Towards the Formalization of Interaction Semantics

Irene Celino
CEFRIEL - ICT Institute
Politecnico di Milano
Via Fucini, 2
20133 Milano, Italy
irene.celino@cefriel.it

Francesco Corcoglioniti
Dip. di Elettronica e Informazione,
Politecnico di Milano
Via Ponzio, 34/5
20133 Milano, Italy
corcoglioniti@elet.polimi.it

ABSTRACT

With the advent of Web 2.0 and the emergence of improved technologies to enhance UI, the importance of user experience and intuitiveness of Web interfaces led to the growth and success of Interaction Design. Web designers often turn to pre-defined and well-founded design patterns and user interaction paradigms to build novel and more effective Web interfaces. The rational behind Interaction Design patterns is based on user behavior and Web navigation studies. The "semantics" of user interaction is therefore a rich and interesting area that is worth exploring in association with traditional Semantic Web approaches.

In this paper, we present our first attempts of an ontological formalization of interaction patterns and its implications. To prove our concept, we illustrate the mapping approach we employed to put in relation that interaction formalization with data-specific ontologies, to create Web interfaces to browse and navigate that specialized kind of information; the aforementioned ontologies and mapping rules are the basis of the internal operation of a Semantic Web application framework called STAR:chart, leveraged to build the Service-Finder portal; finally, we present our evaluation results.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Web-based interaction*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/ Hypermedia; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Design, Semantics, Human Factors

Keywords

Interaction Semantics, Semantic Web, Web Interfaces

I-SEMANTICS 2010 September 1-3, 2010 Graz, Austria

1. INTRODUCTION

One of the key success factors of the so-called Web 2.0 is definitely the attention given to the user experience. Usability and intuitiveness of Web interfaces became central in any Web site design and the rise of Interaction Design techniques applied to the Web are a demonstration of this fact. AJAX technologies themselves are so heavily employed not only for "aesthetic" reasons, but also to make the final user navigation easier and more natural, in a way more similar to the interaction with desktop applications.

The Semantic Web community also turned to the Interaction Design world, by organizing several dedicated workshops in the *Semantic Web User Interaction* series[1], with the objective of exploring "the issues and challenges faced by users accessing and interacting with the Semantic Web", and in the *Visual Interfaces to the Social and Semantic Web* series[2]. Therefore, the large majority of the works submitted to those workshop series is aimed to investigate the most suitable user interface to browse the Semantic Web (or, more recently, the Linked Data cloud).

However, the opposite point of view should be also taken into account: how the Semantic Web technologies can be successfully employed to support the (Web) Interaction Design community? What is the "semantics" of user interaction? How ontologies and rules can help Web designers in completing their tasks? This is the main objective of our investigation and in this paper we will present the first results of our research in this challenging yet almost unexplored field.

The remainder of the paper is organized as follows: in Section 2 we motivate our research and we explain our claim; Section 3 explains our formalizations of interaction semantics in terms of our ontologies and our mapping approach; a successful implementation of our concept in the Service-Finder portal is described and evaluated in Section 4. In Section 5 we discuss the future works to improve our conceptualization and the possible research lines we foresee in the field of Semantic Web and Interaction Design; some concluding remarks are offered in Section 6.

2. MOTIVATION AND APPROACH

As apparent from the last few years of Web evolution, more and more frequently the principle "Data is the new Intel Inside" [21] is becoming the foundation and the driving principle of Web application development. We live in a

[1]Cf. `http://swui.semanticweb.org/`.
[2]Cf. `http://www.smart-ui.org/events/vissw2010/`.

knowledge-intensive world where the meaning of information and the various facets of data are core to all our applications.

The need for presenting knowledge-intensive data in Web user interfaces is therefore a common need; often, the Web-based interfaces must be customized to visually represent those data in the best possible way, by preserving and underlining the data meaning. Finally, to foster users' adoption, the Web interface must also be improved to assure its usability and to ease user interaction.

Our concept tries to tackle those challenges by proposing the adoption of a *mapping* approach between the *data semantics* and the *interaction semantics* to support the development of Web interfaces. Data semantics refers to the knowledge of the specific domain represented in the data, while interaction semantics regards the concepts, patterns, methods and best practices of the Interaction Design field. Our idea is that Web interaction patterns have their own "semantics", which can be formalized and employed in a mapping at conceptual level with data-specific ontologies. Thus the need for an ontological formalization of interaction semantics.

Our approach can be summarized as follows: firstly, we analyze and conceptualize the interaction "semantics" in terms of its common patterns and their combination in the design of Web architecture and navigation; then, we propose a mapping approach that puts in relation the meaning of data to be presented with the interaction conceptualization.

This approach is in line with the methodologies and tools developed by the Semantic Web community in the first years of 2000, usually referred to as "Semantic Web Portals" [18]; our research as well started in 2001 when we firstly explored the application of Semantic technologies to Web portals, as highlighted by our contributions [14, 11] to the Semantic Web Portal research line.

The novelty of the approach presented in this paper, however, lies in the explicit introduction of the experience of the Web Interaction Design community. In particular, we make a distinction between the part of interaction that depends on the very nature of the data (the natural way to interact and perform operations on the data) and the rendering of the interaction in the actual user interface (how the interaction patterns are implemented in Web pages).

While the previous attempts were targeted to directly map the data to its visual model (like, for example, in the employment of the well-known Fresnel RDF presentation vocabulary [22] or in the more recent LESS approach [2]), we introduce a level of abstraction between the data and the details of their visual presentation. In the next section, we introduce the main primitives of our interaction conceptual model and we explain how those abstractions can be exploited to fulfill our objective.

3. TOWARDS AN INTERACTION ONTOLOGIZATION

To better clarify our approach towards the formalization of interaction semantics, in Figure 1 we visually represent the various steps and ontological artifacts employed. The two "users" whose points of view are to be taken into consideration are the *data manager*, i.e. the person holding the knowledge about the data semantics, formalized in the domain ontology, and the *Web designer*, i.e. the professional Web expert in interaction semantics.

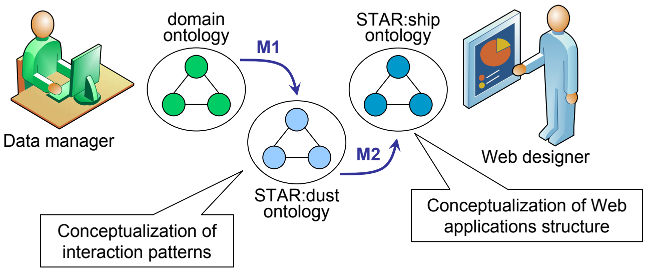

Figure 1: The three interplaying conceptual models.

In order to bridge between the two different points of view, we introduce two ontology artifacts: a low-level conceptualization of Web application structures – the *STAR:ship* ontology, formalizing the knowledge in Web development – and a more high-level conceptualization of interaction patterns – the *STAR:dust* ontology. The domain data operated by the data manager are then "transformed" into a representation closer to the Web designer point of view by means of two *mapping steps* (M1 and M2 in Figure 1), which reflect the correspondences between the conceptual models.

The rest of the section is devoted to the explanation of the two conceptual formalizations and of the mapping steps.

3.1 STAR:ship ontology

In the Web design community, several efforts are aimed to model the most frequent interaction patterns. A good example is the popular Yahoo! Design Pattern Library[3], which lists more than 50 patterns divided in categories (layout, navigation, selection, rich interaction and social); those patterns span from the field auto-completion to the navigation breadcrumbs, from the progress bar to the tool-tip messages. The level of granularity of each of those interaction patterns is quite fine-grained: each pattern is very detailed, especially in terms of its visual appearance and its usability/accessibility characteristics.

A part of our formalization of interaction semantics is therefore devoted to the conceptualization of the Web structural and visual elements that constitute the elementary building blocks of a Web page organization. We named this formalization *STAR:ship ontology* (represented on the right side of Figure 1), the specification of the primitives to model the structure of a Web application, i.e. its sitemap in terms of areas, pages, "units" or "widgets" and links. This formalization is not completely a new idea, since it builds on the long-term research and know-how in Web engineering (like in [12]); it was also explored by some preliminary research work in the Semantic Web community [17].

In Listing 1, a small extract of the STAR:ship ontology expressed in N3 notation [4] is offered; the "Widget" and "Event" classes are defined, representing the basic building blocks of Web pages (the widgets) and the events generated by users' interaction; also, a sample widget is defined, to describe a common search form element, which is implemented by a specific piece of software (a Java class, in the example) and triggers the actual searching event.

However, this kind of conceptualization is still too far from the specification of the data meaning, expressed by means of domain-specific ontologies (on the left side of Fig-

[3]Cf. `http://developer.yahoo.com/ypatterns/`.

ure 1). A direct mapping between a domain ontology and the STAR:ship conceptual model would require quite a deep understanding of both the details in the data and the minute traits of the visual appearance[4]. We therefore introduce a level of indirection by means of another interaction formalization at a different (more coarse-grained) level of granularity.

```
# STAR:ship namespace declaration
@prefix ss: <http://swa.cefriel.it/ontologies/
    starship#> .

# definition of the widget class
ss:Widget a owl:Class ;
    rdfs:subClassOf ss:Unit .

# definition of the event class
ss:Event a owl:Class .

# definition of the triggersEvent property
ss:triggersEvent a owl:AnnotationProperty .

# example of widget definition
ss:SearchFormWidget a owl:Class ;
    rdfs:subClassOf ss:Widget ;
    ss:hasJavaClass "it.cefriel.swa.star
          .widgets.SearchFormWidget"^^xsd:string ;
    ss:triggersEvent ss:submitSearchEvent .
```

Listing 1: Extract from the STAR:ship ontology.

3.2 STAR:dust ontology

Any Web user knows that there are several ways to interact with Web pages: from simple browsing by clicking on links to searching via search boxes; from understanding the information structure by scanning navigation menus to actively contributing by compiling forms or editing wiki pages, as effectively described in [13].

Therefore, the second and innovative part of our interaction semantics formalization is constituted by the *STAR:dust ontology*, the specification of the different roles the data should play in the Web application. In this second ontology, we define the *fruition modalities*, i.e. the possible interaction styles of the final users with the presented information (searching, browsing, detail viewing, tagging, editing, rating, etc.).

```
# STAR:dust namespace declaration
@prefix sd: <http://swa.cefriel.it/ontologies/
    stardust#> .

# definition of the PresentationProperty class
sd:PresentationProperty a rdfs:Class ;
    rdfs:subClassOf rdf:Property .

# definition of some presentation properties
sd:RelevantProperty a sd:PresentationProperty .
sd:SortableProperty a sd:PresentationProperty .
sd:EditableProperty a sd:PresentationProperty .
```

Listing 2: Extract from the STAR:dust ontology.

In Listing 2, a small extract of the STAR:dust ontology is offered; the "PresentationProperty" class is defined, representing the basic property to be visualized to describe a resource; some examples of presentation properties are listed, to indicate, respectively, an important property to be displayed, a property indicating some dimension on which a list of resources can be sorted, and a property of the resource that can be edited/added/modified by the user. Those presentation properties are then grouped into fruition modalities and used in the ontological mappings, as explained in next subsection.

It is evident that the STAR:dust ontology (as its "fine particles" name suggests) is at a higher abstraction level w.r.t. the STAR:ship ontology, because the two models represent different points of view on the interaction: while the conceptualization of the Web application structure (STAR:ship) reflects the Web designer mindset, the conceptualization of the fruition modalities (STAR:dust) expresses the best practices to access and interact with the data, thus representing a point of view much closer to the data expert's one.

3.3 Ontological mappings

In terms of the relations between the three models at hand (STAR:ship, STAR:dust and the domain ontology), in order to seamlessly pass from the data to their visualization in the Web application, we need two mapping steps (represented by the M1 and M2 arrows in Figure 1).

The mapping M2 between the two interaction ontologies we introduced is a step that can be automatized: each fruition modality defined in STAR:dust can be enforced through some standard combinations of the Web building blocks described in STAR:ship. For example, the *searching* fruition modality – the situation in which the final user search for items of interest in the Web application – can be realized by a proper composition of patterns: (a page with) a search box, (another page with) a list of results and the tools to action that list, like sorting and pagination features, and (a page for) a detailed view on each of the returned search results. A complete formalization of interaction semantics would therefore include some "default" mappings that put in connection the two abstraction levels of interaction.

On the other hand, the mapping M1 between the domain ontology and the STAR:dust ontology must be realized every time a Web user interface is built on top of a new kind of dataset. Differently from the approaches employed in the Semantic Web Portals research trend [18], however, this kind of mapping does not require dealing with fine-grained specification of data selectors and content formatting. On the contrary, this mapping can be directly operated by the data expert, without the need of a specific knowledge about Web design and development.

```
# definition of mapping in STAR:dust
sd:PresentationMapping a owl:Class .

# class and properties connected via a mapping
sd:onClass a owl:ObjectProperty ;
    rdfs:range  rdfs:Class .

sd:mappingSource a owl:ObjectProperty ;
    rdfs:domain sd:PresentationMapping ;
    rdfs:range  rdf:Property .

sd:mappingDestination a owl:ObjectProperty ;
    rdfs:domain sd:PresentationMapping ;
    rdfs:range  sd:PresentationProperty .
```

Listing 3: Definition of the mapping structure in the STAR:dust ontology.

[4]Actually, this expensive direct mapping is the adoption cost of Fresnel and LESS approaches cited in Section 2.

The mapping contains simple descriptions of the "roles" that different data play in the Web application and of the operations that are possible on those data. In Listing 3, the relvant portion of the STAR:dust ontology is offered, with the structural definition of a mapping. This mapping operates on a domain concept ("onClass"), putting in relation a domain property ("mappingSource") with a STAR:dust property ("mappingDestination").

For example, a datatype can be defined as "searchable", thus expressing the need for a search fruition modality for all data belonging to that type; in a similar way, a resource marked as "editable" calls for an editing fruition modality to allow users to modify its content.

3.4 Multiple views, multiple mappings

When building an application on top of a dataset, the need can arise to provide different "views" on such data, thus reflecting different points of view, different user requirements, different goals or functionalities.

It is worth noting that our approach is sound with respect to that need: it is indeed possible to draw multiple mappings between the different conceptualization, in order to address different situations. In other words, it is possible to have more than one correspondence between a data in the domain ontology and a fruition modality and between a fruition modality and a way to realize it in different Web applications.

On the one hand, different mappings can be defined between a domain ontology and the interaction conceptualization, in order to take into account different user privileges (users allowed to edit pages vs. users authorized to browse only), different user expertise or different seeking goals [13] (expert users who need all the data details vs. casual users looking for a general understanding), or even completely different applications (a search engine vs. a data analysis tool).

On the other hand, different mappings between our two interaction ontologies STAR:dust and STAR:ship can be introduced to derive different compositions of Web building blocks to construct Web pages. In the fruition modality exemplified before, the same searching behavior can be made concrete in different ways; for example, the modality can include/exclude a search "disambiguation" step between the submission of a search form and the display of a result list, or it can provide an "I'm feeling lucky" function that directly brings the user to a detailed view of the first search result. Another example would be the distinction in design between a desktop Web application and the same application for a mobile device; the same search pattern would require a different sub-division of the elementary blocks in pages to allow the typical interaction experience imposed by each device.

4. PROOF OF CONCEPT AND EVALUATION

In order to verify our concept and to test our approach, we implemented the basic ideas presented in the previous sections in a Semantic Web application framework we called STAR:chart. The architectural choices and the details of its realization are offered in [10]. We also employed this application framework to build some Web applications[5], among which the Service-Finder portal[6] represents the largest and most complete instantiation. In this section, we briefly describe how we designed and developed the Service-Finder portal and how we evaluated the results.

[5] Cf. `http://swa.cefriel.it/STAR`.

4.1 The Service-Finder portal

Service-Finder is a research project aimed at building a search engine for Web services [15]. The Service-Finder interface is the community portal that lets the users search and browse through Web services, understand the various services features and contribute to their descriptions in terms of tagging, rating, categorization and free text wiki-like descriptions.

In this scenario, the domain ontology is the *Service-Finder ontology*[7], which describes Web services with all their characteristics, service providers, portal users and their contributions. All the data described w.r.t. this ontology is stored in a knowledge base built on top of the OntoBroker Semantic Web middleware and inference engine [20], which is queried via a SPARQL-like query language.

The mapping between the Service-Finder ontology and the STAR:dust ontology was therefore created with the purpose of defining the role that the different entities (services, providers, users, etc.) play in the portal. For example, services can be searched (while providers cannot) and have three different types of search (free text, by category, by tag); some of the services characteristics are "core" and must be put in evidence on service pages, while other features are details; some of the services properties can be edited, others have a rating meaning, others can be used as a sorting criteria in search result lists; etc.

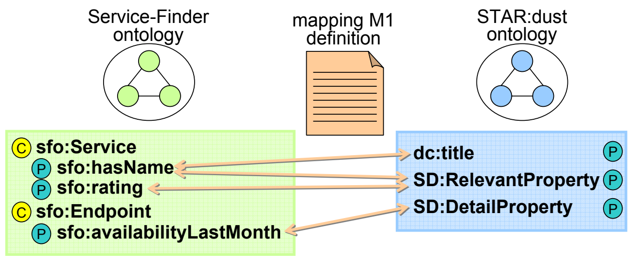

Figure 2: Example of mapping in Service-Finder.

In Figure 2, we visually represent an example of mapping between the two ontologies. The first two mapping arrows represented in the figure are expressed in details in the RDF mapping triples offered in the following Listing 4 (where `sd` is the prefix for the STAR:dust ontology, `sfo` for the Service-Finder ontology and `dc` for the Dublin Core vocabulary).

```
:sampleServiceMapping  a  sd:PresentationMapping ;
    sd:onClass             sfo:Service ;
    sd:mappingSource       sfo:hasName ;
    sd:mappingDestination  dc:title ;
    sd:mappingDestination  sd:RelevantProperty ;
    sd:isMultiValue        "false" ;
    sd:noValueText         " - " .
```

Listing 4: Example of mapping with STAR:dust.

[6] Cf. `http://demo.service-finder.eu`.
[7] Cf. `http://www.service-finder.eu/ontologies/ServiceOntology`.

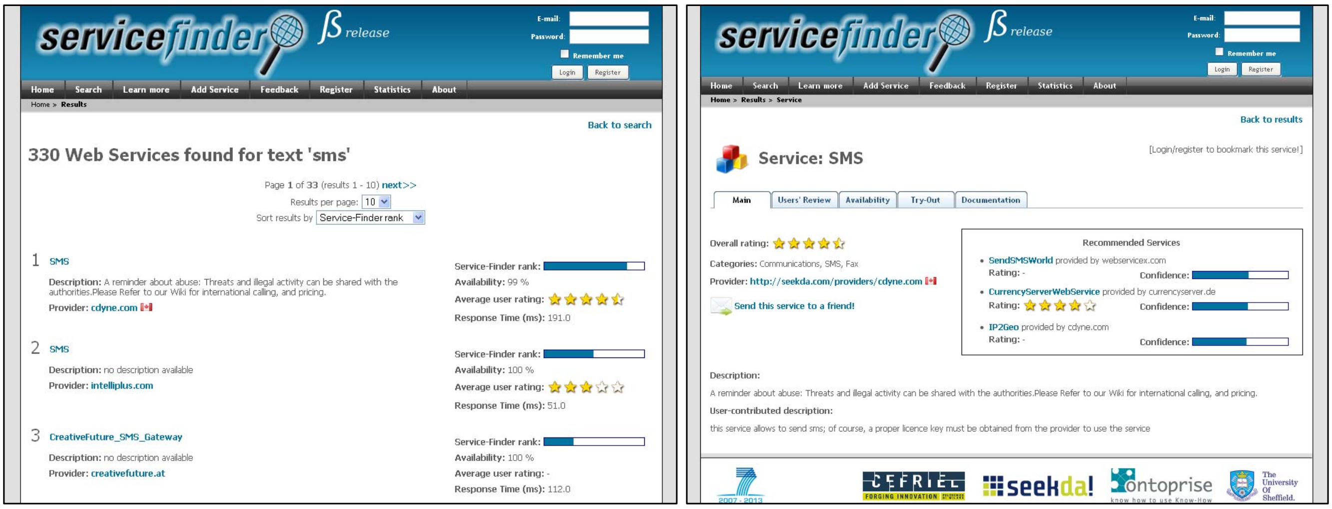

Figure 3: Screenshots of the Service-Finder portal: a search result page (on the left) and the service description page of the first result (on the right).

The above mapping excerpt says that, for all the resources of type `sfo:Service`, there is a piece of data – the value of the property `sfo:hasName` – which plays a double role in the presentation of the resource: it can be used as the title (`dc:title`) of the resource and it is a characteristic of a certain importance (`sd:RelevantProperty`), and therefore it should be always displayed (for example, both when the resource is briefly presented, like in a search result list, and when it is described in details). Moreover, each resource of that type has at most one value for that property and, in case this value is not present, it should be replaced by the string " - ".

The screenshots in Figure 3 give an idea of the visual rendering resulting from such a mapping: the same service named "SMS" is visualized in the search result list (image on the left) and in the detailed description page of the service itself (image on the right). In both cases, the value of the property `sfo:hasName` is put in evidence (since it was defined as a `sd:RelevantProperty`) and used as the title for the service (because it was mapped to `dc:title`).

While this is just a (partially simplified) example and while our mapping approach is just one of the possible implementations of the idea of drawing correspondences between two conceptual models, the presented example demonstrates the simplicity of our conceptual idea and the feasibility of its adoption.

The approach of the aforementioned example was systematically applied to all the entities and characteristics of the Service-Finder ontology, resulting in 76 definitions of `sd:PresentationMapping` instances, which involve 7 domain classes and 34 domain properties mapped for 102 times to 15 STAR:dust properties.

This mapping paved the ground for the construction of the whole Service-Finder portal, which presents data about more than 23,000 Web services and over 8,000 providers and was visited by more than 350 unique users, out of which around 60 registered and "frequent" users.

4.2 Evaluation of the results

While we generally agree with [23] on the difficulty of evaluating Semantic Web applications, we decided to evaluate the results of our proof of concept as a generic Web application. This allows us to concentrate our attention on the general usability of the resulting application, regardless the underlying adoption of Semantic Web technologies, thus letting us evaluate the feasibility of our concept in a generic and realistic scenario.

Taking inspiration from the analysis of [16], which highlights the attention points and the possible evaluation metrics of (Semantic) Web applications, we decided to assess the our approach in the Service-Finder portal along three dimensions: flexibility, performances and usability. The first dimension reflects the point of view of a developer who wants to use our approach to build a knowledge-intensive ontology-based Web application; since our claim is that our interaction semantics formalization ease this developer task, this is probably the main point of our approach evaluation. However, since the final beneficiaries of the development result are Web users, we also took into account their point of view, thus evaluating the performances and the usability of the Service-Finder portal.

From the developer point of view, an important factor to adopt a Web application framework is its *flexibility* to changing requirements and its *extendibility* in terms of features. In this respect, we admit that the current mapping approach is still complicated, due to the lack of proper support tools: we manually wrote the ontology mapping, while some more "user-friendly" tools can be envisioned. However, we are happy to say that the basic approach can be extended very easily: in the Service-Finder portal development we followed a "continuous beta" approach, adding from time to time new features and enabling the display and the manipulation of a growing number of data. Also the developers working on the system changed over time: 4 different developers collaborated to build the Service-Finder portal and a total number of 6 researchers dealt with the STAR:chart framework in its various instantiations.

Including new features, thus supporting new interaction patterns in the framework and then in the portal was very straightforward; the direct experience of the developers involved in the Service-Finder portal creation demonstrates that adding new elements in the STAR:ship ontology, new fruition modalities in the STAR:dust ontology and new combinations of the two to increase the capabilities of the framework was indeed very easy. Some developers were involved in advanced stages of the development and a short training was enough to learn how to add new interaction patterns.

For what regards *performance evaluation*, we employed some common Web testing tools, like the popular Firebug[8] and YSlow[9], which analyze the response time and other parameters as "experienced" by the final users. Among other indicators, YSlow measures: the number of HTTP requests; the presence, position and dimension of Javascript and CSS; the management of images; the caching of information, etc. We selected a set of testing pages (the Service-Finder homepage and search page, the search result pages for some sample queries, the description page for some sample services with a few or a lot of details) and we tested them with Firebug and YSlow, both installed as Firefox browser add-ons.

The results of our evaluations state that the Service-Finder portal has good response times (the data fetching and elaboration takes always less than half a second); some characteristics show room for improvement (like the management of scripts by the STAR:chart framework), but this was an expected conclusion, since ours is just a research prototype and not a commercial framework. All in all, we judge this result as very positive, since the operation of ontology mapping in the system does not negatively affect the portal response time. We believe that a proper engineering of our prototype would not make a poor impression if compared to more popular and robust CMS systems.

Finally, to evaluate the *usability* of the resulting portal, we employed the well-known System Usability Scale, usually known as SUS questionnaire [8]; it consists of ten sentences, which the compiler can agree or disagree on with a five-point Likert scale [19]; the final score of a SUS questionnaire is in the range 1-100 and a good evaluation (as suggested by [3]) is set between 65 and 70 at least. This kind of evaluation reflects the point of view of the system final user, but its results can be used by Web developers to understand the room for improvement of their applications.

We asked to our Service-Finder registered users to compile the SUS questionnaire; around 20 users participated to this evaluation campaign and the average SUS score for those evaluators was 72.32, which represents a global sign of satisfaction about the usability of the Service-Finder portal. In particular, our evaluators stated that "the system was very easy to use" (average score: 4.0/5) and they disagreed on the need for "the support of a technical person to be able to use this system" (average score: 1.5/5). The interested reader can find all the details about this and other evaluations of the Service-Finder portal in [7].

This result proves that our approach led to the implementation of a very usable portal, thus demonstrating that our approach is not only conceptually meaningful, but also technically sound and pragmatically useful.

[8]Cf. `http://getfirebug.com/`.

[9]Cf. `http://developer.yahoo.com/yslow/`.

5. CHALLENGES AND CONCEPT EVOLUTION

Far from being a complete and comprehensive formalization of interaction semantics, the work presented in this paper represents an attempt to join the Semantic Web research with the Interaction Design world. We believe that the studies about how to improve the user experience progressed so much in the last few years, that the knowledge technologies can be successfully employed to model the meaning of interaction paradigms and to operate those models to support the development of Web applications that respect the principles of user-friendliness.

It is easy to say that the next step in our exploration should be the completion and the systematization of the interaction patterns formalization: our ontologies should be improved, enriched and enhanced to cover a wider spectrum of patterns and interaction modes. Moreover, another missing piece to complete our picture could be the development of tools and methods to better support, ease and – if possible – automate the mapping phase. However, we do not want to restrict ourselves to those future works.

The rise of the Open Linked Data movement and the rapid and unprecedent growth of the so-called Linked Data Cloud demonstrate that ontologies and data describing a specific domain cannot be found only in "closed", centralized or bounded repositories; on the contrary, knowledge-intensive sources of data are more and more frequently discovered on the open and decentralized Web of Data [6]. What is more, new links and connections between linked data sources can emerge over time, thus enriching entities' descriptions. This fact poses new challenges, since on the one hand the data schemata cannot never be considered complete, but are "fluid" and possibly continuously changing; on the other hand, the more the data grows, the easier to find data inconsistencies.

The challenge of how to present linked data to the human user is therefore open. Currently, the research line seems to concentrate on Linked Data Browsers, i.e. generic tools that allow users to navigate between data sources by following links expressed as RDF triples. The first example of this browser was certainly Tabulator [5], conceived by the Web inventor Tim Berners-Lee; several other tools of this kind exist today, like Marbles[10] or Disco[11].

The capabilities of Linked Data Browsers to display in a "fancy" or meaningful way the data they give access to, however, is very limited: they are usually able to display only pictures in HTML `img` tags and connections to other resources as hyperlinks. While offering a "perfect" interface over the linked data cloud is not the purpose of this family of tools, it is undeniable that a broader approach keeping into account the basic rules of Interaction Design would improve a lot the usability of such tools (and their usefulness outside the community of Semantic Web researchers). Therefore, this is definitely a possible direction to improve our approach.

The impossibility to know in advance the complete schema of the data to be displayed, however, could impede the mapping approach that we proposed in this paper. Of course, it is always possible to define a "default" visual representation for all those resource properties that are present but

[10]Cf. `http://marbles.sourceforge.net/`.

[11]Cf. `http://www4.wiwiss.fuberlin.de/bizer/ng4j/disco/`.

were not "foreseen" (like the generic visualization of Linked Data Browsers). A method to retrieve not only the (linked) data, but also their expected visualization (a sort of metadata about the linked data) could provide a solution to this problem. In this regard, we would welcome a wider adoption of the vOID RDF vocabulary to describe linked datasets [1] and the further specification of the SPARQL endpoint Service Description [24].

Our approach towards the formalization of interaction semantics took into consideration also the collaborative and community patterns usually employed in Web 2.0 applications. The possibility to engage the final user in the editing, correction and enrichment of data is in our opinion a very valuable opportunity, especially in those cases in which a manual check or improvement of the data is needed. When talking about linked data, therefore, the same paradigm applies: the so-called "collective intelligence" of a Web community could also be leveraged to clean or integrate the data. To this end, displaying the linked data is not enough, because the user need also other kinds of information that help him to successfully complete his tasks, like the source of data or the timestamp.

A first application that display the data together with its provenance information and retrieval date is sig.ma [9]; even if visually very confusing for the casual user, this application shows the importance of a correct understanding of the data "context": whenever an inconsistent piece of information is present, in order to resolve the inconsistency a user should know something else (its source or its time validity). This is another direction in which we would like to extend our approach and our research: what is the best and more natural way to present provenance or time information to the user? What are the correct interaction patters to be employed to support user editing? We would like to be able to provide usable and well designed applications that convey the functionalities of sig.ma.

6. CONCLUSIONS

In this paper we presented our early attempts to formalize with ontological artifacts the interaction semantics in terms of Web design patterns and user fruition modalities. We also illustrated a successful proof of concept, a complete implementation of our basic idea to the development of a complete and highly usable Web application and its evaluation.

Since we believe this is only the starting point in the direction of a better interplay of Interaction Design studies and Semantic Web research, we also offered our reflections and our thoughts about the possible extensions of the proposed approach, especially with regards to the field of linked data for the human users.

Acknowledgments

This research has been partially supported by the Service-Finder EU co-funded project (FP7-IST-215876) and by the NeP4B Italian co-funded FIRB project (MIUR-2005).

7. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *Procedings of the Linked Data on the Web Workshop (LDOW 09) at WWW09*, 2009.

[2] S. Auer, R. Doehring, and S. Dietzold. LESS - Template-based Syndication and Presentation of Linked Data for End-users. In *7th Extended Semantic Web Conference (ESWC2010)*, June 2010.

[3] A. Bangor, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4:114–123, May 2009.

[4] T. Berners-Lee. Primer: Getting into RDF and Semantic Web using N3. Technical report, 2005. Available from: http://www.w3.org/2000/10/swap/Primer.

[5] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and Analyzing Linked Data on the Semantic Web. In *Procedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06)*, 2006.

[6] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[7] S. Brockmans, I. Celino, D. Cerizza, D. Dell'Aglio, E. Della Valle, M. Erdmann, A. Funk, H. Lausen, and N. Steinmetz. D7.5 - Final Report on Assessment of Tests for Beta Release, 2009. Available from: http://www.service-finder.eu/attachments/D7.5.pdf.

[8] J. Brooke. SUS: a "quick and dirty" usability scale. In P.W. Jordan and B. Thomas and B.A. Weerdmeester and A.L. McClelland, editor, *Usability Evaluation in Industry*. Taylor and Francis., London, 1996.

[9] M. Catasta, R. Cyganiak, and G. Tummarello. Towards ECSSE: live Web of Data search and integration. In *Procedings of the Semantic Search 2009 Workshop at WWW2009*, 2009.

[10] I. Celino, D. Cerizza, F. Corcoglioniti, A. Guarino, A. Turati, and E. Della Valle. STAR:chart – Preserving Data Semantics in Web-Based Applications. In *Proceedings of the 12th International Conference on Business Information Systems, BIS 2009, Poznan, Poland*, volume 21 of *Lecture Notes in Business Information Processing*, pages 97–108, 2009.

[11] I. Celino and E. Della Valle. Multiple Vehicles for a Semantic Navigation Across Hyper-environments. In A. Gómez-Pérez and J. Euzenat, editors, *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, volume 3532 of *Lecture Notes in Computer Science*, pages 423–438. Springer, 2005.

[12] S. Ceri, P. Fraternali, and A. Bongio. Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks*, 33(1-6):137–157, 2000.

[13] C. W. Choo, B. Detlor, and D. Turnbull. Information Seeking on the Web: An Integrated Model of Browsing and Searching. *First Monday*, 5(2), 2000.

[14] E. Della Valle and M. Brioschi. Toward a Framework for Semantic Organizational Information Portal. In C. Bussler, J. Davies, D. Fensel, and R. Studer, editors, *Proceedings of the 1st European Semantic Web Symposium (ESWS 2004)*, volume 3053 of *Lecture Notes in Computer Science*, pages 402–416. Springer, 2004.

[15] E. Della Valle, D. Cerizza, I. Celino, A. Turati,

H. Lausen, N. Steinmetz, M. Erdmann, W. Schoch, and A. Funk. Realizing Service-Finder – Web Service Discovery at Web Scale. In *Proceedings of the 2nd European Semantic Technology Conference, ESTC 2008, Vienna, Austria*, 2008.

[16] T. Heath, J. Domingue, and P. Shabajee. User Interaction and Uptake Challenges to Successfully Deploying Semantic Web Technologies. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI 2006), collocated with ISWC 2006*, 2006.

[17] S. Henninger. Disseminating Usability Design Knowledge through Ontology-Based Pattern Languages. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI 2006), collocated with ISWC 2006*, 2006.

[18] H. Lausen, Y. Ding, M. Stollberg, D. Fensel, R. Lara Hernandez, and S.-K. Han. Semantic web portals: state-of-the-art survey. *Journal of Knowledge Management*, 9(5):40–49, 2005.

[19] R. Likert. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 140:1–55, 1932.

[20] Ontoprise. OntoBroker 5.0. User Manual. Technical report, October 2007. Available from: http://www.ontoprise.de/.

[21] T. O'Reilly. Google Admits "Data is the Intel Inside". http://radar.oreilly.com/archives/2007/12/google-admits-data-is-the-inte.html, 2007.

[22] E. Pietriga, C. Bizer, D. Karger, and R. Lee. Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In *5th International Semantic Web Conference (ISWC2006)*.

[23] J. van Ossenbruggen, A. Amin, and M. Hildebrand. Why Evaluating Semantic Web Applications is Difficult. In *Proceedings of the Fifth International Workshop on Semantic Web User Interaction (SWUI 2008), collocated with CHI 2008*, 2008.

[24] G. T. Williams. SPARQL 1.1 Service Description – W3C Working Draft 1 June 2010. Technical report, 2010. Available from: http://www.w3.org/TR/sparql11-service-description/.