

Collaborative Semantic Tables

Anna Goy, Diego Magro, Giovanna Petrone, Marino Segnan

Dipartimento di Informatica, Università di Torino
{goy, magro, giovanna, marino}@di.unito.it

Abstract. The scenario defined by current Web architectures and paradigms (such as Cloud Computing), poses challenges and opportunities to users. On the one hand, they have to manage huge amounts of digital resources handled by different applications in spite of their possibly related content; on the other hand, they are enabled to share knowledge and participate to content creation. The interaction of these two aspects provided a great impulse to collaborative resource management: in this paper we present T++, an environment that exploits semantic knowledge about digital resources in order to face these challenges, by providing an integrated and smart management of heterogeneous *information objects*.

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

1 Introduction

The current ICT scenario, and in particular Web architectures and paradigms, are posing new challenges to Personal Information Management [4]. Many aspects of human-computer interaction have been influenced; in this paper, we focus on the most relevant for our approach: (a) Users have to deal with a huge number of heterogeneous resources, stored in different places, encoded in different formats, handled by different applications as belonging to different types (images, emails, bookmarks, documents, ...), despite their possibly related content. (b) Web 2.0 and, more recently, Cloud Computing, and 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 resources and knowledge sharing. The interaction of these two aspects provided a great impulse to user collaboration in managing shared resources.

In this paper we present *Semantic Table Plus Plus* (Sem T++), an environment aimed at supporting users in collaborative resource management. In particular, in Sem T++, two types of semantic knowledge are modeled: (1) knowledge about *information objects*, i.e., information resources as such, and (2) knowledge about their *content*. The goal of this paper is to show how collaborative annotations, based on the formal semantic representation of these two types of knowledge, support users in the organization, retrieval and usage of shared digital resources.

The rest of the paper is organized as follows: in Section 2 we discuss the motivations of our approach and the main related work; in Section 3, we briefly present T++, and in Section 4 we describe its semantic enhancement, by presenting its architecture and the semantic model underlying it, and showing how the system supports users in collaboratively handling semantic descriptions of digital resources. 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 [10], a previous paper introducing T++. As far as the approach presented in this paper is concerned, one of the most relevant research areas is represented by Kaptelinin and Czerwinski's book *Beyond the Desktop Metaphor* [12], which contains an interesting presentation of the problems of the so-called *desktop metaphor*, and of the approaches trying to replace it. Within this framework, one of the most interesting models discussed in the mentioned book is Haystack [13], a flexible and personalized system enabling users to define and manage workspaces referred to specific tasks. Another interesting family of approaches are those grounded into Activity-Based Computing (e.g., [3], [19]), where the main concept around which the interaction is built is *user activity*. Also [9] and [16] propose a system supporting lightweight informal interactions, as well as larger and more structured collaborative projects, by relying on activity-based workspaces, handling collections of heterogeneous resources.

Strategies exploited to organize resources have been studied within the field of Personal Information Management; in particular, multi-facets classification of resources has been taken into account: resources can be tagged with meta-data representing different aspects (*facets*), leading to the creation of *folksonomies*, bottom-up classification models collaboratively and incrementally build by users [6]. Interesting improvements of such tagging systems have been designed by endowing them with semantic capabilities (e.g., [1]), in particular in the perspective of knowledge workers [14].

Another important research thread, aiming at coupling desktop-based user interfaces and Semantic Web, is represented by the so-called *Semantic Desktop* [17]. In particular, the NEPOMUK project (nepomuk.semanticdesktop.org) defined an open-source framework, based on a set of ontologies, for implementing semantic desktops, focusing on the integration of existing applications, in order to support collaboration among knowledge workers. Finally, [7] presents an interesting model connecting the Semantic Desktop to the Web of Data.

3 The Starting Point: Table Plus Plus

The *Table Plus Plus* (T++) project, described in [10] and [11], proposes an interaction model supporting users in collaboratively handling digital resources, based on the

metaphor of *tables*, populated by *objects*. T++ is characterized by 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, to plan a journey.

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 [2] for a more detailed discussion of notification filtering).

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" [10, p. 32]. The most peculiar aspect of T++ tables is the collaborative management of table resources: table participants, in fact, can (a) modify objects, delete them, or add new ones; (b) invite people to "sit at the table" (i.e., to become a table participant); (c) define meta-data, such as comments and annotations (see below).

Heterogeneous objects management 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 visibility labels, comments, and tags).

Within the T++ project, we developed a proof-of-concept prototype, consisting in a cloud application (a Java Web App deployed on the Google App Engine) accessible through a Web browser. The current version exploits Dropbox and Google Drive API to store files corresponding to table objects and Google Mail to handle email conversations. We are investigating the availability of open API provided by other common file sharing and online editing tools in order to improve interoperability and to enable users to configure the preferred tools to be exploited for object sharing and editing.

In [11] we also reported 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 and using T++. The results showed that, with T++, performing the required tasks is faster and user satisfaction is higher.

4 The Semantic Enhancement of T++

4.1 Architecture

On the basis of T++, we designed an enhanced version, *Sem T++*, in which semantic knowledge plays a major role in supporting resource management on tables. Fig. 1 shows the relevant components of *Sem T++* architecture:

- *Semantic Knowledge Manager*: it manages the semantic descriptions of table objects (stored in the Semantic KB) and invokes the Reasoner, when required. Moreover, it handles the connection with GeoNames (www.geonames.org), as described in Section 4.2. In the current prototype, it uses the OWL API library (owlapi.sourceforge.net) to interact with the ontologies and the Semantic KB, and the GeoNames Search Web Service (www.geonames.org/export) to query GeoNames. It interacts with the following knowledge bases (written in OWL: www.w3.org/TR/owl-features):
 - *Table Ontology*: it represents the (static) system semantic knowledge concerning *information objects*.
 - *Geographic Ontology*: it represents the (static) system semantic knowledge concerning geographic entities and features.
 - *Semantic KB*: it contains all the facts about the individuals involved in the semantic representation of table objects.
- *Reasoner*: it provides the system with "new" object features which can be exploited to support the user in table object management. The current proof-of-concept prototype uses Fact++ (owl.cs.manchester.ac.uk/tools/fact).
- *Object Manager*: it manages the "used objects" (i.e., objects on a table or included in objects on a table) and the references to elements used in the interaction with the user (e.g., available object types and object properties, corresponding to Table Ontology classes and relations). The Object Manager plays a mediation role between the Table Manager (and thus, indirectly, the UI) and the components which represents the system "intelligence", i.e. the Semantic Knowledge Manager and the Smart Object Analyzer.
- *Smart Object Analyzer*: it is a service that provides the Object Manager with the analysis of table objects, in order to discover information about them; for example, it looks for parts included in the analyzed object (e.g., images, links, etc.). In the current prototype, it exploits a Python Parser Service, able to analyze HTML documents¹.

¹ In the current prototype we focused on HTML documents since they are very common, quite easy to parse, and their semantic characterization introduces interesting aspects (e.g., the need of distinguishing between *information content* and *content of the HTML file*).

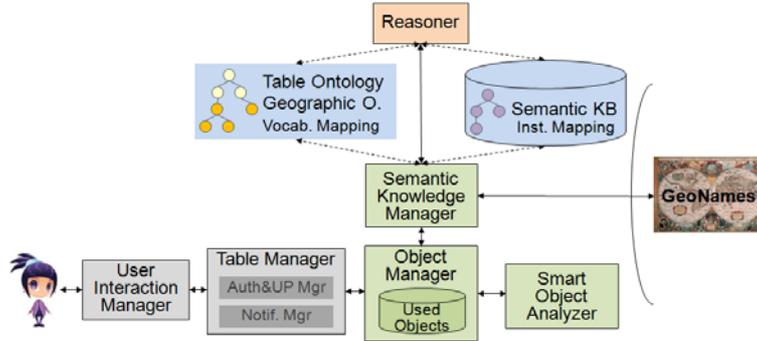


Fig. 1. Sem T++ architecture (relevant components).

4.2 Semantic Model

The core of our proposal is the Table Ontology, which models knowledge about information resources. It is grounded in the Knowledge Module of O-CREAM-v2 [15], 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) [5] and some other ontologies extending it, among which the Ontology of Information Objects (OIO) [8]. The Table Ontology enables us to describe digital resources lying on tables as *information objects*, with properties and relations; for instance: a table object can have parts (e.g., images within a document), which are in turn information objects; it can be written in English; it can be stored in a PDF file, or it can be an 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 Monte Bianco, probably the document talks about Monte Bianco.

The most relevant class in the Table Ontology is *InformationElement*: all table objects are instances of it. Moreover, we introduced some specific subclasses (e.g., *Document*, *Image*, *Video*, *Audio*, *EmailThread*, etc.), to provide a more precise characterization of the different types of objects that can lay on a table. In order to characterize such classes, we relied on: (1) a language taxonomy defined in O-CREAM-v2, representing natural, formal, computer, visual languages; (b) a set of properties (some of them inherited from O-CREAM-v2). 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* : $part(x, y, t)$ – to represent relations such as the one between a document and an image or a hyperlink included in it.
- $specifiedIn(x, y, t)$ – to represent relations such as the one between a document and the language it is written in (e.g., Italian).
- $hasTopic(x, y, t)$ – to represent the relation between an information element (e.g., a document) and its main topic.

- *hasObjectOfDiscourse*(x, y, t) – to represent what a resource "talks about" (it is a subproperty of *OIO : about*).
- *identifies*(x, y, t) – to represent, for instance, the relation between a hyperlink and the resource it points to.

An important fragment of the proposed semantic model refers to particular properties which model *candidate relationships*. The idea is that the system, endowed with suitable axioms, can infer *candidate* features, mainly from included objects; for example, the Reasoner can infer that Monte Bianco is a *candidate* object of discourse of a document from the fact that the document itself includes an image of it. 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 Monte Bianco is actually an object of discourse of the document (*doc*), then a new relation *hasObjectOfDiscourse*(*doc*, *MonteBianco*, *t*) is added to the knowledge base. Analogous axioms are available for the *hasTopic* relation, to support the inference of *hasCandidateTopic* relationships.

Besides knowledge about information objects, all tables in Sem T++ are equipped with *geographic knowledge*. We decided to model geographic knowledge for two reasons: (a) Together with *time* (currently not modeled in Sem T++), *space* represents a cross-domain feature, which is – at least partially – represented by geographic knowledge². (b) Geographic knowledge represents for us a testbed, i.e., an example of knowledge that characterizes the "content" of table objects: in the future, tables could be equipped with semantic knowledge about specific domains, by relying on the same mechanism we used for geography.

In order to provide tables with geographic knowledge, we exploited *GeoNames*, a huge, open geographical database containing over 10 million geographical entities. Moreover, we equipped the system with a Geographic Ontology and a *Vocabulary Mapping*, defining correspondences between Geography Ontology classes/properties and GeoNames *feature classes/codes*. GeoNames entities, in fact, are categorized into *feature classes* (e.g., *A*, corresponding to the generic concept of *country* or *region*) and further subcategorized into more specific *feature codes* (e.g., *A.ADMI*, corresponding to the concept of *primary administrative division of a country*). For each topic³ mentioned on a table, the Semantic Knowledge Manager searches for corresponding GeoNames entities. If one or more results are found, the system currently asks the user to select the correct one, if any⁴; then a new individual is created in the Semantic KB, as instance of the Geographic Ontology class identified through GeoNames feature class/code (thanks to the *Vocabulary Mapping*). For example, an instance of the class *Mountain* is created for the topic *Mont Avic* (a mountain in Val d'Aosta, a northwestern Italian region). Moreover, a new mapping between such an instance and the corresponding GeoNames entity is created in the Semantic KB (see

² *Geographic knowledge* here means commonsense competence such as the ability to georeference places, or knowing that Monte Bianco is a mountain, and not scientific knowledge.

³ In the current prototype, we consider only topics, although also objects of discourse could be taken into account.

⁴ Mechanisms to support a (partially) automatic disambiguation are under study.

Instances Mapping in Fig. 1), thus making all information available in GeoNames (e.g., its location on a map) also available on Sem T++ tables.

The described model enables table participants to specify and combine different selection parameters in order to find objects on a table. For example, to get all email threads talking about *Monte Bianco trekking* (i.e., having it as main topic), the user can specify the following parameters: *topics={Monte_Bianco_trekking}*, *types={emailThread}*. Each parameter value corresponds to a user selection; object types are references to Table Ontology classes (*emailThread* in this example); values for other properties, e.g. *Monte_Bianco_trekking*, are references to individuals in the Semantic KB. Moreover, the user could provide more general queries, such as asking for all resources talking about mountains, thanks to the facts that topics (e.g., *Monte-Bianco*) are represented as instances of classes in the Geographic Ontology (e.g., *Mountain*). User queries are handled by the Semantic Knowledge Manager, which accesses the Semantic KB and invokes the Reasoner, in order to provide the objects matching the parameters.

4.3 Collaboratively Handling Semantic Descriptions of Digital Resources

One of the most challenging aspects of the presented semantic model is the creation and update of semantic representation of table objects. In fact, when a new object is created, or when an existing one is modified (e.g., when a table participant includes a new image or link in it), the corresponding semantic representation must be created or updated. In the following we will see how the collaborative management of semantic descriptions of table objects represents a step to face such a challenge, and, in particular, how this makes T++ tables actual *shared semantic spaces*.

Consider the new object case (the update case works in an analogous way): table participants can create new objects from scratch (e.g., when they start writing a new document), or they can add an existing resource (e.g., a bookmark pointing to a Web site) to the table. In both cases, the system builds a new semantic representation in four steps, in which different components play their role:

1. *Smart Object Analyzer*: some properties (e.g., mereological composition, types of the parts, formats) are automatically determined by the Smart Object Analyzer.
2. *Semantic Knowledge Manager/Reasoner*: other properties (e.g., candidate topics – see Section 4.2) are inferred by the Reasoner.
3. *User*: some of the inferred properties need a "user confirmation" (e.g., candidate topics); moreover, users can add properties (typically objects of discourse).
4. *Semantic Knowledge Manager/GeoNames*: the selected topic is linked to the corresponding geographic entity (through GeoNames), if any.

In order to describe the sketched process into more details, consider a usage scenario in which Maria participates in a table concerning the activities of a small ONG for environment safeguard, *Our Planet*, together with some other volunteers. Maria has to write an article for an online local newspaper, discussing the situation of a local old mule track in Champorcher (a small municipality in Val d'Aosta). Maria creates a new table object (an HTML document), writes some text in it, adds a picture of the envi-

ronments and a hyperlink to a resolution by the Municipality of Champorcher concerning an enlargement project for the mule track. When Maria clicks the "save&update" button, the creation of the object semantic representation is triggered.

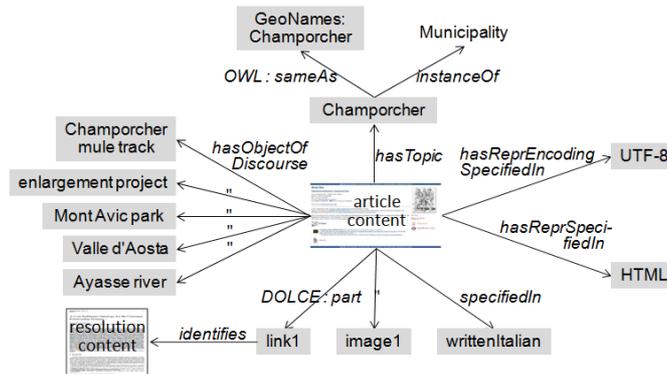


Fig. 2. Semantic representation of Maria's article.

In step 1 the Smart Object Analyzer discovers that the object is encoded in HTML/UTF-8, written in Italian, and that it contains two parts: an image and a hyperlink. In step 2 the Reasoner infers *candidate topics* and *candidate objects of discourse*. In particular, Maria can: select suggested candidates; select items within the list of values already present on the table and associated to other objects; add new topics/objects of discourse from scratch (step 3). The final semantic representation of her article could be the one shown in Fig.2. In step 4, for the new topic (*Champorcher*) a corresponding GeoNames entity is searched and found, thus a new instance (linked to GeoNames through a new entry in *Instances Mapping*) is created in the Semantic KB.

Fig. 3 (left-hand side) shows a mockup of Sem T++ user interface; on the left, table participants can access the main functionalities available on the table in focus, i.e. access to table objects and three basic services: *Chat* (for synchronous communication among table participants), *Blackboard* (for asynchronous communication), and shared *Calendar*. Other functionalities, e.g., settings, are available through the "menu" in the left-bottom corner. On the right, there is the list of tables and the list of participants at the table in focus (which is highlighted): participants not currently sitting at the table are in light grey. The central area shows the panel triggered by a click on the "objects" label on the left, listing table objects. Objects can be visualized by clicking on their names. Moreover, for each object, the visibility label is reported ("pub" means *public*, i.e., visible to all table participants; objects could also be *private*, i.e., visible only to their creator), together with an access to comments (if available), to the semantic description of the object ("sem prop" column), and a delete button.

When Mario, another volunteer, sits at the *Our Planet* table, he is notified about the new object created by Maria (new/modified object names are highlighted). Mario looks at its semantic description, by clicking on the button in the "sem prop" column: a new panel, shown in Fig. 3 (right-hand side), is displayed. From this panel, it is

possible to change the object type (an example scenario is described below) and to add/delete semantic properties, with the exception of the "contains" property, which refers to *DOLCE* : *part(x, y, t)*: object parts (e.g., images included in a document) can be modified only by editing the object.

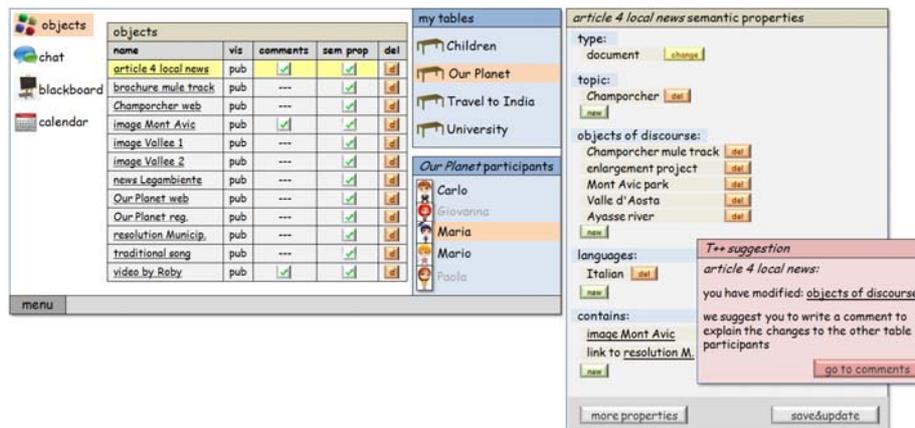


Fig. 3. Sem T++ user interface (mockup).

Mario finds that one of the objects of discourse (*Ayasse river*), up to him, is misleading: he thus decides to remove it and to add a new object of discourse (*Ourty alpine pasture*). When Mario clicks the "save&update" button, the semantic description is updated. Moreover, the table suggests Mario to write a comment in order to explain his decision to modify it (see Fig. 4, right-hand side); such a comment is attached to the modified object, but handled by the table, and is thus immediately accessible to other participants, as they sit at the *Our Planet* table.

A very similar case occurs when a user adds to a table an object by classifying it in a way which is then modified by another table user. For example, Maria could have added an image which, actually, is the scan of a document, and Mario could decide to change the object type from *Image* to *Document*. Again, Sem T++ suggests him to use the table-level annotation mechanism in order to share the reasons of the change.

This very simplified usage scenario shows the support provided by Sem T++ to the collaborative handling of semantic descriptions of table objects. Sem T++ has been designed having in mind a very "democratic" model of collaboration, in which people sitting at a table have the same privileges and thus all of them can modify objects, and in particular their semantic representations. However, Sem T++ also provides effective and easy-to-use table-level annotation mechanisms which represent a significant support to such a collaborative activity, together with the other standard mechanisms enabling discussion and communication among table participants, i.e., the *Blackboard* and the *Chat* (see [11] for details).

The coordination of collaborative activities taking place on Sem T++ tables deserves a final comment. As far as possible concurrent resource editing (e.g., document modification) is concerned, Sem T++ relies on existing applications, which typically

handle issues related to collaborative editing (such as Google Drive). As regards object management (adding, deleting, modifying table objects) and collaborative semantic annotation, the default configuration is the already mentioned simple policy allowing every table participant to freely modify, add, or remove all objects and annotations. Obviously, there are specific contexts (e.g., when tables are used to handle business activities, such as project management, for example) requiring more structured policies, modeling different user roles, privileges and specific workflows. We are working on the definition of mechanisms enabling users to configure such policies on each table.

5 Conclusions and Future Work

In this paper we presented Sem T++, an environment supporting users in collaborative resource management, by showing how formal semantic knowledge about *information objects* and their content can support an integrated, user-friendly management of heterogeneous shared resources. Sem T++ is a work in progress and many aspects, concerning the collaboration model and the user interface have not been discussed here. In particular, we are designing configuration mechanisms enabling users to define specific policies to coordinate collaborative activities, and specifically collaborative annotation management [18], on each table. Moreover, we are going to complete the integration of semantic modules into T++ prototype, in order to plan a user evaluation aimed at testing how the semantic model presented in this paper supports users in organizing and retrieving information objects on tables, but also the effort required to create and update semantic descriptions of table objects.

References

1. 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)
2. 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-WS (2010)
3. 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)
4. Barreau, D.K., Nardi, B.: Finding and reminding: File organization from the desktop. *ACM SIGCHI Bulletin* 27 (3), 39-43 (1995)
5. 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)
6. Breslin, J. G., Passant, A., Decker, S.: *The Social Semantic Web*. Springer, Heidelberg (2009)
7. 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)
8. Gangemi, A., Borgo, S., Catenacci, C., Lehmann, J.: *Task Taxonomies for Knowledge Content*. Metokis Deliverable D07 (2005)
 9. 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)
 10. 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)
 11. Goy, A., Magro, D., Petrone, G., Segnan, M.: *A Cloud-Based Environment for Collaborative Resources Management*. *Int. J. Cloud Applications and Computing* 4(4), in press (2014)
 12. Kaptelinin, V., Czerwinski, M. (eds.) *Beyond the Desktop Metaphor*. MIT Press, Cambridge, MA (2007)
 13. 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)
 14. 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)
 15. Magro, D., Goy, A.: *A Core Reference Ontology for the Customer Relationship Domain*. *Applied Ontology* 7(1), 1–48 (2012)
 16. 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) 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)
 17. Sauer mann, 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)
 18. Su, A.Y., Yang, S.J., Hwang, W.Y., Zhang, J.: *A web 2.0-based collaborative annotation system for enhancing knowledge sharing in collaborative learning environments*. *Computers and Education* 55(2), 752–766 (2010)
 19. Volda, 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)