Shared and Personal Views on Collaborative Semantic Tables

(Article begins on next page)

Shared and Personal Views on Collaborative Semantic Tables*

Anna Goy, Diego Magro, Giovanna Petrone, Claudia Picardi, Marino Segnan

Dipartimento di Informatica, Università di Torino, Torino, Italy
{annamaria.goy, diego.magro, giovanna.petrone, claudia.picardi, marino.segnan}@unito.it

Abstract. The scenario defined by current Web architectures and paradigms poses challenges and opportunities to users, in particular as far as collaborative resource management is concerned. A support to face such challenges is represented by semantic annotations. However, especially in collaborative environments, disagreements can easily rise, leading to incoherent, poor and ultimately useless annotations. The possibility of keeping track of "private annotations" on shared resources represents a significative improvement for collaborative environments. In this paper, we present a model for managing "personal views" over shared resources on the Web, formally defined as structured sets of semantic annotations, enabling users to apply their individual point of view over a common perspective provided in shared workspaces. This model represents an original contribution and a significative extension with respect to our previous work, even being part of a larger project, SemT++, aimed at developing an environment supporting users in collaborative resource management on the Web.

Keywords: Collaborative Workspaces · Personal Information Management · Personal Views · Ontology-based Content Management · Semantic Technologies.

1 Introduction

Human-computer interaction has greatly changed in the last decade, due to the wide availability of devices and connectivity, and to the consequent evolution of the World Wide Web. In particular, Personal Information Management [1] is facing new challenges: (a) Users have to deal with a huge number of heterogeneous resources, stored in different places, encoded in different formats, handled by different applications and

* POSTPRINT VERSION. Cite as: A. Goy, D. Magro, G. Petrone, C. Picardi, M. Segnan. Shared and Personal Views on Collaborative Semantic Tables. In P. Molli, J. Breslin, M. E. Vidal (Eds.), Semantic Web Collaborative Spaces, LNCS 9507, pp. 13–32, Springer, 2016. DOI 10.1007/978-3-319-32667-2_2

belonging to different types (images, emails, bookmarks, documents, ...), despite their possibly related content. (b) Web 2.0 and, more recently, Cloud Computing, in particular the Software-as-a-Service paradigm, have enhanced the possibility of user participation in content creation on the Web, as well as the possibility of resource and knowledge sharing. The interaction of these two aspects provided a great impulse to user collaboration in managing shared resources.

A first step in the direction of providing users with a smart support to face these challenges is represented by semantic technologies, and in particular by semantic annotations. However, in collaborative environments, where users have to provide coherent semantic annotations of shared resources, disagreements can rise, leading to incoherent, poor and ultimately useless annotations: Either the team works subtractively, keeping only what everyone agrees upon (the resulting annotation being much less useful to everyone), or addictively, keeping everything (which leads to a confusing annotation).

In order to solve these problems, the possibility of keeping track of "private annotations" on shared resources can represent a great improvement for collaborative environments supporting semantic annotation of shared resources. The idea is to provide users with a personal view over shared resources, where their annotations remain stored independently from what other team members do. Personal annotations cooperate with shared ones in the organization and retrieval of shared resources.

In this paper, we present a model for managing such personal views over shared resources on the Web. In particular, in our model, personal views are structured sets of semantic annotations, enabling users to apply their individual point of view over a common perspective provided in shared workspaces. The presented model is part of a larger project, *Semantic Table Plus Plus* (SemT++), which will be briefly described in the following, in order to provide the framework for the personal views model.

The rest of the paper is organized as follows: In Section 2 we discuss the main related work, representing the background of our work; in Section 3, we briefly present the main features of SemT++, and in Section 4 we describe the model supporting personal views. Section 5 concludes the paper by discussing open issues and future developments.

2 Related Work

A survey and a discussion of existing Web-based applications supporting collaboration, including groupware and project management tools or suites, can be found in [2] and [3], where T++ was introduced. Strategies for organizing resources have been studied within the field of Personal Information Management, where one of the most relevant research topic is well represented by [4], a book by Kaptelinin and Czerwinski containing a survey of the problems of the so-called *desktop metaphor* and of the approaches trying to replace it. In this perspective, an interesting family of approaches are those grounded into Activity-Based Computing (e.g., [5, 6]), where *user activity* is the main concept around which the interaction is built. A similar approach is proposed in [7] and [8], where the authors describe a system supporting collaborative interactions by relying on activity-based workspaces handling collections of heterogeneous resources. Another interesting model discussed in the mentioned book is Haystack [9], a system enabling users to define and manage workspaces referred to specific tasks. The most interesting feature of Haystack workspaces is that they can be personalized.

A research field that is particularly relevant for the approach presented in this paper is represented by studies about systems supporting multi-facets classification of resources. In these systems, resources can be tagged with metadata representing different aspects (*facets*), leading to the creation of bottom-up classifications, collaboratively and incrementally built by users, usually called *folksonomies* [10]. Interesting improvements of tagging systems have been designed by endowing them with semantic capabilities (e.g., [11]), in particular in the perspective of knowledge management [12].

Another important research thread, aiming at enhancing desktop-based user interfaces with semantic technologies is the Semantic Desktop project [13]. In particular, the NEPOMUK project (nepomuk.semanticdesktop.org) defined an open source framework for implementing semantic desktops, aimed at the integration of existing applications and the support to collaboration among knowledge workers, while [14] presents an interesting model connecting the Semantic Desktop to the Web of Data.

A different research area that is relevant for the approach presented in this paper is represented by the studies about resource annotation. The simplest tools supporting annotation enable users to add comments (like sticky notes) to digital documents (e.g., www.mystickies.com, among many others). In these tools, typically, no semantics is associated with user annotations. At the opposite side of the spectrum, we can find NLP-oriented annotation tools, in which annotations are usually labels, referring to a predefined annotation schema, associated with phrases within a document. Some of these systems support collaboration among annotators (e.g., GATE Teamware [15], or Phrase Detectives [16]). Many other approaches provide frameworks for semantic annotation in different domain: Uren et al. [17] include a survey of annotation frameworks with a particular attention to their exploitation for knowledge management, while Corcho [18] surveys ontology-based annotation systems.

As far as the support to shared and personal views on resource annotations is concerned, existing systems tend to focus on a single perspective, sometimes favoring the shared one (e.g., in wikis), sometimes favoring the personal one (e.g., in social bookmarking systems). However, there are some research works which try to integrate shared and personal annotations, like for instance [19], where the need for supporting personal annotations in collaborative environments is motivated. There have been also efforts to provide users with the possibility of adding both private and public notes to digital resources (e.g. [20]). An interesting survey of tools supporting collaborative creation of different types of structured knowledge can be found in [21]: The authors conclude by listing features that users would like to have in collaborative tools supporting knowledge creation, among which "having private and public spaces". With respect to this aspect, the focus of our approach is on structured (ontology-based) *semantic* annotations (mainly describing resource content), and we aim at supporting the integration of both perspectives, enabling users to clearly see at a glance both shared and personal annotations.

3 Overview of SemT++

The SemT++ project is an enhancement of T++, which is described in [2] and [3]. The T++ environment allows users to collaboratively manage digital resources. It is based on the metaphor of *tables* populated by *objects* and it has the following main features.

Tables as thematic contexts. In T++, users can define shared workspaces devoted to the management of different activities. Such workspaces are called *tables* and support users in the separated, coherent and structured management of their activities. Users can define new tables, at the preferred granularity level; for instance, a table can be used to manage a work project, to handle children care, or to plan a journey.

Uniform management of heterogeneous objects and workspace–level annotations. Objects lying on tables can be resources of any type (documents, images, videos, to-do items, bookmarks, email conversations, and so on), but T++ provides an abstract view over such resources by handling them in a homogeneous way. Table objects, in fact, are considered as *content items* (identified by a URI) and can be uniformly annotated (by comments and annotations).

Workspace awareness. Workspace awareness is supported by three mechanisms: (a) On each table, a presence panel shows the list of table participants, highlighting who is currently sitting at the table; moreover, when a user is sitting at a table, she is (by default) "invisible" at other tables (*selective presence*). (b) Standard awareness techniques, such as icon highlighting, are used to notify users about table events (e.g., an object has been modified). (c) Notification messages, coming from outside T++ or from other tables, are filtered on the basis of the topic context represented by the active table (see [22] for a more detailed discussion of notification filtering).

User collaboration. An important aspect of T++ tables is that they are collaborative in nature, since they represent a shared view on resources and people: "Tables represent *common places* where users can, synchronously or asynchronously, share information, actively work together on a document, a to-do list, a set of bookmarks, and so on" [2, p. 32]. Table participants, in fact, can (a) invite people to "sit at the table" (i.e., to become a table participant); (b) modify and delete existing objects, or add new ones; (c) define metadata, such as comments and annotations (see below).

T++ has been endowed with semantic knowledge, with the goal of offering users a smarter support to resources management and sharing: SemT++ is, thus, the semantic, enhanced version of T++. In the following we will present SemT++ architecture and prototype (Section 3.1 and 3.2) and we will summarize the semantic model implemented in SemT++.

3.1 Architecture

The architecture of SemT++ is shown in Fig. 1.

The *User Interaction Manager* handles the interaction with users, i.e., the generation of the User Interface and data exchange with the TO Manager.

The *TO* (*Table Objects*) *Manager* handles the processes implementing the business logic of the system, namely all the operations taking place on SemT++ tables (e.g., adding/deleting objects, comments, etc.).

The *Smart Object Analyzer* analyzes table objects and extracts information about them; for instance, it finds object parts (e.g., images, links, etc.), it detects the language used and the encoding formats.

The *TO* (*Table Objects*) *Semantic Knowledge Manager* manages the semantic descriptions of table objects, stored in the *TO Semantic KB*, and based on the *Table Ontology*, i.e., the system semantic knowledge concerning information objects (see Section 3.3).

The *Domain Knowledge Manager* is in charge of the semantic knowledge concerning the content of table objects, stored in the *Domain Knowledge Bases*, and based on one or more *Domain Ontologies*, representing the system semantic knowledge concerning the domain to which table resources refer (see Section 3.3). Moreover, the Domain Knowledge Manager handles the connection with Linked Open Data (LOD). The TO Semantic Knowledge Manager and the Domain Knowledge Manager also invokes the Reasoner, when required.

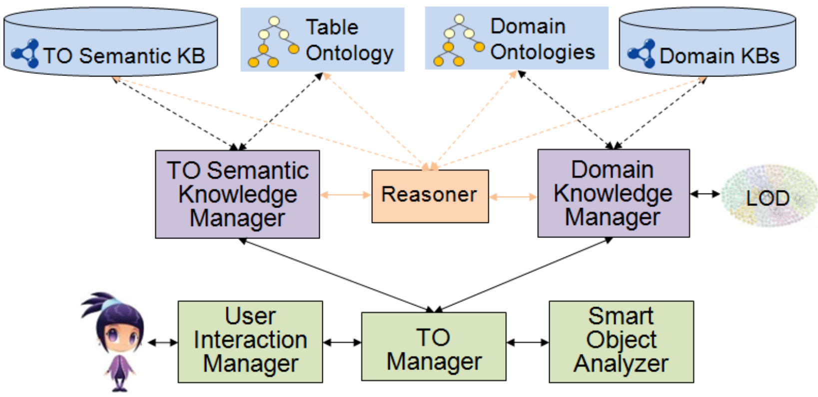

Fig. 1. Sem T++ architecture

3.2 Prototype and Evaluations

The sketched architecture has been implemented in a proof-of-concept prototype: The backend is a cloud application (a Java Web application deployed on the Google App Engine), while the frontend - implemented by the User Interaction Manager – is a dynamic, responsive Web page, implemented in Bootstrap (getbootstrap.com), using AJAX and JSON to connect to server-side modules (Java Servlets) and to exchange data with the backend.

In the current version of the prototype, the TO Manager relies on Dropbox and Google Drive API to store files corresponding to table objects and Google Mail to handle email conversations. The Smart Object Analyzer exploits a Python Parser Service, which provides the analysis of table objects; currently, it analyzes HTML documents.[1] The Table Ontology and the Domain Ontologies are written in OWL (www.w3.org/TR/owl-features); the TO Semantic Knowledge Manager and the Domain Knowledge Manager exploit the OWL API library (owlapi.sourceforge.net) to interact with them, while the Reasoner is based on Fact++ (owl.cs.manchester.ac.uk/tools/fact). The knowledge bases, containing assertions concerning the semantic description of table objects, are stored in a Sesame (rdf4j.org) RDF triplestore, accessed by the TO Semantic Knowledge Manager and the Domain Knowledge Manager through Sesame API.

In order to evaluate our approach, we implemented a testbed case of domain knowledge, i.e., we instantiated a Domain Ontology on geographic knowledge, while the Domain Knowledge Manager connects to GeoNames Search Web Service (www.geonames.org/export), as a significative example of LOD dataset. Further details about this choice can be found in [23].

We evaluated the most important functionalities of SemT++ through some user tests.

Goy et al. [3] report the results of a user evaluation of T++ in which we asked users to perform a sequence of pre-defined collaborative tasks (communication, resource sharing, and shared resources retrieval) using standard collaboration tools (like Google Drive and Skype) and using T++. The results demonstrate that performing the required tasks with T++ is faster and it increases user satisfaction.

Goy et al. [24] present the results of an empirical study about the impact of the semantic descriptions. Potential users of SemT++ were asked to go through a guided interaction with SemT++, aimed at selecting table objects on the basis of multiple criteria offered by their semantic descriptions; then, participants answered a post-test questionnaire. The analysis of users' answers confirmed our hypothesis: An environment supporting the uniform management of different types of resources (documents, images, Web sites, etc.) and the possibility of selecting them by combining multiple criteria (among which resource content) is highly appreciated, and contributes to provide an effective access to shared resources and a less fragmented user experience.

Goy et al. [25] describe a qualitative user study aimed at analyzing user requirements and defining the model supporting collaborative semantic annotation of table objects. Participants were organized into small groups and were asked to collaboratively annotate shared resources, by providing annotations in the form of tags, describing the resources content. Each group experimented different collaboration policies (unsupervised vs supervised, with users playing different roles). At the end of the experiment, participants were asked to fill in a questionnaire where they had to rate their experience and express their opinions about different features. From this user

---

[1] Besides being a very common format, quite easy to parse, HTML poses interesting challenges to the semantic modeling, since it introduces a further layer – the HTML encoding – between the "digital object", encoded for instance in UTF-8, and the information content representing the Web page itself. We are extending the Smart Object Analyzer functionality in order to analyze other formats.

study, we extracted a set of guidelines for designing the model handling collaborative semantic annotation of table objects in SemT++.

3.3 Semantic Model

The core of the approach used in SemT++ to provide users with a flexible and effective management of shared heterogeneous resources is its semantic model, represented by the ontologies and knowledge bases introduced above.

In the following, we will briefly describe it, before concentrating on the focus of this paper, i.e. the framework supporting *personal views* over annotations of shared resources. A more detailed description of SemT++ semantic model can be found in [24] and [23].

The Table Ontology models knowledge about information resources. It is grounded in the Knowledge Module of O-CREAM-v2 [26], a core reference ontology for the Customer Relationship Management domain developed within the framework provided by the foundational ontology DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [27] and some other ontologies extending it, among which the Ontology of Information Objects (OIO) [28]. The Table Ontology enables us to describe resources lying on tables as *information objects*, with properties and relations. for instance: A table object (e.g., a document) can have parts (e.g., images), which are in turn information objects; it can be written in English; it can be stored in a PDF file, or it can be a HTML page; it has a content, which usually has a main topic and refers to a set of entities (i.e., it has several objects of discourse). Given such a representation, reasoning techniques can be applied, in order to infer interesting and useful knowledge; for example, if a document contains an image of the Garda lake, probably the document itself is (also) about the Garda lake.

The most relevant class in the Table Ontology is *InformationElement*, with its subclasses: *Document*, *Image*, *Video*, *Audio*, *EmailThread*, etc. All table objects are instances of one of them. In order to characterize such classes, we relied on a set of properties (some of them inherited from O-CREAM-v2) and a language taxonomy defined in O-CREAM-v2, representing natural, formal, computer, visual languages. A complete account of such properties is out of the scope of this paper; in the following we just mention the most important ones:

- *DOLCE* : *part*(*x, y, t*) – it represents relations such as the one between a document (*x*) and an image or a hyperlink (*y*) included in it, holding at time *t*.[2]
- *specifiedIn*(*x, y, t*) – it represents relations such as the one between a document (*x*) and the language (*y*) it is written in (e.g., Italian), holding at time *t*.
- *hasAuthor*(*x, y, t*) – it represents the relation between an information element (*x*) and its author (*y*), holding at time *t*.
- *hasTopic*(*x, y, t*) – it represents the relation between an information element (*x*) and its main topic (*y*), holding at time *t*.

[2] Parameter *t*, representing time, is omitted in the OWL version of the Table Ontology.

— *hasObjectOfDiscourse*(*x, y, t*) – it represents the relation between an information element (*x*) and the entity (*y*) it "talks about", holding at time *t;* it is a subproperty of *OIO* : *about*.
— *identifies*(*x, y, t*) – it represents, for instance, the relation between a hyperlink (*x*) and the resource (*y*) it points to, holding at time *t*.

As we mentioned above, we chose commonsense geographic knowledge as a testbed example of system domain competence. This knowledge is represented by a Geographic Ontology coupled with a Geographic KB, containing information retrieved from GeoNames, a huge, open geographical database containing over 10 million geographical entities. For each topic and object of discourse used to describe the content of resources on a given table, the Domain Knowledge Manager searches for corresponding GeoNames entities. If the search result is not void, after a possible disambiguation phase (currently done by the user), the instance representing that topic/object of discourse is classified in the Geographic Ontology (see [23] for more details about the geography testbed).

Fig. 2 graphically depicts a simplified example of the semantic description of a table object, i.e., a Web page, with the following properties: The author is a company for touristic promotion of Garda lake (*VisitGarda inc.*); it is written in English; it is encoded in HTML (specifically UTF-8/HTML5); it contains a figure (*image1*) and a link (*link1*) to a brochure (*windsurf-brochure-content-1*); its main topic is the Garda lake, and it talks about trekking and climbing, restaurants and events in the lake area, windsurfing, and spas. Since the topic (*Garda lake*) is a geographic entity, the Geographic Knowledge Manager found the corresponding GeoNames entity (*GeoNames: Lake Garda*), and classified it as an instance of the class *Lake* (in the Geographic Ontology).

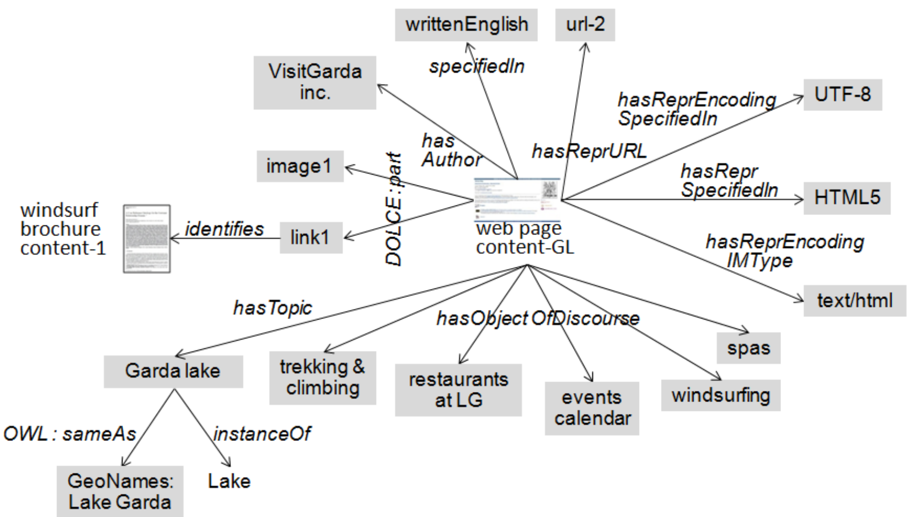

Fig. 2. Simplified example of the semantic description of a table object (Web page)

An important fragment of the proposed semantic model refers to *candidate relationships*: The Reasoner, on the basis of axioms like the following one, can infer *candidate* features, mainly from included objects:

InformationElement(x) ∧ DOLCE : part(x, z, t) ∧ hasObjectOfDiscourse(z, y, t)
→ hasCandidateObjectOfDiscourse (x, y, t)

For example, the Reasoner can infer that the city of Trento (*y*) is a candidate object of discourse of a document (*x*) – at time *t* – from the fact that the document itself (*x*) includes a video (*z*) about Trento (*y*) – at time *t*.

When the Reasoner infers such candidate relationships, the system asks the user for a confirmation: If (and only if) the user confirms, for instance, that *Trento* is actually an object of discourse of the *document*, then a new relation *hasObjectOfDiscourse(doumentc, Trento, t)* is added to the knowledge base.

The semantic descriptions of table objects based on the model just sketched enable table participants to specify and combine multiple criteria in order to select objects on a table. For example, to get all email threads talking about Garda lake windsurfing (i.e., having it as main topic), a user can specify the following parameters: *topics={Garda_lake_windsurfing}, types={emailThread}*. The current User Interface enabling such a functionality is a sequence of simple Web forms, and is described in [24]. Moreover, the user could provide more general queries, such as asking for all resources talking about lakes, thanks to the facts that topics and objects of discourse (e.g., *Garda_lake*) are represented as instances of classes in the Geographic Ontology (e.g., *Lake*). Finally, specific information about topics and objects of discourse (characterizing the content) retrieved from LOD sets, such as GeoNames, can provide table participants with a sort of "explanation" about such property values.

3.4 Collaborative Semantic Annotation

When a new object is added to a table, or when an existing one is modified (e.g., when a table participant includes a new image in it), the corresponding semantic representation is created or updated. Consider the new object case (the update case is analogous): The semantic representation is created (updated) as follows:

- The Smart Object Analyzer sets some property values (e.g., *DOLCE* : *part*, properties related to encoding formats), and proposes candidate values for other properties (e.g., *specifiedIn*, *hasAuthor*).
- The Reasoner, invoked by the Semantic Knowledge Manager, proposes other candidates (e.g., topics and objects of discourse).
- The user can confirm or discard candidate values, and add new ones.

The property values that mostly depend on the personal view of each table participant about the content (and scope) of table resources are *hasTopic* and *hasObjectOfDiscourse*. In the case of *hasTopic*, table participants need to agree on a single value expressing the main topic of the resource; in the case of *hasObjectOfDiscourse*, they need to agree on a set of values.

In order to facilitate collaboration and the achievement of an agreement, we designed and implemented a collaboration model for handling semantic annotations on table resources, based on the outcomes of a qualitative user study. Both the study and the implemented model are described in [25]. In the following, we briefly summarize the most relevant features of SemT++ collaboration model.

SemT++ provides three alternative collaboration policies: (1) *Consensual*, where the editing of semantic descriptions of table objects is always possible for all participants in a totally "democratic" way. (2) *Authored*, where the final decision about the semantic annotation of a table resource is taken by its creator (owner). (3) *Supervised*, where the final decision about the semantic annotation of a table resource is taken by the table supervisor. SemT++ enables the table creator to select the collaboration policy to be applied for handling the collaborative process of building semantic descriptions of table objects. Moreover, a resource semantic description can also be simply marked as "approved" by participants (see the checkbox at the bottom of Fig. 4 and 5). Finally, SemT++ explicitly encourages table participants to use the communication tools available on the table, i.e., the Blackboard for posting asynchronous messages, the Chat for synchronous communication, and free-text Comments which can be attached to table objects, prompting them to add an optional comment whenever they edit the annotation (see also [29]).

4 Personal Views Over Shared Resources

One of the most interesting results of the user study discussed in [25] is that many users said they would be interested in the possibility of having *personal annotations*, visible only to the author of the annotation itself. Users explained that they would see this functionality as particularly useful for the search and retrieval of table resources based on tags describing their content. Moreover, the importance of supporting personal annotations in collaborative environments has also been claimed in the literature; see, for instance, [19].

Starting from this suggestion, we designed a new functionality for SemT++ enabling table participants to keep their own perspective over table resources. Results observed during the empirical study suggest that disagreements about semantic annotations of shared resources are quite common, especially as far as resource content is concerned. This fact reflects the intuitive common experience that people often have different interpretations of the "meaning" of an information object (e.g., a document, a movie), and it can be quite difficult to reach a consensus about the list of issues/topics it is about. The availability of *personal views* over semantic annotations, within a collaborative environment, represents an advantage, since users can maintain "private" annotations (possibly sources of disagreement) over shared resources.

SemT++ view management resorts on semantic annotations (i.e., semantic properties) of an information object to collaboratively handle resources on the Web. To illustrate a semantic annotation, suppose that x represents an information object, y

corresponds to an entity, and *t* represents a timestamp, then the property assertion *hasObjectOfDiscourse(x, y, t)* is a semantic annotation of the object *x* (see Section 3.3). From the point of view of a single table participant (*tp*), each semantic annotation can be as follows:

- Case A: visible to all table participants (including *tp*), but not explicitly "liked" by *tp*;
- Case B: visible to all table participants (including *tp*) and "liked" by *tp*;
- Case C: visible only to *tp*.

Given a SemT++ table, a *shared view* corresponds to the set of all semantic annotations that fall in cases A or B, while a *personal view* is the set of all semantic annotations in cases B or C. Moreover, we say that a table participant *likes* an annotation to mean that she agrees with it and thus she has explicitly imported it from the *shared view* into her *personal view*. When a table participant (*tp*) adds an annotation to a table object, she can decide to add it to the *shared view* (in this case it is also automatically marked as *liked* by *tp*, ending up in case B), or only to her *personal view* (case C). An annotation initially added to the *personal view* (C) can later on be shared (i.e., moved to case B). Moreover, as already mentioned, *tp* can mark as *liked* annotations added by other participants and belonging to the *shared view* (which means moving an annotation from case A to case B). Finally, *tp* can delete annotations:

- If the annotation belongs to case A, it is deleted from the *shared view* (see [25] for a detailed account of collaboration policies handling decisions about annotation removal), but maintained in the *personal views* of users who *like* it;
- If the annotation belongs to case B, it is deleted from the *shared view*, maintained in the *personal views* of users who *like* it, but deleted from *tp personal view*;
- If the annotation belongs to case C, it is simply deleted from *tp personal view*.

To support workspace awareness, *tp* can see the author of an annotation and users who like it, by right-clicking on it.

As we mentioned above, the availability of shared and personal views over semantic annotations is particularly interesting for annotations representing the content of table resources, i.e., the *hasTopic* and *hasObjectOfDiscourse* properties. However, the mechanism is available for all "editable" annotations, i.e. property values that are ultimately set by users (e.g., *specifiedIn*, representing the natural languages used in a document, or *hasAuthor*, representing document authors). Property values that are set by the system, e.g., mereological composition and encoding formats (see Section 3.4), are instead automatically assigned to case B for every table participant: They automatically belong to the shared view and everybody likes them, i.e., they belong also to all personal views.

In the following we will provide a usage scenario presenting shared and personal views on SemT++ tables (Section 4.1), focusing on the most interesting property with respect to this issue, i.e., *hasObjectOfDiscourse*; then, we will describe the underlying mechanisms enabling views management (Section 4.2), and sketch our evaluation plan (Section 4.3).

4.1 Usage Scenario

Aldo is a volunteer working for Our Planet, a NGO for environment safeguard. Some months ago he created a table (named *Our Planet*) to collaborate with a small team of other local volunteers. Now Aldo has to write an article for an online local newspaper, discussing the situation of the Champorcher mule track: To this purpose, he needs to retrieve information about that topic, available on the *Our Planet* table. He thus asks for the *topics* present on the table, selects *Champorcher mule track*, and gets the list of table objects having it as main topic. Among the results there is a resolution by the Municipality of Champorcher concerning an enlargement project, and two images of Champorcher surroundings. After reading the Municipality's resolution, Aldo creates a new table object (an HTML document, since the article will be published online), writes some text in it, adds one of the just retrieved images, and includes a link to the resolution.

When Aldo decides to leave the table and clicks on the "save&update" button, the table asks him to take a look at the annotations describing his article. Since the resource is new, only the following properties have already been set by the system (see Section 3.4):

- *Type*: *Document* (i.e., the class in the Table Ontology to which Aldo had assigned the resource when created);
- *Format*: *HTML* (referring to formal properties such as *hasRepresentationSpecifiedIn*(*x, y, t*));
- *Contains*: link to *resolution, image23* (referring to the formal property *DOLCE* : *part*(*x, y, t*)).

For the following properties, SemT++ suggests some candidate values:

- *Main topic* (referring to the formal property *hasTopic*(*x, y, t*));
- *Objects of discourse* (referring to the formal property *hasObjectOfDiscourse*(*x, y, t*)) - candidate objects of discourse are visible in Fig. 3;
- *Language* (referring to the formal property *specifiedIn*(*x, y, t*));
- *Authors* (referring to the formal property *hasAuthor*(*x, y, t*)).

The window displaying the properties of the new table object (Aldo's article) is shown in Fig. 3: The panel referring to objects of discourse is open; immediately below there is a text field where the user can add new values (objects of discourse in this case). Moreover, the panel with SemT++ suggestions (i.e., candidate objects of discourse) is available.

Aldo can select suggested values by clicking on them, or write new ones in the text field, with the support of an autocompletion functionality, based on the values already available on the table.

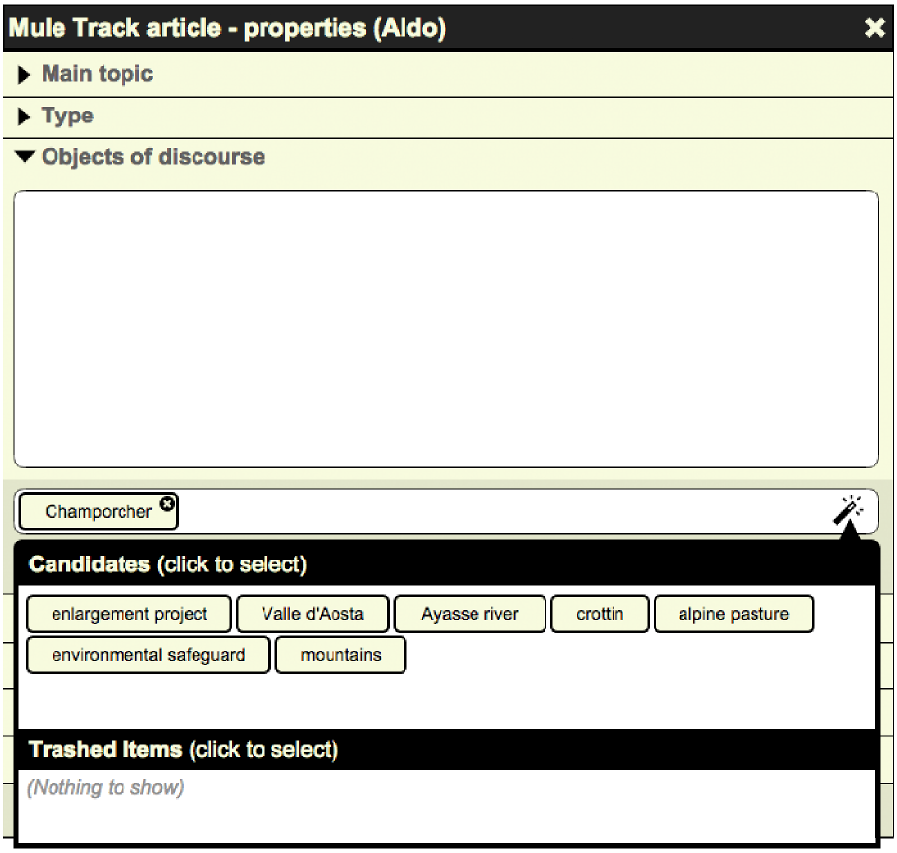

Fig. 3. SemT++ UI displaying semantic annotation of a new table resource

In our scenario, Aldo – besides confirming the suggested main topic, language and author – selects the following objects of discourse suggested by the system: *Champorcher*, *enlargement project*, *Ayasse river*; it also adds *Mont Avic park* and *demonstration 14/5*. His choices can be seen in Fig. 4, which represents the perspective of another table participant, Maria, on the same resource. When adding values to a property, whether new values or suggested ones, the user can choose to add them only to her/his personal view or also to the shared view (via the drop-down menu next to the + button, Fig. 4). In our scenario, Aldo has added all values to the shared view, thus making them visible to all table participants (including Maria).

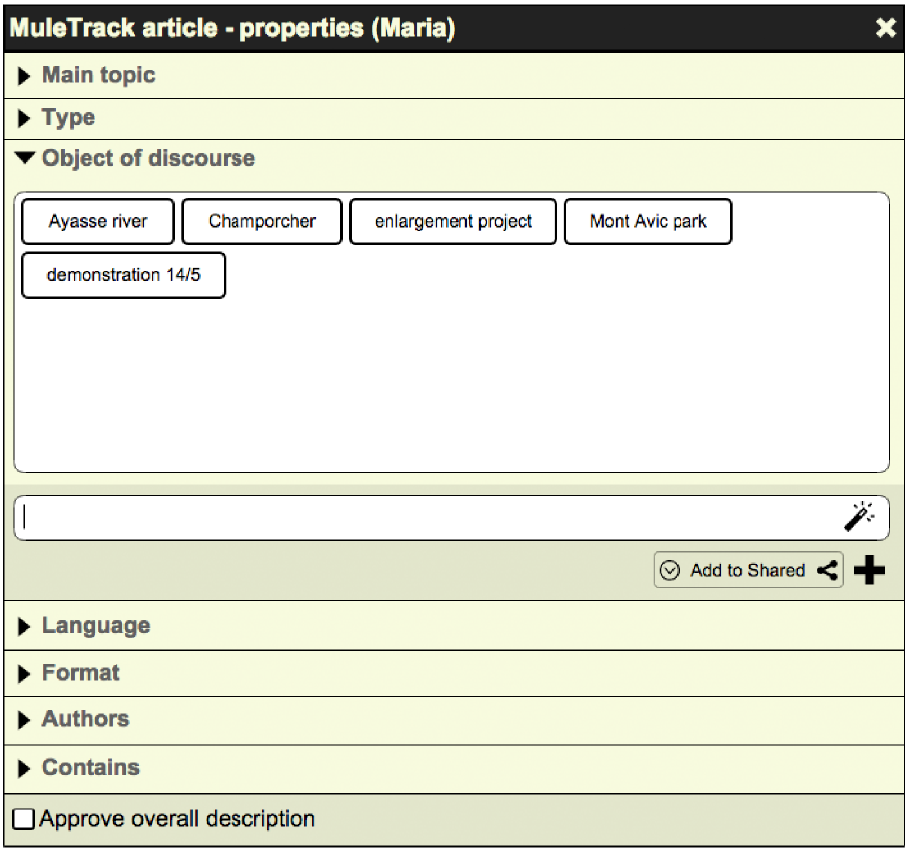

Fig. 4. SemT++ UI displaying semantic annotation of an already annotated table resource

When, later on, Maria sits at the *Our Planet* table, she reads Aldo's article and then accesses its properties (Fig. 4). By right-clicking on a value, Maria can see the author of the annotations (Aldo in our current example) and users who agree with (*like*) them (again, only Aldo in this case). She decides to mark as *liked* some values (i.e., *Champorcher*, *enlargement project*, *Mont Avic park*) and to add to the shared view a couple of new objects of discourse, i.e. *Our Planet* and *Legambiente VdA* (since, according to her opinion, in the article Aldo also talks about the activity of these two NGO). Moreover, she leaves a comment explaining her point of view. By adding opinions to Maria's comment, other table participants can discuss the opportunity of having such objects of discourse in the shared view.

The *Our Planet* table implements an *authored collaboration policy* (see Section 3.4), enabling Aldo (who is the author of the resource in focus) to make the final decision about property values. Thus, on the basis of the opinions expressed by table participants, Aldo decides to stop the discussion and to delete the two annotations added by Maria. He writes a final comment to explain his decision.

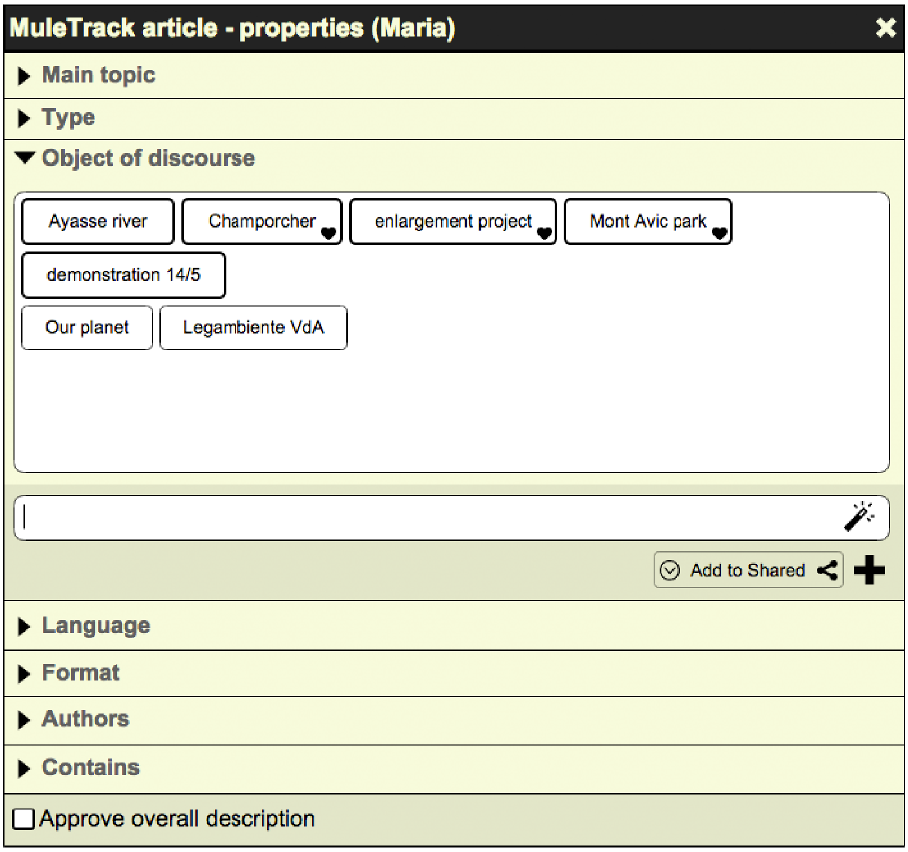

Fig. 5. SemT++ UI displaying semantic annotation of a table resource after some discussion among users

The two annotations, deleted from the shared view, remain available in Maria's personal view, so that she will be able to use them to select table objects in the future. Maria's final view on Aldo's article is shown in Fig. 5: Bold face boxes indicate annotations belonging to the shared view; a small heart means that Maria likes the annotations (i.e., hearts mark shared annotation also belonging to Maria's personal view); thin face boxes represent "private" annotations, belonging only to Maria's personal view.

4.2 Semantic Model Enabling Personal Views

As we mentioned in Section 3.1, the TO Semantic Knowledge Manager is in charge of managing the semantic descriptions of table objects (stored in the TO Semantic KB and based on the Table Ontology). In order to handle personal and shared views, following the model described above, we implemented a submodule, called *View Man-*

ager, having the role of handling views. Fig. 6 shows the new internal architecture of the TO Semantic Knowledge Manager.

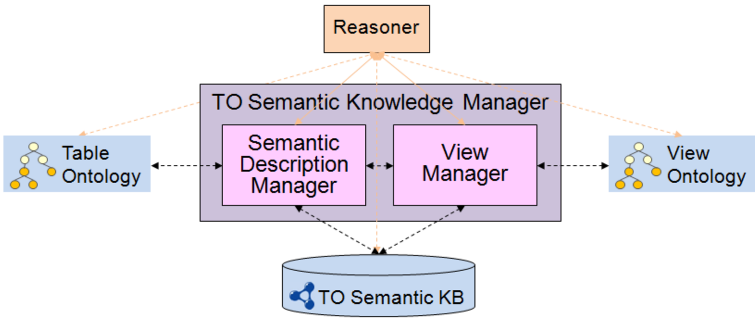

Fig. 6. Internal architecture of the TO Semantic Knowledge Manager of SemT++

The View Manager is endowed with specific knowledge represented by a *View Ontology*, according to which:

- A *context* (an instance of the *Context* class) represents a context in which a given set of assertions holds.
- A *view* (an instance of the *View* class) is a particular type of context (*View* is a subclass of *Context*).
- Assertion sets (instances of the *AssertionSet* class) are linked (by the *holds_in* relation) to contexts (and views).

An assertion set is represented by a named-graph, grouping assertions in the *TO Semantic KB triplestore* (the IRI of the *AssertionSet* instance is the name of the corresponding named-graph).

When a new table is created, the View Manager creates the following entities, stored in the TO Semantic KB, and depicted in Fig. 7:

- A new *Context* instance (e.g., *infoObjectContext*), representing the "table context", where assertions that are valid in all views (e.g., assertions made by the system) hold;
- A new *AssertionSet* instance (e.g., *infoObjectAssertionSet*) linked to *infoObjectContext* by the *holds_in* relation;
- A new *View* instance (*sharedView*), representing the shared view on that table;
- A new *AssertionSet* instance (*sharedAssertionSet*) linked to *sharedView* by the *holds_in* relation;
- A relation (*situated_in*) between *sharedView* and *infoObjectContext*, which enables the shared view to inherit all the assertions holding in the table context; in this way, for example, assertions made by the system automatically hold in the shared view.

Moreover, for each new table participant, the View Manager creates:

- A new instance (*tp*) of the *TableParticipant* class;

- A new *View* instance (*tpView*), representing *tp personal view* on the table, linked to *tp* by the *has_view_owner* relation;
- A relation (*situated_in*) between *tpView* and *infoObjectContext*, which enables the personal view to inherit all the assertions in the table context, so that, for example, assertions made by the system automatically hold in all personal views.
- A new *AssertionSet* instance (*tpAssertionSet*) linked to *tpView* by the *holds_in* relation;
- A new *AssertionSet* instance (*tpAuthorAssertionSet*) linked to *tp* by the *asserted_by* relation.

The system, on the basis of both the View Ontology and the Table Ontology, guarantees that each view (i.e., the shared one and all personal views) is *consistent*, thus enabling the Reasoner to run on each view, in order to make inferences, such as – for instance – those supporting the suggestion of candidate values (see Section 3.3).[3]

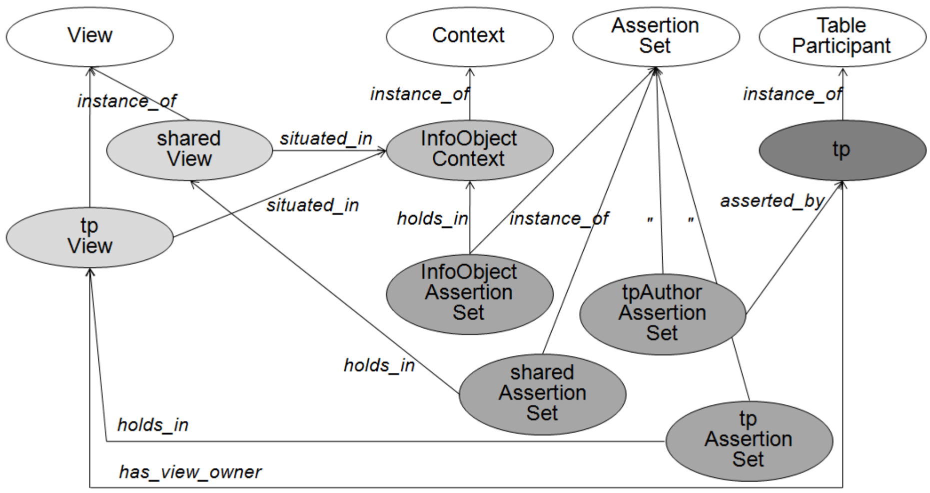

Fig. 7. The (simplified) semantic model for handling shared and personal views on a table in SemT++

The described model enables SemT++ to support the usage scenario sketched in Section 4.1. Since the *Our Planet* table had been created months before, the model depicted in Fig. 7 is already up. The following schema summarizes what happens for each step in the scenario: The left column represents the user actions (triggering events), the right column the consequent system action (in this column, the arrow represents a cause-effect relation):

---

[3] From the implementation point of view, this requires that all the assertions in the triplestore are loaded in the OWL knowledge base (together with the ontology).

Aldo clicks on the "save& update" button	The properties automatically set by the system are added to the *infoObjectAssertionSet* ↓ The TO Semantic Knowledge Manager runs the Reasoner and calculates candidates (thanks to the *DOLCE* : *part* relation holding between the image included in the article and the article itself, as well as between the link to the resolution included in the article and the article itself)
Aldo selects some candidate values and adds some new ones, deciding to add all of them to the shared view	The corresponding assertions are added to the *sharedAssertionSet*, to *AldoAssertionSet*, and to *AldoAuthorAssertionSet*
Maria looks at the author of an annotation	The TO Manager gets from the TO Semantic Knowledge Manager the reference to Aldo, retrieved by the View Manager from the *asserted_by property* linking *AldoAuthorAssertionSet* to its author (Aldo)
Maria looks at users who agree with an annotation (i.e., table participants who *like* it)	The TO Manager gets from the TO Semantic Knowledge Manager the reference to Aldo, retrieved by the View Manager from: (i) The *has_view_owner* property linking *AldoView* (i.e., Aldo's personal view) to Aldo; (ii) The *holds_in* property linking Aldo's personal view to *AldoAssertionSet*
Maria marks as *liked* some values	The corresponding assertions are added to *MariaAssertionSet*
Maria adds a couple of new objects of discourse to the shared view	The corresponding assertions are added to the *sharedAssertionSet*, to *MariaAssertionSet*, and to *MariaAuthorAssertionSet*
Aldo decides to delete Maria's annotations from the shared view	The corresponding assertions are deleted from the *sharedAssertionSet* (they remain available in *MariaAssertionSet* and in *MariaAuthorAssertionSet*)

In order to display the properties of Aldo's article (depicted in Fig. 5), the TO Manager gets from the TO Semantic Knowledge Manager all the information needed to display bold face boxes (for shared annotations), small hearts (for shared annotations also belonging to Maria's personal view), and thin face boxes (for personal annotations).

4.3 Future Evaluation Plan

Although the proposed approach is grounded in the results of a qualitative user study (described in detail in [25]), we plan to conduct a comparative evaluation, aimed at verifying the benefits of the availability of a personal point of view on resource annotations, in the context of collaborative workspaces. The evaluation will be carried out with the same methodology as the preliminary user study, in order to be able to compare the results. Therefore, three groups of users will be asked to collaborative annotate a few resources in a pre-defined scenario. Each group will repeat the experience twice, experimenting with different policies (*supervised*, *authored* and *consensual*). While in the initial study users performed the task with Google Documents, in the conclusive evaluation they will have the chance to use the SemT++ environment with the shared/personal views enhancement.

The goal of the evaluation is to determine to which degree limitations and difficulties perceived in the preliminary test are successfully overcome by our approach to collaborative annotation, and in particular by the availability of personal and shared views. This goal will be achieved by providing participants with a questionnaire they will have to fill in after performing the assigned tasks. The research questions we will address by means of the evaluation are listed below; the quality of the experience is qualitatively measured by the following parameters: Interest, engagement, usefulness, and difficulty. With respect to the initial scenario, where personal views were not available, the evaluation aims at answering the following *research questions*:

- Is the *experience* of collaborative annotation improved?
- Are the participants better satisfied with the *final annotations*?
- Are the participants better satisfied with the *collaboration and communication* within the group?

5 Conclusions and Future Work

In this paper we presented a model for handling both shared and personal views on Web resources. The presented approach is part of the SemT++ project, aimed at providing users with a collaborative environment for the management of digital resources in collaborative thematic workspaces. In the paper we described how personal views are handled in SemT++ as structured sets of semantic annotations, represented by semantic assertions grouped into named-graphs within a triplestore knowledge base, and supported by an ontology-based representation, modeling contexts, views, etc.

The approach described in this paper can be exploited to support other improvements of the user experience. For example, as far as users agree on making their personal views visible to other people, such views could represent an interesting source of knowledge about users interpretation of the annotated resources. Moreover, since the reasoner provides an explanation for the inconsistencies it detects, we will study the possibility of exploiting this information for interactively supporting users in solving such inconsistencies. Finally, we are studying the impact of making tables public, together with their shared and personal views: This is an interesting perspective that deserves a deeper analysis.

References

1. Barreau, D.K., Nardi, B.: Finding and reminding: File organization from the desktop. ACM SIGCHI Bulletin 27 (3), 39-43 (1995)

2. Goy, A., Petrone, G., Segnan, M.: Oh, no! Not Another Web-based Desktop!. In: 2nd Int. Conf. on Advanced Collaborative Networks, Systems and Applications, pp. 29–34. XPS Press, Wilmington, DE (2012)
3. Goy, A., Petrone, G., Segnan, M.: A Cloud-Based Environment for Collaborative Resources Management. Int. J. Cloud Applications and Computing 4(4), 7–31 (2014)
4. Kaptelinin, V., Czerwinski, M. (eds.): Beyond the Desktop Metaphor. MIT Press, Cambridge, MA (2007)
5. Bardram, J. E.: From Desktop Task Management to Ubiquitous Activity-Based Computing. In: Kaptelinin, V., Czerwinski, M. (eds.) Beyond the Desktop Metaphor, pp.223-260. MIT Press, Cambridge, MA (2007)
6. Voida, S., Mynatt, E. D., Edwards, W. K.: Re-framing the Desktop Interface Around the Activities of Knowledge Work. In: UIST'08, pp. 211-220. ACM Press, New York, NY (2008)
7. Geyer, W., Vogel, J., Cheng, L., Muller M. J.: Supporting Activity-Centric Collaboration through Peer-to-Peer Shared Objects. In: Group'03, pp.115–124. ACM Press, New York, NY (2003)
8. Muller, M. J., Geyer, W., Brownholtz, B., Wilcox, E., Millen, D. R.: One-Hundred Days in an Activity-Centric Collaboration Environment based on Shared Objects. In: CHI'04, pp. 375–382. ACM Press, New York, NY (2004)
9. Karger, D.R.: Haystack: Per-User Information Environments Based on Semistructured Data. In: Kaptelinin, V., Czerwinski, M. (eds.) Beyond the Desktop Metaphor, pp. 49–100. MIT Press, Cambridge, MA (2007)
10. Breslin, J. G., Passant, A., Decker, S.: The Social Semantic Web. Springer, Heidelberg (2009)
11. Abel, F., Henze, N., Krause, D., Kriesell, M.: Semantic Enhancement of Social Tagging Systems. In: Devedžić, V., Gašević, D. (eds.) Web 2.0 & Semantic Web, pp. 25–56. Springer, Heidelberg (2010)
12. Kim H., Breslin J.G., Decker S., Choi J., Kim H.: Personal Knowledge Management for knowledge workers using social semantic technologies. Int. J. of Intelligent Information and Database Systems 3(1), 28–43 (2009)
13. Sauermann, L., Bernardi, A., Dengel, A.: Overview and Outlook on the Semantic Desktop. In: 1st Ws on The Semantic Desktop at ISWC 2005, vol. 175. CEUR-WS (2005)
14. Drăgan, L., Delbru, R., Groza, T., Handschuh, S., Decker, S.: Linking Semantic Desktop Data to the Web of Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) The Semantic Web – ISWC 2011. LNCS, vol. 7032, pp 33–48. Springer, Heidelberg (2011)
15. Bontcheva, K., Cunningham, H., Roberts, I., Tablan, V.: Web-based collaborative corpus annotation: Requirements and a framework implementation. In Witte, R., Cunningham, H., Patrick, J., Beisswanger, E., Buyko, E., Hahn, U., Verspoor, K., Coden, A. R. (eds.) LREC 2010 workshop on New Challenges for NLP Frameworks, pp. 20–27 (2010)
16. Chamberlain, J., Poesio, M., Kruschwitz, U.: Phrase detectives - a Web-based collaborative annotation game. In: International Conference on Semantic Systems (I-Semantics), pp. 42–49. ACM Press, New York, NY (2008)
17. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. J. Web Semantics 4(1), 14–28 (2006)
18. Corcho, O.: Ontology based document annotation: trends and open research problems. Int. J. of Metadata, Semantics and Ontologies, 1(1), 47–57 (2006)

19. Torres, D., Skaf-Molli, H., Díaz, A., Molli, P.: Supporting Personal Semantic Annotations in P2P Semantic Wikis. In: Bhowmick, S. S., Küng, J., Wagner, R. (eds.) Database and Expert Systems Applications – DEXA 2009. LNCS, vol. 5690, pp 317–331. Springer, Heidelberg (2009)
20. Bateman, S., Brooks, C., Mccalla, G., Brusilovsky, P.: Applying collaborative tagging to e-learning. In: WWW 2007 workshop on Tagging and Metadata for Social Information Organization (2007)
21. Noy, N. F., Chugh, A., Alani, H.: The CKC Challenge: Exploring Tools for Collaborative Knowledge Construction. Intelligent Systems 23(1), 64–68 (2008)
22. Ardissono, L., Bosio, G., Goy, A., Petrone, G.: Context-Aware Notification Management in an Integrated Collaborative Environment. In: UMAP 2009 workshop on Adaptation and Personalization for Web2.0, pp. 23–39. CEUR (2010)
23. Goy, A., Magro, D., Petrone, M., Rovera, C., Segnan, M.: A Semantic Framework to Enrich Collaborative Semantic Tables with Domain Knowledge. In: IC3K 2015 – Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management – Vol. 3: KMIS, pp. 371–381. SciTePress (2015)
24. Goy, A., Magro, D., Petrone, G., Segnan, M.: Semantic representation of information objects for digital resources management. Intelligenza Artificiale, 8(2), 145–161 (2014)
25. Goy, A., Magro, D., Petrone, G., Picardi, C., Segnan, M.: Ontology-driven collaborative annotation in shared workspaces. Future Generation Computer Systems, Special Issue on Semantic Technologies for Collaborative Web, 54, 435–449 (2016)
26. Magro, D., Goy, A.: A Core Reference Ontology for the Customer Relationship Domain. Applied Ontology 7(1), 1–48 (2012)
27. Borgo, S., Masolo, C.: Foundational choices in DOLCE. In: Staab, S., Studer, R. (eds) Handbook on Ontologies, Second Edition, pp. 361–381. Springer, Heidelberg (2009)
28. Gangemi, A., Borgo, S., Catenacci, C., Lehmann, J.: Task Taxonomies for Knowledge Content. Metokis Deliverable D07 (2005)
29. Goy, A., Magro, D., Petrone, G., & Segnan, M.: Collaborative semantic tables. In: Molli, P., Breslin, J., Vidal, M. E. (eds.) SWCS 2014 - Semantic Web collaborative spaces, vol. 1275. CEUR (2014)