Semantic representation of information objects for digital resources management

Published version:

DOI:10.3233/IA-140070

(Article begins on next page)

Semantic representation of information objects for digital resources management

Anna Goy, Diego Magro, Giovanna Petrone and Marino Segnan
Dipartimento di Informatica, Università di Torino, Torino, Italy
E-mail: {goy,magro,giovanna,marino}@di.unito.it

Abstract. Current users of digital devices have to face the management of a huge amount of heterogeneous digital resources, the switch between activity contexts, and the interaction with many different applications and services. This situation leads to a very fragmented interaction experience, which poses a great cognitive overload for the users and risks to cause lack of efficiency and loss of information. Starting from the limitations of both traditional mechanisms for Personal Information Management (based on the notions of *files* and hierarchical *folders*) and new proposals (tagging and *folksonomies*), in this paper we present Semantic T++, a system supporting users in collaboratively handling digital resources, based on the notion of "tables" (thematic Web-based collaborative workspaces), populated by "objects" (shared digital resources). Semantic T++ exploits a formal semantic representation of such objects to support users in organizing, selecting and using them. Its core is represented by an ontology which models table objects as "information elements" having properties and relations mainly (but not only) related to their content. Reasoning techniques can be applied to infer knowledge useful to provide users with a flexible access to table objects, based on different criteria, which can be defined and combined by the user on the basis of her needs. In order to evaluate our model, we demonstrated its technical feasibility by developing a proof-of-concept prototype, and we showed its advantages in the access to personal and shared resources by discussing the results of a user test.

Keywords: Intelligent Web-User Interfaces, ontology-based access to digital resources, ontology-based user interaction, semantic technologies, Personal Information Management.

1. Introduction

Current User Interfaces (UI) for laptops and personal computers are mainly based on the *desktop metaphor* [27, 19], an interaction model whose application dates back to the early Eighties. The desktop metaphor handles documents, applications, and resources as items spatially organized on top of a (virtual) desktop, and provides a storage system (i.e., the *file system*) based on the metaphor of physical office shelves, where documents and applications are stored as *files*, organized in a hierarchical structure of *folders*. The desktop metaphor has been an effective way to fulfill the needs of individual users, interacting with stand-alone personal computers. However, although it is still widely used, the context of Human-Computer Interaction (HCI) has deeply changed.

The first "revolution" started in the Nineties and was represented by the WWW, which enabled users to "navigate" Web sites, to store bookmarks, to interact with each other through chat tools and forums, besides email. More or less ten years later, another big change impacted HCI, i.e., Web 2.0. The WWW became a network of highly interactive services, including many functionalities previously supported by stand-alone applications (from word processing, to home banking and Public Administration services). During the same years, wireless connections and mobile devices reached a widespread availability, and gradually became the most used way to access online services. This trend has been coupled, in more recent years, with the success of the Cloud Computing paradigm, which provided a great impulse to information sharing and online collaboration, together with new possibilities for data storage and applications running in the Cloud.

The summarized changes deeply modified some concepts which have a great influence on HCI. The concept of stand-alone application running on a personal computer has been replaced by online services, possibly running in the Cloud; the notion of *file* has been flanked by a variety of heterogeneous "objects", such as documents, emails, posts, events, bookmarks, streaming; the support to information sharing and online collaboration has become essential, as well as the support to ubiquitous access.

From the point of view of the proposal presented in this paper, the most relevant consequences of the sketched evolution are those that impact on *Personal Information Management* (PIM) [4, 17]: an increasing number of digital resources, encoded in different data formats, handled by different applications, and possibly stored in different places, are available, making contents scattered over different locations and tools, and raising the well-known *information fragmentation* problem [18] (see also [6, 17]). As Ravasio and Tscherter well summarize: "One cause that intensifies the cognitive load is the system-sided separation of the various information classes, such as bookmarks, emails, and files. Users think of their data as one single body of information. It is necessary to reunite all user-owned data at a single storage location [...] joining items that belong together from the user's point of view" [25, p. 275]. Furthermore, as Jones states: "We may find ourselves maintaining several separate, roughly comparable but inevitably inconsistent, organizational schemes for electronic documents, paper documents, email messages, and web references" [18, p. 8].

Moreover, the pervasive usage of electronic devices (e.g., tablets and smartphones), has increased the number of activities which are carried on by relying on ICT tools, with the consequence that users have to handle, and switch between, many separate *activity contexts*, which refer to different, but often linked, aspects belonging to work activities as well as private life management (from children care, to leisure planning).

Finally, *information fragmentation* has been coupled with another well-known problem, the *information overload:* available information is too much, and users need proper tools, enabling them to organize, filter, find, share, and use the huge amount of information they face every day. As far as this aspect is concerned, it is important to make a distinction between two aspects: the first one is *finding* what we need within the huge amount of information available, mainly on the Web; facing this issue is the goal of *searching* and *filtering* approaches and tools. The second aspect is "keeping found things found" [18, p. 4]: once found, in fact, information and resources have to be organized in a way that makes them accessible and usable. This is the core of PIM, and everyday experience tells us that it is far from being an achieved goal.

Taking on the challenge posed by the sketched scenario means designing a new interaction model and developing new tools supporting a unified representation of the heterogeneity described above, in order to provide a homogeneous management of contents and resources, independent from applications and formats, but able to effectively handle separate activity contexts.

The goal of this paper is to show how Artificial Intelligence, and in particular Knowledge Representation, can contribute to provide a solution: formal semantic representations, in fact, can be exploited to provide PIM tools with knowledge about *information objects*, i.e. information resources as such; this knowledge can support a flexible and smart access to personal and shared digital resources.

In order to achieve this goal, we have developed *Semantic T++*, i.e., a significative enhancement of the *Table Plus Plus* (T++) interaction model, described in [14] and [15]. *Semantic T++* aims at facing the specific needs raising within the PIM field as described in the sketched scenario, i.e., the need of a new flexible and effective way of organizing and accessing digital resources. In particular, Semantic T++ provides users with different classification and selection criteria, mainly based on content, but also on other resource characteristics, which can be combined on the basis of specific user needs, in order to provide an effective access to personal and shared digital resources.

The rest of the paper is organized as follows: Section 2 discusses the main related work; Section 3 briefly presents T++, states the goal of its semantic enhancement, sketches a simple usage scenario; then it describes the system architecture, the semantic model it relies on, and the user interaction supported, as far as resources selection is concerned; the section concludes by explaining in details how the system enables the presented usage scenario. Section 4 presents the evaluation of the approach, by describing a proof-of-concept prototype of the backend, and presenting results of a user test. Section 5 concludes the paper.

2. Related work

A good survey and discussion of the issues raised by UI based on the desktop metaphor, and of the approaches proposed to replace it with different interaction models, can be found in [20]. As far as this paper is concerned, one of the most relevant approaches presented in the book is Haystack [21]. Karger's approach is grounded into the consideration that individual users have different needs and preferences concerning how resources should be handled and which relations and attributes help retrieving and organizing them into coherent workspaces. In Haystack, heterogeneous resources (e.g., digital and physical documents, persons, tasks, etc.), uniformly identified by URIs (Uniform Resource Identifiers), can be annotated, linked to other items, retrieved, and viewed by users with a high flexibility degree. Another interesting approach, overviewed in the same book, is Activity-Based Computing (ABC), proposed by Bardram [3]. In the ABC paradigm the main concept modeling user interaction is *user activity*. As we will see, the notion of user activity plays a major role also in our proposal. However, in a perspective closer to Haystack, we claim that the resource organization model should not be based on a single principle: thus, the notion of user activity should be coupled with other principles, such as *semantic content*, in order to organize resources in a flexible and effective way (see Section 3).

As we already stated above, with respect to the problem of resource organization, the desktop metaphor mainly relies on the notion of *files*, grouped into hierarchically structured *folders*. Such an approach has become pervasive: email clients, bookmark managers, menus of almost every tool and application (e.g., Project Management tools, photo editing software, etc.) have applied the same principles. The first consequence is the increasing number of folders with (almost) the same names and related contents, handled by different applications [6]. Moreover, the classification of resources into a hierarchy of categories requires a significant cognitive effort for the user [22], and does not usually support multiple classification, which represents a strong limitation in many contexts.

Strategies users exploit to organize and find resources have been studied within the research field of Personal Information Management (see, for instance, [4], [28]). In particular, a strategy that can be exploited to support multiple classification is to attach meta-data, representing different aspects (*facets*), to items, thus enabling a *multi-facets* classification of resources, representing a user-centered view which can replace or flank the "machine-centered" view of resources as files [11]. This approach has been widely adopted on the Web in social tagging systems (see [1] for a summary survey), where resources are accessed through a *tag cloud.* Such social tagging systems lead to the creation of bottom-up classification models, called *folksonomies*, categorization structures collaboratively and incrementally built by users [8].

Tagging systems and folksonomies are often considered as a way to overcome folder hierarchy (taxonomy) rigidity [24]. However, recent studies, which have tried to compare the two models, show that also tagging systems have some limitations [9]. For instance, tags inherits some of the typical problems of natural language interpretation, such as ambiguity, synonymy, and polysemy [24]. Some interesting improvements of tagging systems have been realized by exploiting them in conjunction with semantic technologies; e.g., the GroupMe! system [1] enables users to define groups of content items they consider relevant for some topic, thus supporting a semantic-aware resources organization.

An important research thread which aims at bringing together desktop-based UI and the Semantic Web is represented by the so called *Semantic Desktop* [26]. Its goal is "to enable better organization of the personal information on our computers, by applying semantic technologies on the desktop" [12, p. 33]. The NEPOMUK project (nepomuk.semanticdesktop.org) defined an open-source framework for the Semantic Desktop, focusing on the integration of existing applications in order to support collaboration among knowledge workers. The Nepomuk Semantic Desktop is based on a set of ontologies, modeling the various aspects of the desktop itself; for example, the Personal Information Model (PIMO) defines concepts like *Person*, *Project*, *Event*, and *Task*; the Nepomuk Annotation Ontology (NAO) enables users to tag and rate resources, the Task Model Ontology (TMO) models tasks and to-dos.

A particularly relevant research thread has tried to enhance Personal Information/Knowledge Management tools by adding "intelligence" to them [5, 10]; for instance, support for intelligent search or automatic information classification has been provided by adding semantic knowledge, and thus reasoning capabilities, to Knowledge Management systems. As already mentioned, also tagging systems have been coupled with "intelligent" mechanisms, usually based on semantic resources such as ontologies, in order to better support users; e.g., [1]. However, all these approaches usually focus on *domain* knowledge, ignoring the fact that, in order to manage information resources, potentially belonging to very different domains, usually users also exploit specific knowledge about *information objects* as such. The semantic model presented in this paper is based on the idea of exploiting this particular type of semantic knowledge, which does not refer to any specific domain, but models in details *information resources*. Our proposal relies on a previously developed ontology, O-CREAM-v2 [23], 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) [7], and three other ontologies extending DOLCE [13], i.e., the ontology of Description and Situations (DnS), the Ontology of Information Objects (OIO), and the Ontology of Plans (OoP). In particular, for the approach presented in this paper, besides the DOLCE framework, only the OIO extension is relevant, and only O-CREAM-v2 Knowledge Module is used, which introduces the concept of *InformationElement* as a refinement of *OIO : InformationObject*, together with some other types of information elements and properties. In Section 3.5 we will provide an overview of the relevant classes and properties based on O-CREAM-v2 and used in Semantic T++.

3. Semantic T++

3.1. The previous version: Table Plus Plus (T++)

The Table Plus Plus (T++) project proposes an interaction model supporting users in collaboratively handling digital resources, based on the notion of *tables*, populated by *objects*, and relies on the following main concepts.

User activity. In T++, collaborative workspaces devoted to the management of user activities are represented by *tables*, i.e. "*thematic contexts*, helping the user to manage separate, coherent and structured workspaces, encompassing all types of activities (from personal to work-related ones)" [14, p. 32]. Users can define new tables, linking them to any type of activity, at the preferred granularity level (e.g., a table can be used to manage a work project, to handle children care, to plan a journey, to organize a fund-raising dinner, and so on).

Collaboration. An important aspect of T++ tables is that they are collaborative in nature, since they represent a shared view on resources (and people) [14]. However, tables also support the individual user's view by providing mechanisms for handling *private* resources. In particular, comments attached to table objects and objects themselves can be labeled as "private", i.e., visible only to their author; moreover, also tables can be private, thus being used as individual resource management tools.

Heterogeneous objects management. In T++ *objects* lying on tables can be resources of any type (documents, images, videos, to-do lists, bookmarks, email conversations, and so on), but they are managed in a uniform way. All table objects, in fact, are identified by a URI, and are considered as *content items*. Each object can be handled by a specific application (e.g., a document can be modified through Google Drive, a Web site can be displayed in a Web browser window, and so on), but the specific format of the file storing an object is transparent.

Workspace–level annotations. "Currently, annotations are coupled with a given document or application, usually with both. This implies that annotations are encapsulated within documents" [25, p. 285]. Providing a solution to this problem is one of the main goal of T++, which offers an *abstract view* over resources by implementing a table-level annotation mechanism, supporting different kinds of annotations, such as visibility labels, comments, and semantic tags (see Section 3.5).

Workspace awareness. In T++ workspace awareness is supported by three mechanisms: (a) Notification messages are filtered on the basis of the topic context represented by the active table. (b) 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*), so that the possible annoyance generated by messages coming from different contexts is strongly reduced. (c) Standard awareness techniques, such as icon highlighting, are used to notify users that something has occurred on the table (e.g., an object has been modified).

3.2. The semantic version: goal of Semantic T++

The main contribution of this paper is to describe and discuss a formal semantic representation of objects lying on T++ tables, which provides information about such objects and their relations, supporting users in the organization, retrieval and usage of digital resources. The core of our proposal is the Table Ontology, which is grounded in DOLCE, OIO, and O-CREAM-v2 (see Section 2). 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, or in Italian (or it can have a part written in English and a part written in Italian); it can be stored in a PDF file, or it can be an HTML page; it has an author and a content, which usually is about a main topic and a set of other entities (e.g., a document can tell the story of a village – its topic – and talk about mountains surrounding it, old paths used by shepherds, and so on). 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 is about Monte Bianco; if a Web page contains a link to a resource written in French, maybe also the Web page is in French.

Thanks to table objects semantic representation, and the inferences it enables, users are provided with a flexible access to digital resources, based on different criteria, which can be selected and combined by users themselves on the basis of their needs (see Section 3.6). In order to describe our proposal, in the following, we will focus on a single table and (mainly) on a single user.

3.3. Use case scenario

We sketch a very simple informal usage scenario, in order to provide an intuitive idea of the functionality supported by the semantic model we propose, from a user perspective. In Section 3.7 we will explain what happens "behind the scenes", thus providing a detailed description of how Semantic T++ works.

Aldo is a volunteer working for Our Planet, a NGO for environment safeguard. Some months before he had created a table (named *Our Planet*) to collaborate with a small group of other local volunteers. The table is currently populated by several objects, among which:

- A resolution by the Municipality of Champorcher (a village in Valle d'Aosta, a small mountain Italian region), concerning enlargement works to be done on a local old mule track. This table object is available as a link to a PDF document on the Municipality Web site (henceforth *resolution*).
- A news about a demonstration planned for May 14th for defending Champorcher mule track. This table object is available as a link to a news on Our Planet Web site (henceforth *news*).
- A video on Beppe Grillo's blog[1], talking against the enlargement project for the Champorcher mule track. This table object is available as a link to the video on Grillo's Web site (henceforth *video*).
- An official news by the Regione Valle d'Aosta about Champorcher mule track enlargement project. This table object is available as a link to the Regione Valle d'Aosta Web site, official news section (henceforth *newsVdA*).
- Four images concerning the Champorcher mule track and its surroundings. These table objects are available as JPEG files on Dropbox; see Section 4.1 for details (henceforth *image1/2/3/4*).
- Two email conversations, one between Aldo and the local Councilor for the Environment (having "Champorcher mule track" as subject), and the other one between the table participant Maria and some schools (having "workshop about ancient crafts" as subject). These table objects are available as links to email threads on Google Mail; see Section 4.1 for details (henceforth *email_thread1/2*).
- A *to-do* assigning Aldo the task of writing an article for an online local newspaper, discussing the situation of the Champorcher mule track. This table object is available as a to-do item, handled by the Collaborative Task Manager [2] (henceforth *to-do*).

Aldo sits at the *Our Planet* table and enters the UI area enabling him to access table objects. He asks for objects of type to-do and gets a short list where he can easily find the one assigned to himself. In order to write the article, Aldo needs to retrieve information available on the table concerning the Champorcher mule track, so he asks for the list of topics present on the table, getting *Champorcher*, *Champorcher mule track*, *demonstration 14/5*, *workshop about ancient crafts*, etc. Aldo selects *Champorcher mule track* (as we will see in Section 3.6 he could specify more than one topic), and gets the list of table objects having the Champorcher mule track as main topic, e.g.: *resolution*, *video*, *newsVdA*, *image1/2*, *email_thread1*. Aldo reads the

[1] Beppe Grillo is the leader of the well-known Italian political party *Movimento 5 stelle*.

resolution and decides to have a look at email exchanges on the topic, so he filters the previous list (objects having the Champorcher mule track as topic) by asking for email conversations only and gets *email_thread1*. After reading the email messages, Aldo creates a new table object (an HTML document, since the article will be published online), writes some text in it, and add a link to the resolution.

Now Aldo needs further information, thus he asks for the list of "objects of discourse" (i.e., sort of secondary topics) present on the table. He selects again *Champorcher mule track* (and possibly some other items like, for instance, *mule track enlargement project*), and he gets the list of table objects that talk (also) about the mule track, i.e. – besides the previous ones (having it as topic): *newsVdA*, *image3/4*, *email_thread2*. Aldo reads the news and decides to mention it in the article, so he adds a link to it.

At this point, Aldo decides to look for the "political" opinions about the issue, so he filters the previous list (objects having the Champorcher mule track as object of discourse) by selecting specific authors, like Beppe Grillo and local government (Regione Valle d'Aosta): he gets objects talking about the mule track whose authors are (possibly) sources of "political" opinions.

Then, he needs a couple of images: he thus asks for images having the Champorcher mule track as topic, getting *image1/2*: Aldo selects *image1* and add it to his article.

Finally, Aldo decides to leave the table and clicks on the "save&update" button: the table asks him to confirm or modify some properties of the new object, i.e.:

- *Candidate languages* = *written Italian*. Aldo confirms.
- *Candidate authors* = *Aldo*. Aldo confirms.
- *Candidate topics* = list of entities which are candidate to become the article topic (*Champorcher, Champorcher mule track*). Aldo selects *Champorcher mule track*.
- *Candidate objects of discourse* = list of entities which are candidate to become article objects of discourse (*Champorcher, Champorcher mule track, mule track enlargement project, Mont Avic park, demonstration 14/5, Ourty alpine pasture, Valle d'Aosta, Ayasse river, ...*). Aldo selects: *Champorcher, enlargement project, Mont Avic park, demonstration 14/5*, and adds a new object of discourse, i.e. *Our Planet* (since, in the article, he also talks about the activity of the NGO).

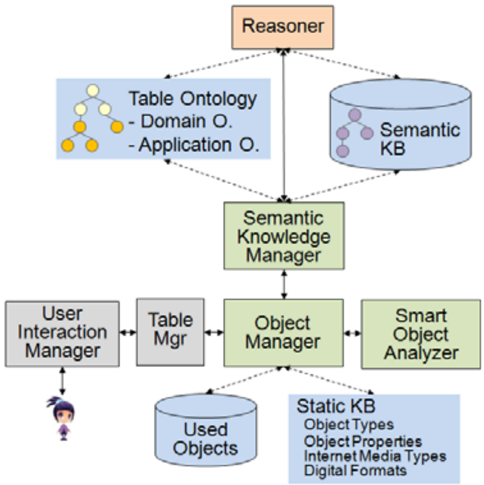

Fig. 1. Semantic T++ architecture (relevant components).

3.4. Architecture

Figure 1 shows the relevant components of Semantic T++ architecture. Arrows represent information exchange: plain arrows refer to API invocation, dotted arrows to knowledge bases access. The system knowledge is structured into the following modules:

- *Table Ontology*: accessed and managed by the Semantic Knowledge Manager, it represents the (static) semantic knowledge of the system. It is actually composed by two ontologies: a Domain Ontology containing the domain knowledge, i.e., concepts and relations pertaining to information objects, and an Application Ontology representing concepts and relations which characterize the specific application, i.e. the way in which Semantic T++ supports digital resource management.
- *Semantic KB*: accessed and managed by the Semantic Knowledge Manager, it contains all the facts about the individuals involved in the semantic representation of table objects. It refers to the Domain and Application Ontologies.
- *Used Objects DB*: accessed and managed by the Object Manager, it contains the list of objects lying on tables or included in objects lying on tables; for each object, it stores a unique identifier (IRI) and some non-semantic

information (i.e., the object location and two boolean flags: *isEditable* and *isOnTable*).

- *Static KB*: accessed and managed by the Object Manager, it contains references to elements used in the interaction with the user, i.e., the list of available object types (corresponding to some Table Ontology classes), a list of object properties (corresponding to semantic relations defined in the Table Ontology, e.g., *hasTopic*), a list of Internet Media Types (such as "text/html"), and a list of digital formats (such as PDF, GIF, ...). All these entities are uniquely identified by an IRI.

The application logic is handled by a set of components interacting with each other. The most important, as far as the semantic knowledge management is concerned, are the following:

- *Object Manager*: it acts as a sort of "object broker", since it manages the "used objects" (i.e., objects on a table or included in objects on a table) and plays a mediation role between the Table Manager (and thus, indirectly, the UI) and the components which represent the system "intelligence", i.e. the Semantic Knowledge Manager and the Smart Object Analyzer.
- *Semantic Knowledge Manager*: it manages the semantic descriptions of table objects (stored in the Semantic KB). It invokes the Reasoner, when required, and interacts with the Object Manager in order to get non-semantic information about table objects.
- *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.).
- *Reasoner*: it provides the system with inferred object features, which can be exploited to support the user in table objects management.

3.5. Semantic model

In this section we provide an intuitive account of the main aspects of the semantic model implemented in the Table Ontology, coupled with a minimum level of formal details. In order to achieve this goal, we describe ontology axioms at an abstract level, which better accounts for the relationships with the mentioned reference ontologies (DOLCE and OIO). To develop the proof-of-concept prototype, we implemented a light version of the Table Ontology, written in OWL, simplifying some aspects: for instance, ternary relations (which include time) have been simplified into binary predicates, by omitting the time parameter *t* (see below), which was useless in our current application context[2].

The most relevant class in the Domain Ontology is *InformationElement*, directly inherited from O-CREAM-v2[3] (see Section 2), which is the most general concept representing the set of all information elements: it is a subclass of *OIO* : *InformationObject* and all parts of an information element are information elements:

- $InformationElement(x) \rightarrow OIO : InformationObject(x)$
- $InformationElement(x) \wedge DOLCE : part(x, y, t) \rightarrow InformationElement(y)$

All table objects handled by T++ are instances of *InformationElement*. Moreover, in the Application Ontology, we introduced some specific subclasses (described below), 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, inherited by O-CREAM-v2, together with some subproperties introduced in our Application Ontology, namely:

- *DOLCE* : *part*(*x, y, t*); to represent relations such as the one between a document and an image included in it.
- *specifiedIn*(*x, y, t*); to represent relations such as the one between a document and the languages 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.
- *OIO* : *about*(*x, y, t*); this is an OIO property representing the relation between an information element and the individuals it "refers to". Since, from the user point of view, only a subset of such "referents" actually represent objects of discourse, we introduced a subproperty of *OIO* : *about* to account for them: $hasObjectOfDiscourse(x, y, t) \rightarrow OIO: about(x, y, t)$. Moreover, we introduced another

2 Handling time could be an interesting future enhancement of Semantic T++.

3 For the sake of readability, in the following we do not use *O-CREAM* prefix, thus O-CREAM-v2 concepts are actually considered as directly belonging to the Table Ontology.

subproperty of *OIO* : *about*, to represent, for instance, the relation between a hyperlink and the resource it points to: *identifies*(*x, y, t*) → *OIO*: *about*(*x, y, t*)

– *hasAuthor*(*x, y, t*); to represent the relation between an information element (e.g., a document) and its authors.
– *hasURL*(x, y, t); to represent the relation between an instance of *WebResource* (see below) and its URL.

We also introduced the class *WebResource*, to represent individuals which can be associated to a URL:

– *WebResource(x)* ↔ *(∀t) (∃y) hasURL(x, y, t)*

The subclasses of *InformationElement* introduced in the Application Ontology to represent the different types of table objects are the following:

– *Document* (to represent documents of different kinds, e.g. reports, letters, brochures, official documents, and many others).
– *Image*, *Video*, *Audio* (to represent images, videos and audio items).
– *EmailThread* (to represent email conversations).
– *ToDo* (to represent tasks assigned to table participants).
– *WebResourceLink* (to represent hyperlinks, typically used within HTML documents); individuals belonging to this class identifies an instance of *WebResource*: *WebResourceLink*(*x*) → (∃*t*)(∃*y*)(*WebResource*(*y*) ∧ *identifies*(*x, y, t*))
– *InformationWebResource*; to represent information elements which are also Web resources, such as Web pages: *InformationWebResource*(*x*) → *InformationElement*(*x*) ∧ *WebResource*(*x*)

Following the framework provided by OIO and O-CREAM-v2, the characterization of information elements requires other properties besides those mentioned above. A complete account of such properties is out of the scope of this paper (see [23] for a more detailed analysis). However, we will mention the most important ones for Semantic T++.

Two information elements, *x* and *y*, can be related by the *encodes(x, y, t)* relation (holding during time *t*); in this case, *x* provides an encoding for *y* such that *y* can be derived from *x*: for example, a JPEG information element (*x*) encodes an image (*y*) and an image editor can visualize the encoded image (*y*) given the encoding element (*x*); see Figure 2.

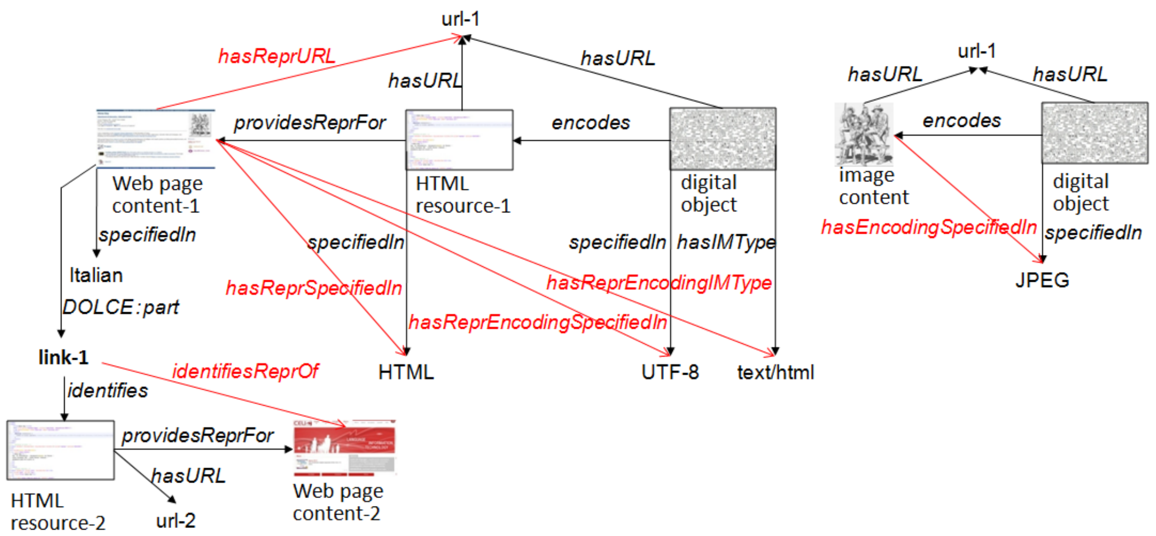

Fig. 2. Semantic T++ "shortcut" properties.

Another relevant relationship is the one between, for instance, a Web page and its HTML representation. In fact, markup languages such as HTML introduce a further layer, between the "content" information element (i.e., the Web page) and its digital encoding, e.g. a UTF-8 digital object. Thus, the UTF-8 digital object *encodes* the HTML resource, which *providesRepresentationFor* the Web page. Moreover, the UTF-8 digital object is

SpecifiedIn UTF-8 and *hasInternetMediaType* text/html; see Figure 2.

Representing table objects as information elements with all of these properties leads to a quite complex model. In order to keep representations more concise and possibly closer to the user's point of view, we introduced some "shortcut" properties, which can be used to simplify the representation. Figure 2 graphically displays the most relevant "original" properties (in black), as well as the "shortcut" relations (in red), showing an example involving a Web page (*Web page content-1*) containing a hyperlink (*link-1*) to another Web page (*Web page content-2*), and a simpler example, involving an image (*image content*).

The "shortcut" properties are the following:

- *hasRepresentationSpecifiedIn*(x, y, t); to represent, for instance, the fact that a Web page x is (indirectly) specified in HTML (y): *hasRepresentationSpecifiedIn*(x, y, t) ↔ (∃z) (*InformationElement*(z) ∧ *Language*(y) ∧ *providesRepresentationFor*(z, x, t) ∧ *specifiedIn*(z, y, t))
- *hasRepresentationEncodingSpecifiedIn*(x, y, t); to represent, for instance, the fact that a Web page x is (indirectly) specified also in UTF-8 (y): *hasRepresentationEncodingSpecifiedIn*(x, y, t) ↔ (∃z)(∃w) (*InformationElement*(z) ∧ *InformationElement*(w) ∧ *Language*(y) ∧ *providesRepresentationFor*(z, x, t) ∧ *encodes*(w, z, t) ∧ *specifiedIn*(w, y, t))
- *hasRepresentationEncodingInternetMediaType* (x, y, t); to represent, for instance, the fact that a Web page x (indirectly) has text/html InternetMediaType (y): *hasRepresentationEncodingInternetMediaType* (x, y, t) ↔ (∃z)(∃w) (*InformationElement*(z) ∧ *InformationElement*(w) ∧ *InternetMediaType*(y) ∧ *providesRepresentationFor*(z, x, t) ∧ *encodes*(w, z, t) ∧ *hasInternetMediaType* (w, y, t))
- *identifiesRepresentationOf*(x, y, t); to represent, for instance, the fact that a hyperlink x (indirectly) identifies a Web page y: *identifiesRepresentationOf*(x, y, t) ↔ (∃z) (*InformationElement*(z) ∧ *identifies*(x, z, t) ∧ *providesRepresentationFor*(z, y, t))
- *hasEncodingSpecifiedIn*(x, y, t); to represent, for instance, the fact that an image x is (indirectly) specified in JPEG (y): *hasEncodingSpecifiedIn*(x, y, t) ↔ (∃z) (*InformationElement*(z) ∧ *encodes*(z, x, t) ∧ *specifiedIn*(z, y, t))
- *hasRepresentationURL*(x, y, t); to represent, for instance, the fact that a Web page x (indirectly) has a URL (y): *hasRepresentationURL*(x, y, t) ↔ (∃z) (*InformationElement*(z) ∧ *providesRepresentationFor*(z, x, t) ∧ *hasURL*(z, y, t))

A very important fragment of the proposed semantic model refers to some particular properties which model *candidate relationships*. The idea behind such properties is that the system can infer *candidate* object features; e.g., the Reasoner can infer that Mont Avic is a *candidate* object of discourse of Aldo's article from the fact that it is an object of discourse of an included image. The axioms supporting such kind of inferences are like the following:

- *InformationElement(x) ∧ DOLCE : part(x, z, t) ∧ hasObjectOfDiscourse(z, y, t) → hasCandidateObjectOfDiscourse(x, y, t)*
- *InformationElement(x) ∧ DOLCE : part(x, z, t) ∧ indentifiesRepresentationOf(z, w, t) ∧ hasObjectOfDiscourse(w, y, t) → hasCandidateObjectOfDiscourse(x, y, 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 Mont Avic is actually an object of discourse of the article, then a new relation *hasObjectOfDiscourse(article, MontAvic, t)* is added to the knowledge base. Analogous axioms are available for the *hasTopic* relation, to support the inference of *hasCandidateTopic* relationships, and for the *specifiedIn* relation, to support the inference of *candidateSpecifiedIn* relationships (e.g., to infer that a document is probably written in Italian if it includes a hyperlink pointing to a Web page written in Italian).

3.6. User interaction

We developed a proof-of-concept prototype consisting in two parts: (a) a cloud application implementing the backend functionality, described in Section 4.1; (b) A User Interface (UI) mockup, aimed at implementing the interaction flow between a table and a user who wants to select objects.

Figure 3 shows a general view of Semantic T++ UI. On the right hand side there are two panels: the list of tables (the one in focus is highlighted) and the list of participants at the table in focus; participants not currently sitting at the table are in light grey. On the left hand side, users can access the main table functionalities, namely table objects selection and three basic services: *Chat* and *Blackboard* (for synchronous and asynchronous communication, respectively), and a shared *Calendar*. Other functionalities (e.g., settings) are available through the "menu" button in the left-bottom corner. The central area shows a panel referring to the functionality in focus (in Figure 3, the Calendar).

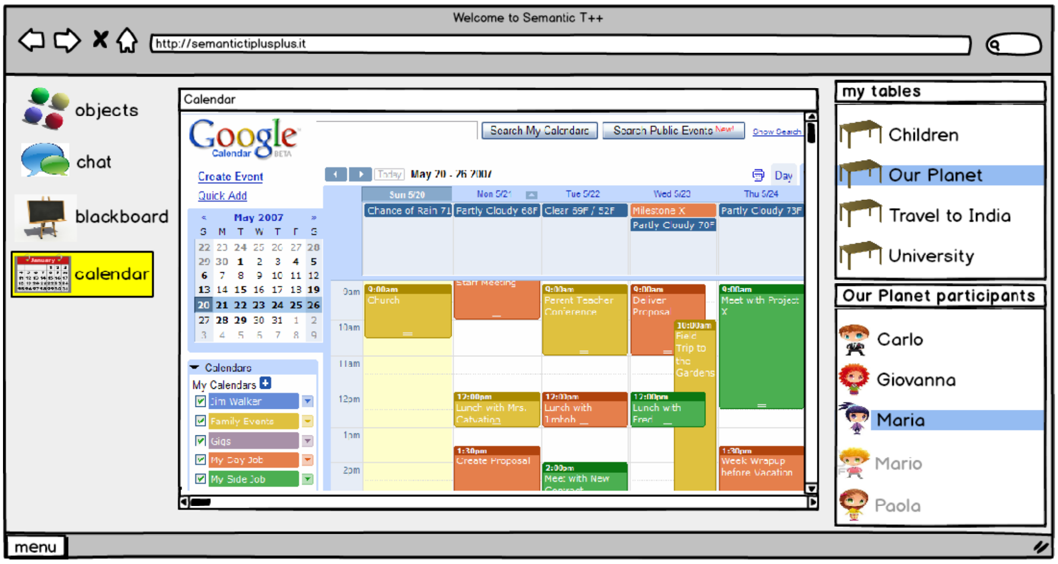

Fig. 3. Mockup of Semantic T++ User Interface: general view.

In this paper we focus on objects selection, showing how formal semantic representations handled by the system can support users in this task. Figure 4 shows the central panel, triggered by a click on the "objects" label on the left. This panel is composed of two tabs, one to access all table objects, and the other one (in focus) for selecting them. Through this tab, users can specify multiple criteria for objects selection, as we exemplified in Section 3.3. In the current mockup, users can select properties (e.g., object type, topic, objects of discourse, authors, etc.) and specify values.

Figure 4 shows the panel as it appears at the beginning of the interaction (the user has to select a property), and after a number of steps; in this example, the user specified values for the following properties: main topic, objects of discourse, resource type, and authors. Different choices for properties are obviously possible: there are no constraints on the number or order of selected properties.

When clicking on the "get objects" button, the user is provided with the list of objects matching her criteria, as shown in Figure 5. Objects can be edited/visualized by clicking on their names. Moreover, for each object, the UI reports its visibility (*public* means visible to all table participants; objects could also be *private*, i.e., only visible to their creator) and provides access to free comments (if available), to the semantic description of the object ("properties" column), and to a delete functionality. Moreover, the user can modify the search by adding or removing criteria through the corresponding link, available for each selection results.

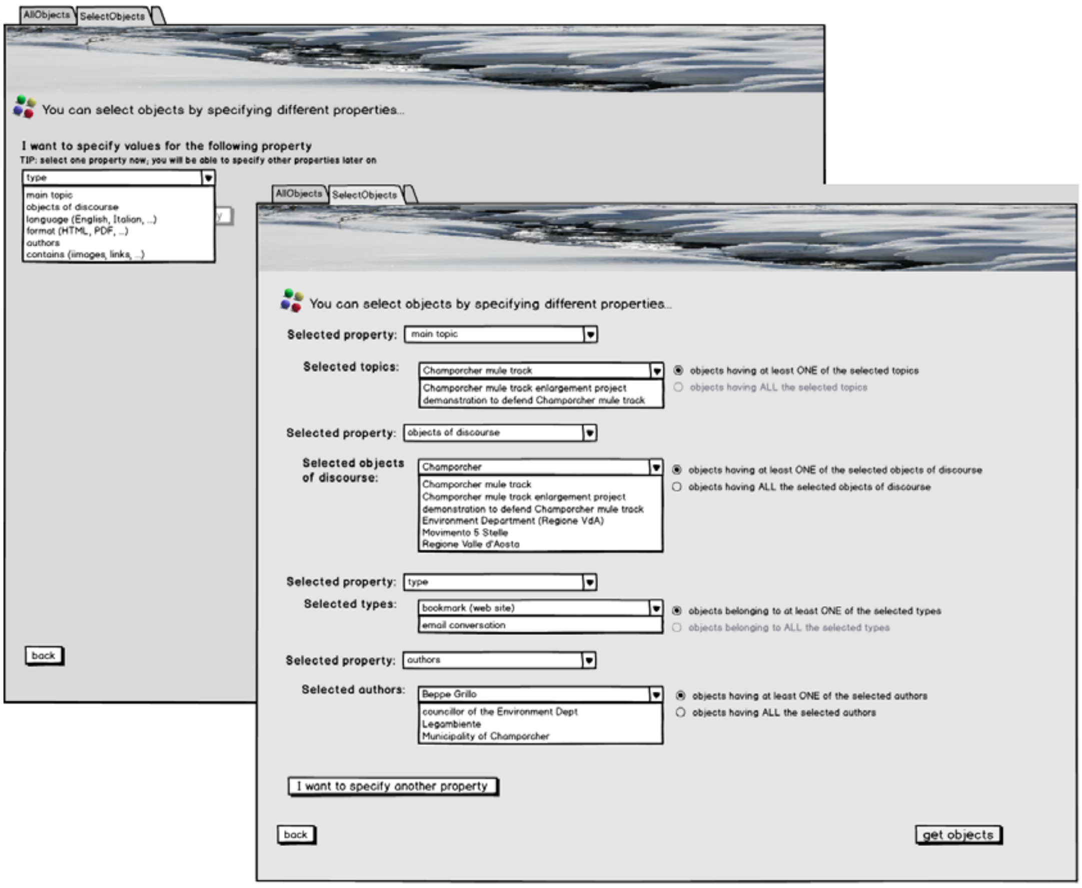

Fig. 4. Mockup of Semantic T++ User Interface: objects selection panel after several steps.

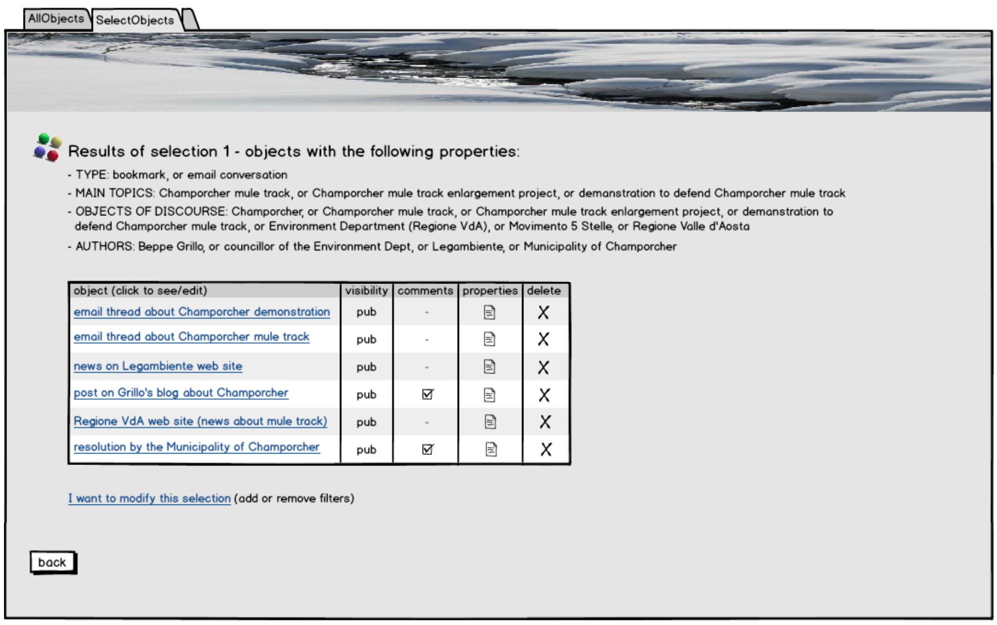

Fig. 5. Mockup of Semantic T++ User Interface: list of objects matching user's criteria.

The current Semantic T++ UI mockup deserves some further comments. It has been designed and implemented with the goal of testing the *interaction functionalities* concerning objects selection, having Web design and usability issues in mind, but without focusing on them. The result is a very "basic" and somehow "old fashion" UI, which definitely needs improvements, but which has been useful to focus on functionalities. However, a couple of issues are worth to be mentioned. We decided to use standard drop-down lists, but obviously more "Web 2.0 oriented" fields, closed to users tagging experience, could have been used (see Figure 6). Moreover, the system should provide results incrementally, while the user goes on specifying selection criteria, instead of waiting for a click on the "get objects" button. As we will see in Section 4.2, these aspects did not influence users answers to our evaluation questionnaire, which did not focus on them.

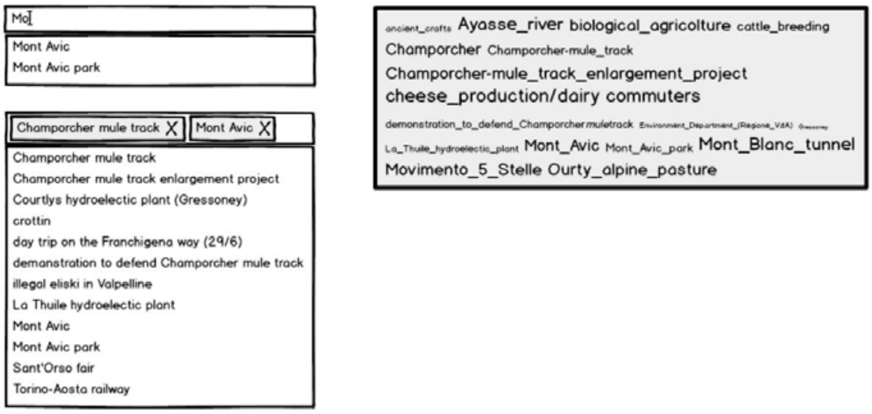

Fig. 6. Mockup of Semantic T++ User Interface: possible alternative solutions.

3.7. Backstage of the use case scenario

In the following we will describe how the presented semantic model supports the usage scenario sketched in Section 3.3. Browsing the scenario, there are several points in which Aldo defines some selection parameters and the system provides him with a list of objects. For example, at the very beginning, Aldo asks for a list of to-dos available on the table; later on, he asks for the list of table objects having the Champorcher mule track as main topic. In other points, Aldo combines multiple criteria, e.g., in order to retrieve email conversations concerning the Champorcher mule track. In all these cases, the Object Manager (OM) receives a request from the Table Manager (TM) and invokes the Semantic Knowledge Manager (SKM) asking for individuals fulfilling the criteria expressed by the selection parameters. For example, in order to get all email threads having the Champorcher mule track as main topic, the OM invokes the SKM providing it with the following JSON object (www.json.org):

{ 'topics': ['http://www.di.unito.it/semtpp/resources/champorcherMuleTrack'],
'and_topics': false,
'types': ['http://www.di.unito.it/semtpp/tableontology#EmailThread'],
'and_types': false }

All attributes (e.g., *topics*, *types*) are optional, and they are specified only if the user has selected the corresponding property. Each attribute corresponding to a selected property is a list of IRI identifying the selected values, and is coupled with a boolean attribute (e.g., *and_topics*) specifying if the items in the list must be interpreted in AND (*true*), or OR (*false*, which is the default value). Values for the object type property are conveyed by references to Table Ontology classes (e.g., *http://www.di.unito.it/semtpp/tableontology#EmailThread*), while values for other properties, such as *hasTopic*, are conveyed by references to individuals in the Semantic KB (e.g., *http://www.di.unito.it/semtpp/resources/champorcherMuleTrack*).

The sketched scenario also includes the creation of a new table object (the article written by Aldo), the inclusion of images and hyperlinks in it, and the user contribution to the object semantic characterization.

The creation of a new table object triggers: (a) the creation of a new record in the *Used Objects DB*, and thus of an IRI for the new object (by the OM); (b) the creation of a new instance of the class specified by the user when selecting the type for the new object (by the SKM).

When the user clicks on the "save&update" button, the OM invokes the SKM, asking it to update the object semantic representation. The SKM gets, from the Smart Object Analyzer (SOA), the analysis of the object content, which contains the following attributes (see Figure 2, Section 3.5): *encoding language* (e.g. HTML5), used to set values for the property *hasRepresentationSpecifiedIn(x, y, t)*; *contentType* (e.g., text/html), used to set values for the property *hasRepresentationEncodingInternetMediaType(x, y, t)*; *encodingFormat* (e.g., UTF-8), used to set values for the property *hasRepresentationEncodingSpecifiedIn(x, y, t)*; *location* (resource URL), used to set values for the property *hasRepresentationURL(x, y, t)*; *languages* detected (e.g., Italian) and *authors* (if available, e.g., in HTML meta-tags), which are proposed to the user for confirmation (see below); the list of all *images/videos/audios* and *hyperlinks* included in the document, as well as the *encoding format* and *contentType* of the resources identified by hyperlinks. On the basis of this information, the SKM updates the semantic representation of the object; Figure 7 shows in a graphical format the final semantic representation of Aldo's article.

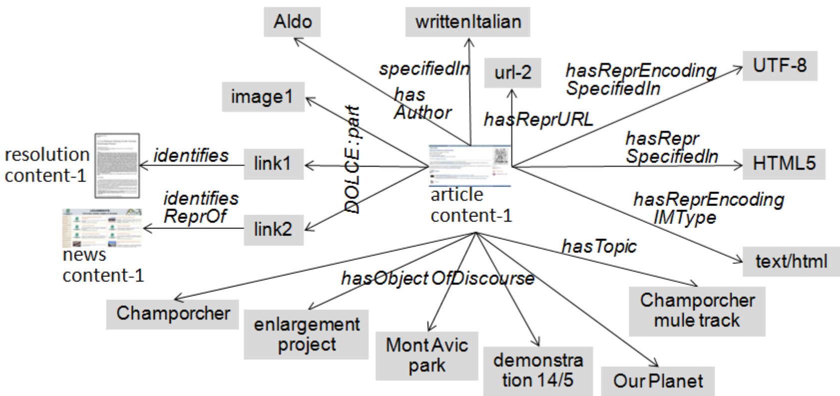

Fig. 7. Semantic representation of Aldo's article

Moreover, the SKM runs the Reasoner, which infers other interesting properties of the new object, among which:

– *candidateSpecifiedIn*(*article content-1*, *written Italian*)
– *hasCandidateTopic*(*article content-1*, {*Champorcher*, *Champorcher mule track*})
– *hasCandidateObjectOfDiscourse*(*article content-1*, {*Champorcher, Champorcher mule track, mule track enlargement project, Mont Avic park, demonstration 14/5, Ourty alpine pasture, Valle d'Aosta, Ayasse river*})

These candidate properties are provided to the OM in order to ask the user a confirmation, a selection, or a change. Notice that the list of candidate languages is composed by the merge of languages inferred by the Reasoner and languages detected by the SOA. Moreover, the user is also asked to confirm the authors eventually identified by the SOA.

In our scenario, Aldo confirms to be the author of the article, confirms the candidate language, selects *Champorcher mule track* as *topic*, selects *Champorcher*, *enlargement project*, *Mont Avic park*, *demonstration 14/5* as objects of discourse, and adds a new one: *Our Planet*. The SKM builds the new relations, thus the article semantic representation is updated, as shown in Figure 7.

Before closing this section, it is worth briefly commenting on the user contribution to objects semantic characterization. When a new object is created (or when an existing one is modified, for example when a table participant includes a new image or link in it), Semantic T++ builds (or update) the semantic representation in three steps:

1. The SOA automatically sets some properties (e.g., mereological composition, types of the

parts, formats), and propose candidates for others (e.g., languages and authors).
2. Other properties (e.g., candidate topics) are inferred by the Reasoner.
3. Candidate properties (suggested by the SOA or by the Reasoner) are proposed to the user for a confirmation; the user can always add properties (e.g., objects of discourse not suggested by the system, as in our scenario).

Especially for long-lasting tables, already populated by objects having a semantic characterization, a great part of the semantic representation of new (or updated) objects is automatically built by the system. Moreover, when the user intervention is needed, the system always provides her with suggestions which greatly simplify her work. Finally the user is never forced to confirm or reject candidate properties: if she does not do it, the system keeps them in "candidate" state.

4. Evaluation

In [15] we described an evaluation of T++ based on two controlled experiments, in which users were asked to perform collaborative tasks both using standard collaboration tools and using T++. The two tests aimed at verifying how T++ supports communication and resource sharing (first test), and shared resources retrieval (second test). We measured the time needed to perform the required tasks as well as user satisfaction (by asking participants to evaluate – on a 1 to 5 scale – the most relevant aspects, e.g., efficiency in sharing a resource, efficiency in finding a shared resource). The results showed that performing the required tasks with T++ is faster than without it, meaning that T++ is better in terms of efficiency. Moreover, user satisfaction was meaningfully higher with T++, demonstrating that it provides users with a better interaction experience if compared with common available sharing tools.

These experiments, however, were performed on the previous, non-semantic, version of the system. Thus, since the semantic model presented in this paper needs a specific evaluation, we went through a twofold approach, aimed at testing both the technical feasibility of the semantically-enhanced system and the effectiveness of the user interaction it supports (as far as access to digital resources is concerned). Thus:

- We developed a proof-of-concept prototype consisting of two parts: (a) a cloud application implementing the *backend* functionalities; (b) A *frontend* mockup, implementing the *interaction functionalities* of the UI devoted to objects selection.
- We submitted a semi-structured questionnaire to 20 potential users, after an interaction session with the Semantic T++ mockup (focused on objects selection).

4.1. The prototype

The backend prototype is a Java Web Application deployed on the Google Application Engine (appengine.google.com) and is based on a Three-Tier Architecture. The First Tier (Presentation Layer), can be run in a Web browser window or as a native Android Application, which communicates with the Second Tier through a RESTful Web Services model, using JSON objects to exchange complex data. The Third Tier (Persistence Layer), is built on top of the Google Application Engine object datastore, using the Java Data Objects (JDO) interface. The Second Tier (Controller and Business Logic) is based on a set of Java servlets which implements the application logic.

The current prototype exploits Dropbox (www.dropbox.com) and Google Drive (www.google.com/drive) API to store files corresponding to table objects, Google Mail to handle email conversations, and the Collaborative Task Manager [2] to handle to-do lists. However, 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, on each table, the preferred tools to be exploited for object sharing and editing.

Concerning the modules which represent the core of Semantic T++ objects management, namely the Object Manager (OM), the Semantic Knowledge Manager (SKM), and the Smart Object Analyzer (SOA), they are all written in Java. Moreover, the SKM uses the OWL API library (owlapi.sourceforge.net) to access semantic knowledge, while the SOA exploits a Parser Service, written in Python, able to analyze HTML documents[4]. Finally, the Reasoner is based on

4 We focused on HTML documents for three main reasons: (1) HTML is easier than other (proprietary) formats (e.g., Google Drive/Microsoft Word) to parse. (2) Writing for the Web is everyday more common. (3) The semantic characterization of HTML documents introduces some interesting aspects (e.g.,

Fact++ (owl.man.ac.uk/factplusplus). The Table Ontology and the Semantic KB are written in OWL (www.w3.org/TR/owl-features).

The *frontend* mockup is a sequence of interactive UI panels (created with Balsamiq: balsamiq.com) supporting all the steps required for objects selection interaction. Some of them are shown in Figures 4 and 5.

4.2. Results of the users test

We selected 20 participants (10 males and 10 females), through a preliminary interview, aimed at identifying users representative of a specific, but wide, category, namely people between 20 and 60, with a Master degree, and used to use Internet and email daily, both in their business and personal life.

We provided participants with a general description of the system and a more detailed presentation of the possibilities for selecting resources on a table. We then described a usage scenario, in which Maria, a volunteer of the Our Planet NGO, has to write an article about the defense of the old Champorcher mule track, against the Municipality enlargement project. Maria sits at the Our Planet table, which is populated by several objects, concerning the various NGO activities. Table objects are already characterized by semantic properties (main topic, objects of discourse, formats, etc.). To write the article, Maria needs to know which are the "political" opinions, expressed by parties, movements, associations, public administrations, and so on about the Champorcher mule track issue. Users are asked to imagine they are Maria and to go through the following steps: (1) tell Semantic T++ their content need (resources talking about the mule track issue); (2) narrow their search to Web sites and email conversations; (3) tell the system they are looking, in particular, for resources having "political" authors.

After the interaction, we asked users to answer two groups of questions. In the first group, they had to provide a score, between 1 (negative) and 5 (positive), and possibly a comment; Table 1 shows the mean and the standard deviation obtained for each question. The second group included open questions: since users were not forced to answer, Table 2 reports the number of users who answered.

The results of the first group of questions (Table 1) show that test participants liked the way Semantic T++ provides access to digital resources (see the 4.3 mean value obtained by question Q4). Moreover, the main innovative aspects of Semantic T++, namely a unique access to heterogeneous objects (Q1) and the possibility of combining different criteria to select table resources (Q2), are appreciated by all users, with substantial agreement (low standard deviation). The availability of a way to search for objects containing a specific resource (Q3), instead, is not so popular (although participants tend to disagree, as the higher standard deviation shows). Some interesting suggestions, however, can be found in the comments that some participants added to their scores for Q3. For example, some users said that knowledge about resources contained in documents (like images, links, etc.) could be exploited to extend the selection criteria also to object parts; others suggested a functionality enabling them to asks the systems something like "give me all documents containing an image of Monte Bianco".

Answers to questions belonging to the second group revealed interesting results too. Almost all participants answered the questions about the best and worst aspects (Q5, Q6), while a few more than half of them provided further opinions (Q7). As far as the best feature is concerned (Q5), the most popular answer has been the possibility of selecting resources on the basis of their "content" (using *topic* and *objects of discourse* properties), indicated by the majority of users, immediately followed by the possibility of selecting different types of objects (documents, bookmarks, emails, ...) through a single UI. Some users also mentioned, as a positive feature, the fact that, for each property, the list of possible values is provided.

As far as the worst feature is concerned (Q6), many users indicated the fact that the specification of selection criteria is too structured (e.g., it requires a sequence of steps). Someone added that for personal resources management, it seems to be too complex, while it could be more helpful to handle shared resources. Some users also claimed that the meaning of some properties is not clear, and suggested to provide some explanation (for instance, through tooltips).

The last question (Q7) offered participants the possibility of providing us with interesting suggestions. Many users complained about the "old fashion" style of the UI: in Section 3.6 we already discussed this aspect. Some users suggested to add "standard" functionalities such as a simple keyword search and an advanced search to combine criteria by means of AND/OR operators.

the need of distinguishing between *information content* and *content of the HTML file*); see Section 3.5.

In summary, the results of the post-test questionnaire seem to support the hypothesis that Semantic T++ provides an enhancement with respect to the problems identified in Section 1. In particular, answers to Q1 and Q5 confirms that the possibility of accessing different types of objects through a single UI is a highly appreciated feature, and our hypothesis is that this feature contributes to provide a better (less fragmented) user experience. Also the good result represented by answers to Q2, coupled with indications in Q5, are promising: the appreciation of the possibility of selecting resources on the basis of their "content" means that the semantic enhancement of T++ represents a step in the right direction.

Table 1: Results of post-test questionnaire: users scores

	Mean	**St. Dev.**
Q1: Access to different resources through a unique user interface	4,7	0,4702
Q2: Possibility of combining different criteria to select resources	3,95	0,7592
Q3: Access to object parts	2,95	1,3169
Q4: Would you like to use a "virtual desktop" offering these features for resources access?	4,3	0,7327

Table 2: Results of post-test questionnaire: open questions

	Answers (/20)
Q5: Which is the best feature for you?	19
Q6: Which is the worst feature for you?	18
Q7: Tell us freely your comments	12

Before closing this section, we would like to summarize some suggestions which can contribute in defining guidelines for building knowledge-based systems which support user interaction on the Web.

Although users appreciated the new functionalities, some of them asked to couple them with standard ones (e.g., keyword search). This claim suggested the following guideline: never *substitute* functionalities users are used to, but always *add* new possibilities. Some features of Semantic T++ already follow this guideline: for example, selecting resources on the basis of their format (a feature typically available in standard file systems) is one of the possible selection criteria.

Another important aspect concerns knowledge formalization, which usually implies a highly structured user interaction. This is partially appreciated (e.g., the provision of the list of property values), but it is also seen as a negative overload (as some users explicitly said). This consideration suggests us the following guideline: when endowing the system with formal semantic knowledge, ensure to maximize flexibility in user interaction.

5. Conclusions

This paper presented a semantic model supporting users in a flexible and effective access to personal digital resources. Such a model is exploited within T++, a collaborative environment characterized by innovative features, the most important being the uniform management of heterogeneous resources (documents, bookmarks, emails, to-dos, etc.). The semantic enhancement of T++ offered the possibility of providing users with a smart and flexible access to their personal and shared digital resources: users can specify and combine different criteria (resource content, format, type, author, etc.), in order to select resources, independently from the application used to handle them (file system, mailer, browser, ...).

The described approach, validated by the results of a preliminary user evaluation, shows that Artificial Intelligence, and in particular a Knowledge Representation approach grounded into well-founded formal ontologies, can contribute in enhancing Human Computer Interaction tools such as Personal Information Management systems.

Many interesting aspects of the proposed approach are not discussed here. For example, since the presentation is focused on the advantages in the perspective of resources selection, other aspects of user interaction are not faced (for example, deleting table objects or changing object properties). Moreover, an aspect which plays a major role in T++, but is not analyzed in this paper, is collaboration. We started facing this issue in [15] and [16], but the collaborative scenario poses interesting challenges to the semantic version of T++ which are worth to be further investigated.

Finally, we have planned to perform a comparative evaluation of Semantic T++ in terms of cognitive overload and capability to retrieve (personal and shared) resources and thus to avoid loss of information.

References

[1] F. Abel, N. Henze, D. Krause and M. Kriesell, Semantic Enhancement of Social Tagging Systems, in: *Web 2.0 & Semantic Web*, V. Devedžić and D. Gašević, eds., Springer, Heidelberg, 2010, pp. 25–56.

[2] L. Ardissono, G. Bosio, A. Goy, G. Petrone, M. Segnan and F. Torretta, Collaborative Service Clouds, *Int. J. Information Technology and Web Engineering* **5**(4) (2010), 23–39.

[3] J. E. Bardram, Activity-based computing: Support for mobility and collaboration, *Ubiquitous Computing–Personal and Ubiquitous Computing* **9**(5) (2005), 312–322.

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

[5] I. Birzniece, Artificial Intelligence in Knowledge Management: Overview and Trends, *Scientific Journal of Riga Technical University*, **11** (2011), pp. 5–11.

[6] R. Boardman, R. Spence and M. A. Sasse, Too Many Hierarchies? The Daily Struggle for Control of the Workspace, in: *Human-Computer Interaction: Theory and Practice*, J. A. Jacko and C. Stephanidis, eds., Erlbaum Associates, Mahwah, NJ, 2003, pp. 616–620.

[7] S. Borgo and C. Masolo, Foundational choices in DOLCE, in: *Handbook on Ontologies – Second Edition*, S. Staab and R. Studer, eds., Springer, Heidelberg, 2009, pp. 361–381.

[8] J. G. Breslin, A. Passant and S. Decker, *The Social Semantic Web*. Springer, Heidelberg, 2009.

[9] A. Civan, W. Jones, P. Klasnja and H. Bruce, Better to organize personal information by folders or by tags? The devil is in the details, *American Society for Information Science and Technology* **45** (2009), 1–13.

[10] L. Diao, M. Zuo and Q. Liu, The Artificial Intelligence in Personal Knowledge Management, in: *2nd Int. Symposium on Knowledge Acquisition and Modeling*, IEEE Press, New York, NY, 2009, pp. 327–329.

[11] P. Dourish, W. K. Edwards, A. LaMarca, J. Lamping, K. Petersen, M. Salisbury, D. B. Terry and J. Thornton, Extending document management systems with user-specific active properties, *ACM Trans. Inf. Syst.* **18**(2) (2000), 140–170.

[12] L. Drăgan, R. Delbru, T. Groza, S. Handschuh and S. Decker, Linking Semantic Desktop Data to the Web of Data, in: *The Semantic Web – ISWC 2011*. LNCS, vol. 7032, L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy and E. Blomqvist, eds., Springer, Heidelberg, 2011, pp 33–48.

[13] A. Gangemi, S. Borgo, C. Catenacci and J. Lehmann, *Task Taxonomies for Knowledge Content*, Metokis Deliverable D07, 2005.

[14] A. Goy, G. Petrone and M. Segnan, Oh, no! Not Another Web-based Desktop!, in: *Int. Conf. on Advanced Collaborative Networks, Systems and Applications*, XPS Press, Wilmington, 2012, pp. 29–34.

[15] A. Goy, G. Petrone and M. Segnan, A Cloud-Based Environment for Collaborative Resources Management, *Int. J. Cloud Applications and Computing* **4**(4) (2014), in press.

[16] A. Goy, D. Magro, G. Petrone and M. Segnan, Collaborative Semantic Tables, in: *Workshop on Semantic Web Collaborative Spaces at ISWC 2014*, CEUR-WS, in press, 2014.

[17] Indratmo and J. Vassileva, A Review of Organizational Structures of Personal Information Management, J. *Digital Information* **9**(26) (2008), 1–19.

[18] W. Jones, *Keeping found things found*, Elsevier, Amsterdam, 2007.

[19] V. Kaptelinin and M. Czerwinski, Introduction: the Desktop Metaphor and New Uses of Technology, in: *Beyond the Desktop Metaphor*, V. Kaptelinin and M. Czerwinski, eds., MIT Press, Cambridge, MA, 2007, pp. 1–12.

[20] V. Kaptelinin and M. Czerwinski, eds., *Beyond the Desktop Metaphor*, MIT Press, Cambridge, MA, 2007.

[21] D. R. Karger, *Haystack*: Per-User Information Environments Based on Semistructured Data, in: *Beyond the Desktop Metaphor*, V. Kaptelinin and M. Czerwinski, eds., MIT Press, Cambridge, MA, 2007, pp. 49–100.

[22] M. Lansdale, The psychology of personal information management, *Applied Ergonomics* **19**(1) (1988).

[23] D. Magro and A. Goy, A Core Reference Ontology for the Customer Relationship Domain, *Applied Ontology* **7**(1) (2012), 1–48.

[24] R. Pak, S. Pautz and R. Iden, Information organization and retrieval: An assessment of taxonomical and tagging systems. *Cognitive Technology* **12**(1) (2007), 31–44

[25] P. Ravasio and V. Tscherter, Users' Theories of the Desktop Metaphor, or Why We Should Seek Metaphor-Free Interfaces, in: *Beyond the Desktop Metaphor*, V. Kaptelinin and M. Czerwinski, eds., MIT Press, Cambridge, MA, 2007, pp. 265–294.

[26] L. Sauermann, A. Bernardi and A. Dengel, Overview and Outlook on the Semantic Desktop, in: *Workshop on The Semantic Desktop at ISWC 2005*, vol. 175, CEUR-WS, 2005.

[27] D. Smith, C. Irby, R. Kimball, W. Verplank and E. Harslem, Designing the Star user interface, *Byte* **7** (1982), 242–282.

[28] J. Teevan, C. Alvarado, M. S. Ackerman and D. R. Karger, The perfect search engine is not enough: a study of orienteering behavior in directed search, in: *SIGCHI Conf. on Human Factors in Computing Systems*, ACM Press, New York, NY, 2004, pp. 415–422.