compared to conventional database management systems is the usage of ontologies as a technology to harmonize and store semantically structured data - concepts define and classify information and contain implicit knowledge characterized by name and position in the hierarchy. In this approach, all Web services are represented in OWL-S with a grounding declaration in WSDL or WADL [21][22][29].

Figure 5 shows the internal discovery process of the *Semantic Services Repository*. An XML query is sent by the *Planning and Execution* component to the *Broker* module. On the basis of a single XML schema, the *Query Handler* interprets the input, that characterizes abstract types of services, and maps them into an internal interpretable data collection, that groups multiple input elements into a single unit. Based on this input data collection a rule engine evaluates the input. A consideration of the *Rule Engine* relies on a set of predefined rules that are stored in a local rule base. When the *Broker* module is involved, the rule base is filled with these predefined rules and a working memory is generated and selected application data and objects are provided. The analysis of the planners' request is done exactly by these rules. Therefore, the rules must express in their condition part the terminology of the problem domain. When a rule matches within the current context the rule's consequence part refers to a particular SPARQL query. This means that the defined conditions are attuned and adapted to filter options of the SPARQL query in the consequence part of the rule. In line with [30], SPARQL is used to define filters that are precisely tailored to the *Service Ontology*. With these filter options the search for matching services can be arbitrarily fine-grained and started very restrictive. If no results are available, the parameters and concept types can be adjusted according to the hierarchy of the *Service Ontology*. This query is forwarded to the *Matcher* module, that connects the service repository and executes the received SPARQL query, which relates to adequate and concrete Semantic Web Service representations of deposited services. When the query matches, a list of semantic services descriptions is returned, which describe ontologically, how to interpret and execute the service. All parameters, that are required for service execution, are taken from the ontological description and added to the assembled concrete request structure for the external service call. The resulting data structure is returned to the *Planning and Execution* component that triggers the process of integration, analysis and harmonization.

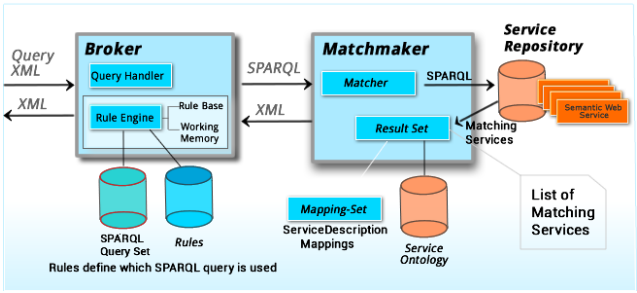

Figure 5. Service discovery process.

In this approach, the grounding description with technical concepts of WADL and WSDL, that describe the communication of external services, has been extended by parameters or properties that refer to corresponding pattern generated SPARQL queries, which are used to call sources like DBpedia or other triple stores. Furthermore, each SPARQL query defines the parameters that are used in the output structures, and which must be mapped to the internal ontology. Therefore, another reference is attached to the grounding description,

 ISBN: 978-1-61208-353-7

which describes how the results of this SPARQL call must be mapped to the internal framework's ontology.

E. Mapping and matching

In the processing chain of the framework content of heterogeneous data channels must be repeatedly aligned: this is called mapping. In this paper, the concepts of mapping and matching are used interchangeably. A distinction of these concepts is just possible in a concrete application scenario, where element transformations occur, but not out of this abstract view. The key aspect of mappings is to combine one representation structure with another. Whereas the major challenge is to create comparable and interoperable mapping functions between the used terminologies, the mappings of the result structures of the stored service descriptions have been defined in a pre-processing phase in a formal description language. For the unambiguous assignment of the models and types of an element the mapping functions are precisely specified by categories.

The *Mapping Core* realizes these mappings at each part of the framework and forms the central interface between the representation structures of each knowledge domain and can be replaced accordingly. Figure 6 illustrates the internal communication and interaction workflow of the individual core components in detail. An incoming request is evaluated by the *Query Module*, as described in Section III-B, and is mapped to the internal domain ontology that is used for further processing in the planning process. Additionally, in the *Planning Module* and the *Execution Engine* the service framework involves knowledge sources that are encapsulated or wrapped by Web service interfaces.

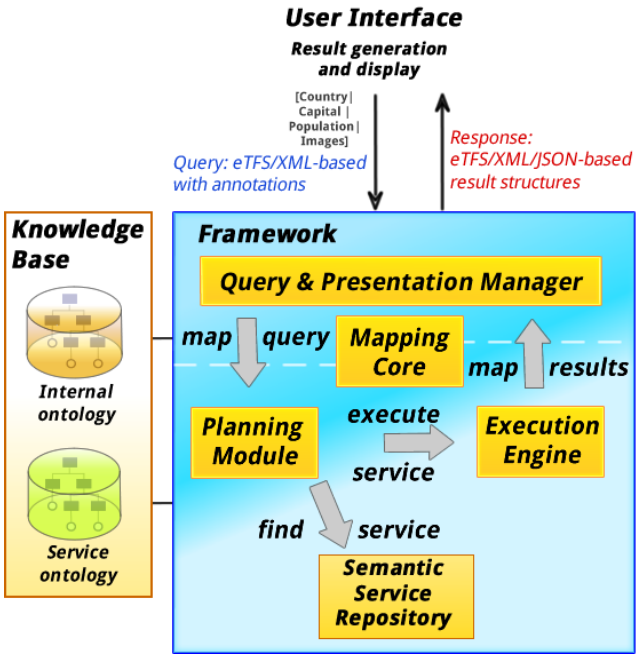

Figure 6. Internal communication workflow - query processing and data type mapping.

As an example, Figure 7 shows the procedure of mapping result structures of a remote relational data base by formal declaration of mappings to the internal framework's ontology. A mapping of the results of the called external sources, such as complex external semantic data structures or simple XML structures to the internal framework's ontology must be fulfilled and with the help of these mapping and transformation rules the process of data harmonization can be controlled.

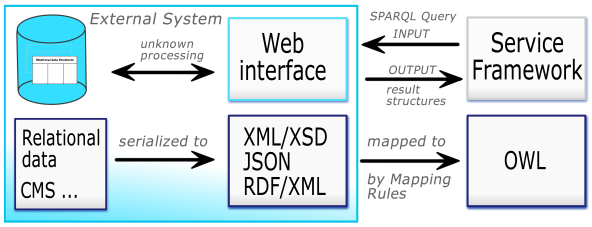

Figure 7. Dataflow of mapping external result structures.

In the case of mapping complex external semantic data structures, formal mapping rules are used that allow a higher quality data type mapping on a more abstract level: new individuals are created and linked to each other or a taxonomy of objects must be mapped to the internal data structure. For the mapping of simple XML result structures, an adaption of Extensible Stylesheet Language Transformations (XSLT) [31] can be used to create a meta XML format that is mapped to individuals of the framework's ontology. Figure 8 shows the mapping of an external data structure that contains the geo-coordinates of a city. The parameters `longitude` and `latitude` are mapped to a new individual of type `GeoLocation`. This new individual is bound by the reference `hasGeoLocation` to the existing individual with the concept type `CityTownVillage`.

The translation of data models of different domains with their vocabulary and definitions is the first step towards semantic harmonization of incoming results of connected services and the most important step in creating comparable content. These translations based on the extracted semantic information are necessary to create correct translations with the aim of identification of equality, similarity and heterogeneity. But particular problems (misspelling and typing errors) for the harmonization emerge, which need to be fixed separately by domain specialists. The analysis searches all generated individuals in the framework's ontology to find structures with the same formal and semantic criteria. Duplicates could just be filtered at the instance level, because the information with their semantics is here clearly and comparable. Found images results

```
CONDITION[
 INSTANCE[jse:CityTownVillage]
  REFERENCE[
    INSTANCE[jse:GeoLocation]
    INSTANCE_XPATH[//geo:results/geo:result]
    SET_RELATION(jse:CityTownVillage#hasGeoLocation
                    ->jse:GeoLocation)
      if(XPATH[//geo:binding/@name]==longitude) {
        XPATH[//geo:binding/geo:literal]->longitude
      }
      if(XPATH[//geo:binding/@name]==latitude) {
        XPATH[//geo:binding/geo:literal]->latitude
      }
]]]
```

Figure 8. Mapping of the geo-coordinates of a city into the framework's ontology.

 ISBN: 978-1-61208-353-7

can only be compared on the basis of textual comparison of title, captions, and descriptions.

F. Output Presentation

The creation of the output structure is the last step in the processing chain. This is done by the *Presentation Manager* which coordinates, transforms and merges the created semantic individuals in a standardized result structure. Depending on the user's query and the type of client application (smartphone, tablet or desktop) the output format is specified (RDF, XML, JSON, eTFS, etc.). The component uses the *Mapping Core* for the filtering and extraction of individuals out of the framework's ontology and the declarative element-based mapping of concepts and properties to data collections of the resulting structures. These prepared contents are delivered to the appropriate connected client applications.

The component works statelessly in terms of domain-specific parameters, the client application defines what processing has to look like and sets the processing context. After delivery of the result structure oblivion continues and the local retrieved data structures are deleted, to avoid a mixture of contents and interpretation errors when the next query is received.

IV. Scenarios

For many people mobile devices, such as tablets or smartphones, have become indispensable in today's modern world. These devices offer the possibility of full mobility, anywhere, at any time information is provided and people make use of these offers. Accordingly the behavior of the users changed and reinforces the focus on Web services, the infrastructure components of the Internet. Companies have recognized the increasing demand of that market where Web services provide dynamic factual knowledge on request, and make commercially use by bundling functionality of their Web services in cloud solutions [32].

With the presented service framework a service-oriented information kiosk system for public places, like museums or hotel lobbies, has been developed, to force and support multi-user collaboration [33]; see Figure 9. Users can connect their smartphones by app (Android, iOS) to a large terminal and share interesting facts, pictures and videos via gesture interaction.

The advantage of this multimedia platform is the seamless combination of a touch-based kiosk system and the access of heterogeneous information sources via the described service framework. It is also possible to interact and control semantic interaction elements (SIE) [34] the kiosks' applications by speech. The in-built smartphone's microphone is used to get the user's speech input (utterance). Figure 10 shows the prototypical implementation of the developed service framework in combination with multi-modal mobile access to external information by a dialog system [24].

Furthermore, a system was developed to interrogate geological facts, such as finding rivers, their length and right or left tributaries, also in combination with a multimodal dialog system [24]. The complete integration of factual knowledge out of standardized *Web Feature Services* [35] that are involved to access geological data is achieved by the service framework, using an adapted domain ontology.

Figure 9. The service-oriented Calisto kiosk system.

Figure 10. Interaction with the Calisto system prototype.

V. Conclusion and Future Work

This paper presented an approach that describes the flexible integration of heterogeneous knowledge sources and their parallel execution and analysis as well as the harmonization of the results by means of an internal framework's ontology. The solution addresses heterogeneous data sources and integrates information in spite of the different query languages and different enriched domain ontologies. Therefore, for better understanding an overview of the configuration and structure of the core components of the developed frameworks was given. Furthermore, the alignment and matching of data structures along with a detailed description of the service discovery process in semantic service repositories were presented. Moreover, it was shown how implemented prototypes of the discussed framework were used in different scenarios by using factual knowledge of their respective domains like geography or touristic information. In this context the framework acts as an important factor to connect and integrate structured information.

Future work will address the point of switching the domains. In order to make it easier for the domain experts a workbench is needed to generate the mapping structures with graphical support, because manual mapping is time consuming and error prone.

Another key aspect is the optimization of result structures. Better filtering mechanisms are tested in order to detect duplicates or merge similar content. Furthermore, depending on

 ISBN: 978-1-61208-353-7

the domain a visual pattern recognition can be used to eliminate duplicate images.

REFERENCES

[1] J. Bih, "Service Oriented Architecture (SOA) a New Paradigm to Implement Dynamic e-Business Solutions," Ubiquity, no. August, 2006, pp. 4:1–4:1.

[2] S. Bergweiler, "Interactive service composition and query," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, pp. 169–184.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in Proceedings of the 2008 ACM SIGMOD international conference on Management of data. New York, NY, USA: ACM, 2008, pp. 1247–1250.

[4] S. Auer et al., "DBpedia: A Nucleus for a Web of Open Data," in The Semantic Web, ser. Lecture Notes in Computer Science, K. Aberer et al., Eds. Springer Berlin Heidelberg, 2007, vol. 4825, pp. 722–735.

[5] "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)," W3C Recommendation 27 April 2007, April 2007, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/soap12-part1/

[6] L. Richardson and S. Ruby, RESTful Web Services. O'Reilly Media, 2007.

[7] M. Hadley, "Web Application Description Language (WADL): W3C Member Submission," August 2009, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/Submission/wadl/

[8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C, W3C Note, March 2001, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/wsdl

[9] M. Gudgin et al., "Soap version 1.2 part 1: Messaging framework (second edition)," W3C Recommendation REC-soap12-part1-20070427, April 2007.

[10] R. T. Fielding et al., "Rfc 2616, hypertext transfer protocol – http/1.1," 1999, [retrieved: July 2014]. [Online]. Available: http://www.rfc.net/rfc2616.html

[11] R. T. Fielding, "Architectural styles and the design of network-based software architectures," PhD Thesis, University of California, Irvine, 2000.

[12] K. Gomadam, A. Ranabahu, and A. Sheth, "SA-REST: Semantic Annotation of Web Resources: W3C Member Submission," April 2010, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/Submission/SA-REST/

[13] O. F. F. Filho and M. A. G. V. Ferreira, "Semantic Web Services: A RESTful Approach," in IADIS International Conference WWW/Internet 2009. IADIS, 2009, pp. 169–180, [retrieved: July 2014]. [Online]. Available: http://fullsemanticweb.com/paper/ICWI.pdf

[14] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001, [retrieved: July 2014]. [Online]. Available: http://www.jeckle.de/files/tblSW.pdf

[15] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," W3C Recommendation, 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[16] T. R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, vol. 5, 1993, pp. 199–220.

[17] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C, W3C Recommendation, 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/owl-features

[18] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax, W3C W3C Recommendation, Feb. 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/

[19] D. Martin et al., "OWL-S: Semantic Markup for Web Services," 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

[20] E. Sirin, B. Parsia, and J. Hendler, "Filtering and selecting semantic web services with interactive composition techniques," IEEE Intelligent Systems, vol. 19, no. 4, 2004, pp. 42–49.

[21] E. Sirin, "Combining description logic reasoning with ai planning for composition of web services," dissertation, University of Maryland, College Park, 2006.

[22] D. Lambert and J. Domingue, "Grounding semantic web services with rules," in Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP), ser. CEUR Workshop Proceedings, A. Gangemi, J. Keizer, V. Presutti, and H. Stoermer, Eds., vol. 426. CEUR-WS.org, 2008, [retrieved: July 2014]. [Online]. Available: http://ceur-ws.org/Vol-426/swap2008_submission_8.pdf

[23] "SPARQL 1.1 Query Language - W3C Recommendation," W3C Recommendation, 2013, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[24] T. Becker et al., "A unified approach for semantic-based multimodal interaction," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, pp. 135–148.

[25] D. Porta, M. Deru, S. Bergweiler, G. Herzog, and P. Poller, "Building multimodal dialogue user interfaces in the context of the internet of services," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, pp. 149–168.

[26] K. G. Clark, "Extensible Markup Language (XML) 1.0 (Fifth Edition): W3C Recommendation," W3C Recommendation, November 2008, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/xml/

[27] M. Ghallab, D. S. Nau, and P. Traverso, Automated Planning - Theory and Practice. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[28] P. Y. Wong, "Compositional development of bpmn," in Software Composition, ser. Lecture Notes in Computer Science, W. Binder, E. Bodden, and W. Löwe, Eds. Springer Berlin Heidelberg, 2013, vol. 8088, pp. 97–112.

[29] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities," in The Semantic Web - ISWC 2002: First International Semantic Web Conference Sardinia, Italy, 2002, pp. 333–347.

[30] J. M. Garcia, D. Ruiz, and A. Ruiz-Cortes, "Improving Semantic Web Services Discovery Using SPARQL-Based Repository Filtering," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 17, 2012, pp. 1–15.

[31] "Xsl transformations (xslt) version 1.0," W3C Recommendation, 11 1999, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/xslt

[32] T. Barton, "Cloud computing," in E-Business mit Cloud Computing. Springer Fachmedien Wiesbaden, 2014, pp. 41–52.

[33] S. Bergweiler, M. Deru, and D. Porta, "Integrating a multitouch kiosk system with mobile devices and multimodal interaction," in ACM International Conference on Interactive Tabletops and Surfaces, ser. ITS '10, ACM. New York, NY, USA: ACM, 2010, pp. 245–246.

[34] D. Sonntag, M. Deru, and S. Bergweiler, "Design and implementation of combined mobile and touchscreen-based multimodal web 3.0 interfaces," in Proceedings of the International Conference on Artificial Intelligence, ser. ICAI-09, July 2009, pp. 974–979.

[35] "Web feature service implementation specification," OGC Document 04-094, Version 1.1.0, Standard, May 2005, [retrieved: July 2014]. [Online]. Available: http://portal.opengeospatial.org/files/?artifact_id=8339

 ISBN: 978-1-61208-353-7