

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

## User-Friendly Interaction in an On-line System Based on Semantic Technologies

### **This is the author's manuscript**

*Original Citation:*

*Availability:*

This version is available <http://hdl.handle.net/2318/91752> since 2015-12-08T11:40:17Z

*Publisher:*

Springer

*Published version:*

DOI:10.1007/978-3-642-28082-5

*Terms of use:*

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

# User-Friendly Interaction in an On-line System Based on Semantic Technologies

Anna Goy and Diego Magro

Dipartimento di Informatica, Università di Torino, C. Svizzera 185, Torino, Italy  
{annamaria.goy, diego.magro}@unito.it

**Abstract.** Nowadays, SME have to take into account new business and management approaches, like CRM, as well as the support provided by ICT. In this scenario, SME would take advantage from a Web-based service like ARNEIS, supporting an intelligent matching between supply and demand of CRM-related tools. Such a service needs a detailed knowledge base, and a friendly user interface (UI) enabling users to interact with the formal knowledge base. In this chapter we claim that both the knowledge representation and the user interaction mechanisms have to be based on a domain analysis (how users talk about CRM). The chapter presents three UI, based on on-line forms, business processes, and natural language. All three exploit semantic templates, i.e., formal representations of dialog topics (key concepts in the descriptions of CRM activities). The proposed approach suggests a general solution to enable users to enter formal representations in systems based on semantic technologies.

**Keywords:** User interaction, User interfaces, Ontology, e-Business, Semantic web.

## 1 Introduction

In order to be competitive in the globalized market Small-to-Medium sized Enterprises (SME) have to take care of two major aspects: (a) The support provided by Information and Communication Technologies (ICT) to their business and, in particular, the trend of relying more and more on Internet and Web technologies: Web-based solutions are available for almost any kind of application, supported by innovative technologies at different levels, ranging from Cloud Computing infrastructures [12] [15], to Web Services [1], to Semantic Web [2]. (b) New business and management approaches and, in particular, the way to handle relationships with customers. The new market, in fact, requires personalized approaches to the single customer and flexible offers that need to be updated frequently. Moreover, in order to be aware of the market and customer behavior trends, data about offers, sales, communications with customers, and so on have to be elaborated very rapidly, to support management and marketing decisions.

For these reasons, Customer Relationship Management (CRM) [16], and ICT products and services supporting it, have received an increasing attention in the last couple of decades, not only from large enterprises, but also from SME. The key

principle of the CRM approach is a one-to-one marketing strategy that implies establishing personalized relationships with the single customer, by producing personalized offers, pricing, after-sale services, etc. Moreover, technological innovative solutions could effectively support the CRM approach, given that CRM requires the processing, integration, and analysis of a huge amount of heterogeneous knowledge (about customers, sales, communications, etc.), as well as effective, fast and integrated communication tools.

Within this scenario, SME would get great benefits from a Web-based service supporting an intelligent matching between supply and demand for CRM-related tools, and such a service can only be based on a clean, complete, and sharable formal knowledge representation of the concepts related to CRM.

In order to face this challenge, we designed an architecture for a Web-based intelligent system supporting SME in finding suitable software solutions for their business, and we chose CRM as a testbed for this architecture. On the basis of this architecture, we developed ARNEIS (Advanced Repository for Needs of Enterprises and Innovative Software), a prototype implementation of a Web-based repository of descriptions of software solutions for CRM, that exploits Web Services and Semantic Web technologies. ARNEIS users are: (a) ICT companies (i.e., software houses) that offer software solutions for CRM, but can find it difficult to get in contact with their potential customers; (b) SME that aim at finding software products or services supporting their CRM activities, but lack the know-how to find the ICT solution fitting their needs.

Such an intelligent repository has to be equipped with a large and detailed knowledge base, e.g. an ontology [20], which represents the shared conceptual vocabulary and is the basis for the matching mechanism used to suggest SME the most suitable software solutions, given their needs. Moreover, ARNEIS requires a Web-based user interface (UI) enabling ICT companies to describe their software solutions and SME to express their requirements and needs. We claim that both these goals, i.e. building the CRM ontology and designing the system UI, have to be based on a domain analysis that takes into account how users, both from ICT companies and from SME, talk (and think) about CRM activities (see Section 2.2).

The CRM ontology developed for the ARNEIS project is described in [23] and [24], while this paper focuses on the definition of the UI enabling the users to interact with it. In particular, Section 2.1 briefly describes the system architecture and Section 2.2 reports the main results of the domain analysis; Section 3 focus on the user interaction management: it provides a brief survey of the relevant related work in the field (Section 3.1), it discusses the UI design choices and describes the mechanisms supporting UI management (Section 3.2). Section 4 briefly discusses some open issues and concludes the chapter.

## **2 The ARNEIS System**

In this section we will describe the ARNEIS architecture and then we will briefly present the domain analysis that enabled us to define the system knowledge base and the user interface design.

## 2.1 System Architecture

Fig. 1 shows a simplified version of the basic architecture of the ARNEIS system, which is described in details in [19].

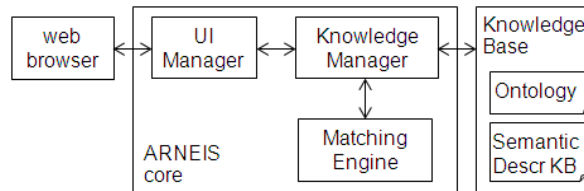


Fig. 1. ARNEIS system architecture (a simplified version taken from [19])

It is a standard three-tiers architecture, where the presentation layer is represented by a standard Web browser.

The application logic layer is represented by the *ARNEIS core*, which includes three main components: the *UI Manager*, the *Knowledge Manager*, and the *Matching Engine*. The *UI Manager* generates the UI taking as input the concepts, properties and relations provided by the *Knowledge Manager* and represented in the *Ontology*; moreover, it collects information provided by the user and forwards it to the *Knowledge Manager*, that builds the semantic representation of software descriptions and SME requirements. The *Knowledge Manager* mediates the interaction between the *UI Manager* and the data layer (its role will be discussed in more detail in Section 3.2). Moreover, it dialogs with the *Matching Engine* (whose description is out of the scope of this chapter), by providing it the semantic representations on the basis of which the *Matching Engine* calculates the correspondence between SME requirements and software house offers.

The data layer contains the *Knowledge Base*, whose main components are the *Ontology*, representing the system semantic competence about the domain (CRM), and the semantic representation of software descriptions and SME requirements (*Semantic Descr KB*). Both the *Ontology* and the *Semantic Descr KB* are written in OWL (<http://www.w3.org/2004/OWL/>). The OWL domain *Ontology* is based on a *CRM Reference Ontology* [20], described in [23] and [24]), which in turn is based on DOLCE (<http://www.loa-cnr.it/DOLCE.html>). The *CRM Reference Ontology* basically models: (1) business activities (e.g., sales, offers, communications, appointments, etc.); (2) business relationships which the company is involved in (e.g., relationships with actual or potential customers); (3) the knowledge that the company has on (or derives from) business activities and relationships; (4) software supporting business activities and knowledge management.

## 2.2 CRM Domain Analysis

In this section we will describe the type of knowledge used by the *UI Manager* to generate the UI, as well as the kind of information that should be elicited from the users in order to build a structured and formal representation of the offered software solutions and of the SME requirements.

The definition of the knowledge and information just mentioned has been based on a detailed domain analysis, aimed at understanding how users (both software houses and SME) talk about CRM. We analyzed two types of information sources: (a) Documents (e.g., brochures, white papers), produced by ICT companies, describing their CRM software tools. (b) Interviews with salesmen from ICT companies, aimed at eliciting the way they describe software solutions for CRM for SME, and managers of SME, aimed at understanding which concepts and terms they use to think about their activities related to customer management.

The analysis of documents and interviews provided us with the following outcomes:

(1) We identified the main concepts, properties and relations involved in the description of CRM activities in SME, which represent the basic requirements for building the CRM Ontology.

(2) We identified a set of *dialog topics*, i.e. concepts that emerged to be the keys of the description of CRM activities. As we will describe in detail in Section 3.2, within the system these concepts are represented as *templates*, i.e. general conceptual patterns that are instantiated with more specific concepts in the different documents/interviews, by means of various linguistic forms. Each template is formally represented as a concept in an Application Ontology: basically, each description of a software solution or a set of SME requirements is an instantiation of a number of such templates (see Section 3.2).

(3) We extracted the linguistic expressions used to refer to dialog topics. The list of such expressions (after a normalization process aimed at eliminating morphological variants) represents a *controlled vocabulary* that is used to provide users with a list of *semantic tags*. Such tags are “semantic” because they are associated with dialog topics, which are represented within the system as concepts in an Application Ontology (see Section 3.2).

Moreover, we proposed to people from SME a short additional interview aimed at eliciting their preferred ways to describe CRM activities. From this additional interview, it emerged that the most natural ways to describe their CRM activities are business processes and natural language. This outcome led us to design a prototype user interface based on Business Process Modeling (BPM) and to start a study to evaluate the feasibility of a Natural Language (NL) UI (see Section 3.2).

### 3 User Interaction in Knowledge-Based Systems

As mentioned in Section 2, the system needs a formal representation of the descriptions of software solutions and of SME technological in order to find the software solutions best matching the SME requirements. Since it is unrealistic to ask users to write descriptions in a formal Semantic Web language, such as OWL, the generation of a user-friendly UI turned out to be an issue of major importance.

### 3.1 Related Work

The importance of the UI to access complex knowledge bases is a topic studied in different research fields. Many authors recognize that there has been an increase of interest in the design of UI for systems based on semantic technologies (see, for instance, the SWUI workshops series: <http://swui.semanticweb.org/>). The main issue to be faced is the fact that the user interacting with a formally encoded knowledge base (e.g., an ontology) should not be exposed to the formal details of the representation: in other words, “semantic technologies should be invisible to users” [5], p. 76.

The most traditional approaches facing this issue aim at supporting user queries that have the goal of searching for information encoded in formal knowledge bases (databases or ontologies). These approaches are mainly based on the translation of keywords (provided by the users) into formal queries (being SQL, or DL, or other formal languages); see, for instance, [29].

Within this thread, some approaches propose graphical (web-based) UI to enable users to easily access knowledge bases; e.g., [27]. Other approaches propose to use NL-based UI; e.g. [6], [10], [14]. In [22], for example, a mechanism is presented for providing a natural language access to databases: a NL query is firstly translated into a query expressed in terms of concepts and relations specified in a domain ontology; in a second step, such a query is translated into a computer language for accessing databases (such as SQL).

In all the mentioned approaches the user input is a query aimed at retrieving information from a (formal) knowledge base. In ARNEIS, the two categories of users interacting with the system have different goals: the goal of a user from a software house is to provide the system with a description of a software solution, while the goal of a representative from a SME is to “explain” the system the SME technological requirements and needs in order to find a suitable software solution. Thus, in ARNEIS, in both cases, the user input is a complex description, based on an existing ontology. In order to elicit this kind of information the system requires a more complex UI with respect to the UI typically provided by the previously mentioned approaches, where the user simply provides the system with a query.

With the broader goal of supporting friendly user interfaces to knowledge-based systems, [7] presents K-Forms, a tool enabling users to define knowledge structure and content through a user-friendly form-based UI; the tool then translate such a knowledge into a formal (RDF/OWL) representation. In [8] K-Search, another tool supporting knowledge search and sharing, is described. Moreover, some authors propose to use Controlled Natural Languages (CNL), i.e., subsets of natural languages that are simpler and less ambiguous [26]. Several CNL have been defined that enable domain experts to specify formal knowledge bases and ontologies, without the need for them to know the formal languages in which the knowledge is ultimately encoded or the knowledge engineering tools. Among these CNL it is worth mentioning CLOnE [18], SOS [13], ACE [17] and Rabbit [21]. As regards ARNEIS, even though SME users should not be forced to learn any “constrained” language, the role of CNL (perhaps for complementing on-line forms in the UI for users from software houses)

should be investigated. In [4] the authors describe a user interface for semantic wiki systems which enables users to enter semantic knowledge by filling in a set of “semantic forms”. The interface renders that knowledge as embedded Rabbit sentences on wiki pages. Their approach exploits a set of “semantic templates”, expressing information about elements of the OWL language (classes, properties, individuals, restrictions, etc.). As explained in the following sections, our approach relies on semantic templates, linked to the domain ontology. However, differently from the above-mentioned work, in ARNEIS templates represent (skeleton of) domain knowledge fragments instead of meta-model elements.

Other works concerned with user interaction with formal representations are authoring tools: in these cases the UI is aimed at enabling the user to design and populate a complex knowledge structure such as an ontology (e.g. [25], Protégé: <http://protege.stanford.edu/>). However, ARNEIS users are not expert in knowledge representation and they could find it difficult to interact with an authoring system, even if provided with a smart (maybe graphical) UI.

Some approaches to Semantic Search study ontology-based Information Retrieval (IR) systems, i.e. systems in which both resources and user queries are represented in a formal semantic language. For example [28] proposes a resource model based on various integrated ontologies, expressed in OWL, enabling resources (documents) representation in terms of ontology elements (i.e., entities and axioms). As discussed by the authors, the issue of how to acquire formal semantic representations of resources is still open. Some steps have been made towards the automatic extraction of such representations from (textual) documents (e.g., [3], [11]), but the results in this field are still not completely satisfactory, especially in those contexts in which complex and detailed concepts should be extracted, as in ARNEIS. As an alternative, “a manual approach can be undertaken” [28], p. 65. In particular, a manual approach makes sense in those cases in which resources are not already available as documents. For instance, in the ARNEIS scenario, a software house may have only simple brochures describing its software products, and thus it seems to be reasonable to provide it with a tool supporting the construction of a formal semantic representation describing their software products, which can be handled as resources by an ontology-based IR system. Our proposal is as a step in this direction.

### **3.2 User Interaction in the ARNEIS System**

In order to face the issue regarding the choice of the most suited user interface for the ARNEIS system, we initially opted for on-line forms, since they are very familiar to Web users; see [7]. In particular, a form-based UI seems to be suited for ICT company users, which are usually skilled enough about the functional and technological aspects of their software products or services. Moreover, such users are probably used to Web-based interaction, often consisting in a sequence of on-line forms. Finally, such users may be motivated to complete a possibly boring task like filling in long on-line forms, since they are describing the software solutions they offer, and this can be viewed as an effective promotion of their products and services.

However, typical users from SME interacting with ARNEIS in order to look for software solutions supporting their CRM activities are not technical experts and they can be in trouble in filling in large on-line forms, possibly requiring technical skills in order to be completed. Thus, on the basis of the results of the additional interview with SME representatives mentioned in Section 2.2, we designed two different types of UI for these users: (a) a UI based on a graphical representation of business processes (BP), based on the idea that business processes are the form in which some companies think about their business and management activities; (b) a Natural Language (NL) UI, in which the SME users can freely express their requirements using natural language texts. These two UI will be available to users from SME as a possible alternative to the form-based UI.

The user interaction supported by all the three types of UI should be based on an analysis of the way in which users (representatives from ICT companies and SME) talk about CRM. For this reason, as we mentioned in Section 2.2., in the domain analysis phase, besides Ontology requirements, we also identified dialog topics, (i.e., key concepts of the descriptions of software solution supporting CRM, and of the descriptions of CRM activities provided by SME), and the linguistic expressions used to refer to such dialog topics (see Section 2.2).

We will present an example to clarify how dialog topics are exploited in the system. One of the templates representing dialog topics we extracted from our analysis corresponds to the concept of “(dynamic) acquisition of data from (another) enterprise application”<sup>1</sup>, which means that the described CRM software application can acquire data by directly communicating with another application. Such a general concept is instantiated in different documents/interviews with more specific concepts by various linguistic expressions: specific instances of this template include more specific concepts in place of “data” and in place of “enterprise application”; for example “real time acquisition of customers info from Ms Outlook”, or “(management of) product records acquired from ERP software”.

Within ARNEIS templates are represented as ontological concepts belonging to an *Application Ontology* [20] linked to the CRM Ontology.

A template represents a structured concept in which there are some *slots* (i.e., *variables*). By default, such slots are filled in by generic concepts (e.g. “data”), that can be replaced by more specific concepts (e.g. “product data”).

Fig. 2 shows the OWL logical representation (generated by Protégé) of the Application Ontology class representing the template corresponding to the concept of “(dynamic) acquisition of data from (another) enterprise application”.

Slots within a template are identified by adding to each concept that represents a default slot filler the expression *and S<sub>i</sub>*, where *S<sub>i</sub>* is a unique identifier (automatically generated) for that slot. Thus, for example, in Fig. 2 the expression (*CRM\_element and S<sub>1</sub>*) identifies a slot (labeled *S<sub>1</sub>*) filled in, by default, by the *CRM\_element* class (intuitively speaking, *CRM\_element* represents all the items that are typically involved in CRM: customers, products, orders, sales, and so on). In order to express

---

<sup>1</sup> Within the ARNEIS system, templates do not have a linguistic form, but only a formal one (in OWL). For the sake of readability, here we provide also a rough linguistic “translation”.



the acquisition of data about customers (e.g., “customers info”), in a template instantiation, the class *CRM\_element* is replaced by a more specific one, i.e. *Customer* (subclass of *CRM\_element*) and the expression *and S<sub>1</sub>* is deleted; in order to express the acquisition of data about products (e.g., “product records”), *CRM\_element* is replaced by *Product\_or\_service* (subclass of *CRM\_element*).

It is worth stressing that the expression *and S<sub>i</sub>* is used merely as a label, to identify slots within a template. This solution preserves the syntactic correctness of the OWL representation, without producing any odd semantic result, since not instantiated templates are never used in any form of semantic reasoning.

Templates based on dialog topics are the basis of the mechanism underlying all three UI we are going to describe in the following, i.e.: (a) the form-based UI devoted to the acquisition of descriptions of software solutions by ICT companies and available also to SME representatives; (b) the BPM-based UI available to SME users in order to provide ARNEIS with a description of their CRM activities in the form of business processes; (c) the study we carried on in order to evaluate the feasibility of a NL-based UI, enabling SME users to express their requirements about CRM support using natural language texts.

```

T3
Software_process
and has_input some (Digital_encoding
and encodes some (Information_object
and refers_to some (CRM_Element and S1))
and has_message_role some (Communication_to_software_processes
and sees_as_encoder some (Software_process and execution_of some (Application_software and S2))
and sees_as_sender some (Software_process and execution_of some (Application_software and S3))
and completely_realized_by some Information_realization_by_physical_bits
and output_of some (Software_process
and execution_of some (Application_software and S4))
and has_output some (part_of some (Digital_encoding
and encodes some (Information_in_CRM_system and S5))
and has_decoder_role some (Communication_to_software_processes
and sees_as_encoder some (Software_process and execution_of some (Application_software and S6))
and sees_as_message some (Digital_encoding
and encodes some (Information_object and refers_to some (CRM_Element and S7))
and completely_realized_by some Information_realization_by_physical_bits
and output_of some (Software_process and execution_of some (Application_software and S8))
and sees_as_sender some (Software_process
and execution_of some (Application_software and S9))
and has_receiver_role some (Communication_to_software_processes
and sees_as_encoder some (Software_process and execution_of some (Application_software and S10))
and sees_as_message some (Digital_encoding
and encodes some (Information_object and refers_to some (CRM_Element and S11))
and completely_realized_by some Information_realization_by_physical_bits
and output_of some (Software_process and execution_of some (Application_software and S12))
and sees_as_sender some (Software_process and execution_of some (Application_software and S13))

```

**Fig. 2.** “(Dynamic) acquisition of data from (another) enterprise application” template

*Form-based UI.* For each template, the UI Manager generates a set of Web-forms, aimed at eliciting the information needed for filling in the slots.<sup>2</sup>

In order to enable the UI Manager to generate the Web forms, the Knowledge Manager performs two (nested) steps:

<sup>2</sup> Since, typically, the knowledge on the basis of which such forms are generated does not change in time, Web forms are pre-compiled off-line; they are re-generated only in case the knowledge bases are modified.

(1) For each template in the Application Ontology, it extracts all the slots by looking for the expressions of the form  $(C \text{ and } S_i)$ , where  $C$  is a named class of the reference Ontology.

(2) For each slot (identified by  $S_i$ ), it extracts from the Ontology all the subclasses of  $C$  and provides the UI Manager with this information.

The UI Manager, in turn, generates a Web form in order to ask the user which subclass of  $C$  (if any) she wants to consider. For example, Fig. 3 shows the form asking the user to select the subclasses of  $CRM\_element$ .

Would you like to acquire data concerning...

<input type="radio"/> Sales_agents	<input type="radio"/> Estimates
<input checked="" type="radio"/> Customers <a href="#">[view more]</a>	<input type="radio"/> Sales
<input type="radio"/> Communications <a href="#">[view more]</a>	<input type="radio"/> Orders
<input type="radio"/> Appointments	<input type="radio"/> Product_or_service offer

**Fig. 3.** Web form for filling in the first slot of the template corresponding to “(dynamic) acquisition of data from (another) enterprise application”

The user selects the desired subclass ( $C_x$ ) and the UI Manager sends this information back to the Knowledge Manager, that substitutes  $C_x$  in place of  $(C \text{ and } S_i)$  within the corresponding template slot. The result (that refers to the example presented above) is shown in Fig. 4.

```

Software_process
and has_input some (Digital_encoding
and encodes some (Information_object
and refers_to some Customer)
...
    
```

**Fig. 4.** Part of the instantiated template after the user answer

There are some issues to be faced in order to make this mechanism work: we will briefly comment them in the following.

The knowledge engineer who configures ARNEIS on a new domain is in charge of defining the Application Ontology containing the templates. When doing this, she identifies the slots and, for each slot, she provides a natural language question that will be used by the UI Manager to generate the form referring to that slot (e.g., “Would you like to acquire data concerning...” in Fig. 3). The link between the slot  $S_i$  within the template  $T_j$  ( $T_j.S_i$ ) and the corresponding question is stored in a *Configuration KB* which is accessed by the UI Manager during the form generation process.

When filling in the forms, the user may want to specify a more specific class (e.g., she wants to acquire data about golden customers, which are represented, within the Ontology, by the class *Golden\_customer*, subclass of *Customer*).When extracting from the Ontology all the subclasses of  $C$  (step (2) mentioned above), the Knowledge Manager actually extracts not only the direct subclasses of  $C$ , but the whole subtree. The UI Manager, on the basis of this information, includes a link to enable the user to

optionally expand the subtree (“view more” links in Fig. 3): in this way she is enabled to select any (direct or indirect) subclass she is interested in.

However, given the complexity of the Ontology, listing all the (named) subclasses of a given class  $C$  could result in a too long list, containing concepts that are relevant from the formal point of view, but not from the user perspective. For this reason, the knowledge engineer can add to the Configuration KB, for each slot  $T_j.S_i$ , a list of the (direct and indirect) subclasses of  $C$  that should be asked to the user, as shown for example in (1)<sup>3</sup>:

$$T_3.S_1 \rightarrow \{\text{Sales\_agent, Customer, ...}\} . \quad (1)$$

The final issue to discuss concerns the dependencies between slots. It is quite common that the suited value for filling in a slot depends on the value assigned to another slot (of the same template). In the example above (see Fig. 2), if we fill in slot  $S_j$  with *Customer* (meaning that we want to acquire information about customers) slot  $S_5$  (representing the knowledge structure that is modified by the data acquisition) has to be filled in with *Customer\_database*. In order to take into account possible dependencies between slot fillers, the Configuration KB contains if-then rules representing such dependencies, as shown for instance in (2):

$$\text{IF } T_3.S_1 = \text{Customer THEN } T_3.S_5 = \text{Customer\_database.} \quad (2)$$

These dependency rules enable the UI Manager to avoid asking the user useless questions: after having asked the filler for slot  $S_j$ , slots  $S_5$  will be filled in automatically.

When the user has completed all the forms proposed by the ARNEIS system, the instantiated templates are saved in the Semantic Descr KB (see Section 2.1): the set of such instantiated templates is the OWL representation of the software solution supporting CRM, proposed by an ICT company, or the set of user requirements, provided by a SME.

*BPM-based UI.* SME users, besides the UI based on on-line forms, can choose the BPM-based UI. In this last case, the user can draw a diagram representing the CRM business processes of her SME, using the tool she prefers. The only requirements are that the output diagram must be compliant with BPM Notation (<http://www.bpmn.org/>) and must be exportable in XPD L (<http://www.wfmc.org/xpdl.html>). When the diagram representing the CRM process is ready and saved as an XPD L file, the user can upload it on the ARNEIS server. The UI Manager acquires the XPD L file and, on the basis of the information it contains, it “re-draws” the diagram in a Web page, as shown in Fig. 5. When the user clicks on an activity, she accesses the tagging page, shown in Fig. 6, that presents her the list of available *semantic tags* (see section 2.2), among which she can select those to be associated to the activity in focus. For instance, in the page shown in Fig. 5 the user selected the “customer contact” (“contatto cliente”) activity and, in the following page (Fig. 6), she associated the semantic tag “multichannel communication” to this activity.

<sup>3</sup> The Configuration KB is an XML document; here we show its content in a more readable way.

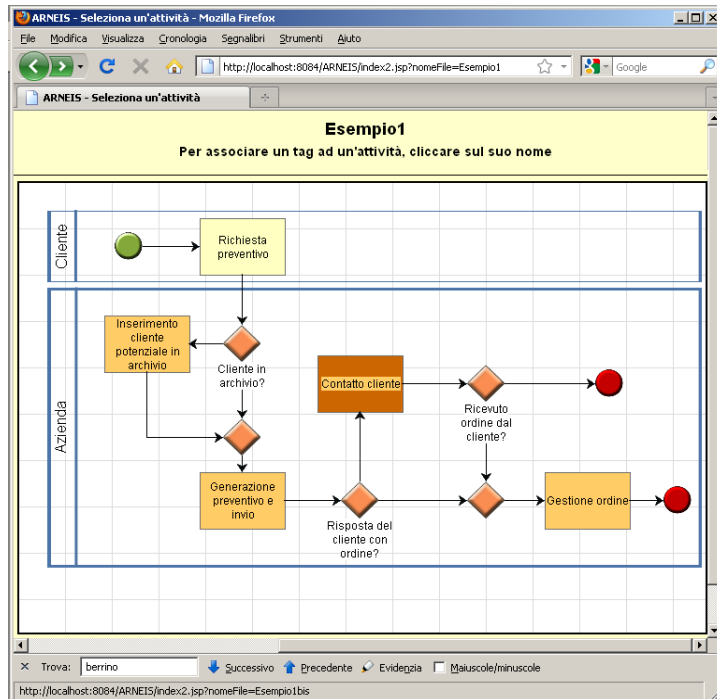


Fig. 5. The BPM-based UI: selecting an activity

Each semantic tag is associated to one or more (possibly instantiated) templates. As a consequence, the user tagging activity corresponds to template selection and thus results in a set of template instances that represent the semantic (OWL) representation of the SME requirements concerning CRM. The OWL representation of the SME requirements is then compared with the software descriptions stored in the Software Repository by the Matching Engine (see Section 2.1).

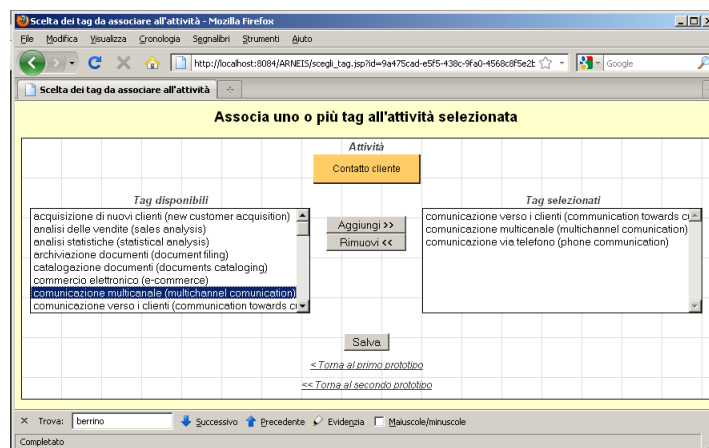


Fig. 6. The BPM-based UI: tagging an activity

*NL-based UI.* The study to evaluate the feasibility of a Natural Language (NL) UI started from an analysis of the type of texts that users could provide to the system. We thus asked SME representatives who indicated NL as the preferred interaction modality (see Section 2.2) to provide us with textual descriptions of their CRM activities and needs. On the basis of these texts, and taking into account the various NLP approaches that emerged from a preliminary survey, we advanced the hypothesis of exploiting Information Extraction (IE) techniques [9], to extract from NL texts the information needed by ARNEIS.

The UI we designed contains a number of questions referring to CRM activities (e.g., “Which are the main activities for Customer Relationship Management, within your company?”). The user can answer these questions by inserting (or uploading) free texts, which are then sent to the ARNEIS interpretation module (which is an additional module within the Knowledge Manager). Such a module takes as input the user answers, extracts relevant information and builds the OWL representation of the description of CRM activities, representing SME requirements, to be matched with the OWL representation of CRM software descriptions.

The goal of the IE phase is to instantiate the templates representing dialog topics. This phase is based on semantic rules that extract from the text the information needed to fill in the template slots. For instance, the IE module contains a semantic rule that matches various linguistic patterns that refer to “(dynamic) acquisition of data from (another) enterprise application” (e.g., “real time acquisition of customers info from Ms Outlook”, or “management of product records acquired from ERP software”); the results of the application of such a rule is an instantiated template (see the fragment shown in Fig. 4).

## 4 Discussion and Conclusions

In this paper, we described the management of the user interaction within the ARNEIS system, an “intelligent” Web-based repository of descriptions of software solutions. ARNEIS enables ICT companies to upload a description of their software solutions supporting CRM, and SME to find software products or services supporting their CRM activities. The functionality of ARNEIS is based on a semantic representation of the domain knowledge, which represents the shared vocabulary to express both software descriptions and SME requirements. In particular, we described the management of the interaction with users, which is based on the instantiation of templates that represents *dialog topics*. Different types of user interfaces (i.e., form-based, BPM-based, and NL-based) enable software houses to upload the semantic representation of the description of their products or services and SME to express their requirements concerning CRM support.

The approach we proposed within the ARNEIS scenario suggests a general solution to face the issue of how to build formal representations of resources in a those systems based on semantic technologies. However, our approach is strongly based on a domain analysis that takes into account how users talk about their business activities and the software applications that could support such activities. In fact, the mechanism described in this chapter relies on templates stored in an Application Ontology and on a Configuration KB, that have to be built by the knowledge engineer who configures the system on a domain (e.g., CRM). Moreover, the feasibility study we carried on the NL-based UI, showed that in such a type of UI there is an additional effort, i.e., the

definition of the semantic rules devoted to the extraction of the information needed to instantiate the templates. These tasks could represent a considerable knowledge acquisition effort. However, the definition of templates and Configuration KB, as well as the definition of extraction rules for the NL-based UI, are typically done once and usually do not require frequent updates. Moreover, the Configuration KB definition can be easily supported by a user-friendly tool that provides the knowledge engineer with simple mechanisms supported by system defaults. The development of such a tool, supporting the knowledge engineer in the system configuration on new domains, is an issue of major importance, given that the knowledge acquisition effort is a key challenge of knowledge-based systems like ARNEIS.

## References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: *Web Services*. Springer, Heidelberg (2004)
2. Antoniou, G., Van Harmelen, F.: *A Semantic Web Primer*. The MIT Press, Cambridge (2004)
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open Information Extraction from the Web. In: 20th International Joint Conference on Artificial Intelligence, pp. 2670–2676. Morgan Kaufmann, San Francisco (2007)
4. Bao, J., Smart, P.R., Braines, D., Shadbolt, N.R.: A Controlled Natural Language Interface for Semantic Media Wiki Using the Rabbit Language. In: Workshop on Controlled Natural Language. CEUR, Aachen (2009)
5. Benjamins, V.R.: Near-Term Prospects for Semantic Technologies. *IEEE Intelligent Systems* 23(1), 76–88 (2008)
6. Bernstein, A., Kaufmann, E.: GINO - A Guided Input Natural Language Ontology Editor. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 144–157. Springer, Heidelberg (2006)
7. Bhagdev, R., Chakravarthy, A., Chapman, S., Ciravegna, F., Lanfranchi, V.: Creating and Using Organisational Semantic Webs in Large Networked Organisations. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) *ISWC 2008*. LNCS, vol. 5318, pp. 723–736. Springer, Heidelberg (2008)
8. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid Search: Effectively Combining Keywords and Semantic Searches. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 554–568. Springer, Heidelberg (2008)
9. Cunningham, H.: Automatic Information Extraction. In: Brown, K. (ed.) *Encyclopedia of Language and Linguistics*, 2nd edn., pp. 665–677. Elsevier, Amsterdam (2005)
10. Cimiano, P., Haase, P., Heizmann, J., Mantel, M.: ORAKEL: A Portable Natural Language Interface to Knowledge Bases. *Data & Knowledge Engineering* 65(2), 325–354 (2008)
11. Cimiano, P., Mädche, A., Staab, S., Völker, J.: Ontology Learning. In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*, 2nd edn., pp. 245–267. Springer, Heidelberg (2009)
12. Creeger, M.: CTO Roundtable: Cloud Computing. *Communications of the ACM* 52(8), 50–65 (2009)
13. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL Syntax – towards a Controlled Natural Language Syntax for OWL 1.1. In: Third International Workshop on OWL: Experiences and Directions. CEUR, Aachen (2007)

14. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) *ESWC 2010, Part I. LNCS*, vol. 6088, pp. 106–120. Springer, Heidelberg (2010)
15. Dikaiakos, M.D., Pallis, G., Katsaros, D., Mehra, P., Vakali, A.: *Cloud Computing. Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing* 13(5), 10–13 (2009)
16. Freeland, J.: *The Ultimate CRM Handbook*. McGraw-Hill, New York (2005)
17. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) *Reasoning Web 2008. LNCS*, vol. 5224, pp. 104–124. Springer, Heidelberg (2008)
18. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: *CLOnE: Controlled Language for Ontology Editing*. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007. LNCS*, vol. 4825, pp. 142–155. Springer, Heidelberg (2007)
19. Goy, A., Magro, D., Prato, F.: ARNEIS: A Web-based Intelligent Repository of ICT Solutions for E-business. In: *10th Int. ACM Conference on Information Integration and Web-Based Application & Services*, pp. 403–406. ACM Press, New York (2008)
20. Guarino, N.: Understanding, Building and Using Ontologies. *Int. Journal of Human-Computer Studies* 46, 293–310 (1997)
21. Hart, G., Johnson, M., Dolbear, C.: Rabbit: Developing a Control Natural Language for Authoring Ontologies. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008. LNCS*, vol. 5021, pp. 348–360. Springer, Heidelberg (2008)
22. Lesmo, L., Robaldo, L.: From Natural Language to Databases via Ontologies. In: *5th International Conference on Language Resources and Evaluation*, pp. 1460–1465 (2006)
23. Magro, D., Goy, A.: The Business Knowledge for Customer Relationship Management: an Ontological Perspective. In: *1st Int. ACM Workshop on Ontology-Supported Business Intelligence*. ACM Press, New York (2008)
24. Magro, D., Goy, A.: Towards a first Ontology for Customer Relationship Management. In: *1st IEEE/ACM Workshop on Applied Ontologies in Distributed Systems*, pp. 637–643. ACM Press, New York (2008)
25. Rector, A.L., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: Designing User Interfaces to Minimise Common Errors in Ontology Development: The CO-ODE and HyOntUse Projects. In: *3rd UK E-Science All Hands Meeting*, pp. 493–499. EPSRC, Swindon (2004)
26. Schwitter, R.: Controlled Natural Languages for Knowledge Representation. In: *23rd International Conference on Computational Linguistics*, pp. 1113–1121 (2010)
27. Thoméré, J., Barker, K., Chaudhri, V., Clark, P., Eriksen, M., Mishra, S., Portr, B., Rodriguez, A.: A Web-based Ontology Browsing and Editing System. In: *18th AAAI National Conference on Artificial Intelligence*, pp. 927–934. AAAI Press, Menlo Park (2002)
28. Tran, T., Bloehdorn, S., Cimiano, P., Haase, P.: Expressive Resource Descriptions for Ontology-Based Information Retrieval. In: *1st International Conference on the Theory of Information Retrieval*, pp. 55–68. Alma Mater Foundation for Information Society, Budapest (2007)
29. Tran, T., Cimiano, P., Rudolph, S., Studer, R.: Ontology-Based Interpretation of Keywords for Semantic Search. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *ASWC 2007 and ISWC 2007. LNCS*, vol. 4825, pp. 523–536. Springer, Heidelberg (2007)