Semantic Web Fostering Enterprise 2.0

Nicola Capuano, Matteo Gaeta, Francesco Orciuoli and Pierluigi Ritrovato
Dipartimento di Ingegneria dell’Informazione e Matematica Applicata (DIIMA)
University of Salerno
84084 Fisciano (SA), Italy
Email: {capuano, gaeta, orciuoli, ritrovato}@diima.unisa.it

Abstract—The term Enterprise 2.0 applies to the use of Web 2.0 technologies as a support for business activities within the organizations. These technologies are exploited to foster inter-persons collaboration, information exchange and knowledge sharing, also outside the organization, to establish relationships based on conversational modalities rather than on traditional business communication. The vision of Enterprise 2.0 places a high value on the importance of social networks inside and outside the organization stimulating flexibility, adaptability and innovation between workers, managers, customers, suppliers and consultants. The integration between the Web 2.0 tools with traditional enterprise software, the aggregation of organization inner data with external data and the choice of adequate knowledge representations are critical aspects to be faced in order to further the growth of smart applications in the Enterprise 2.0 context. In this work we propose an approach, based on Semantic Web techniques, to relax the aforementioned critical issues.

I. INTRODUCTION

In [1], the authors analyse the process by which a team creates a new product and observe that: (i) team members meet to share their knowledge, mostly tacit knowledge (e.g. insights into customer’s needs, information about new technologies, personal skills, etc.); (ii) team members create a new product concept; (iii) team members plus outside participants try to justify the concept using market studies, benchmarking, company strategy, etc., and (iv) team members transform a chosen concept into a prototype.

The aforementioned process steps lead to the conclusion that the knowledge creation is a social as well as an individual process.

From the previous consideration a new trend about the knowledge enablement (set of activities that positively affect the knowledge creation) in the organizations, is emerging with the name of Enterprise 2.0 coined by Andrew P. McAfee, who defines it as: The use of Emergent Social Software Platforms within companies, or between companies and their partners or customers [4].

Several reasons are behind the sudden development of Enterprise 2.0. Some of the most common means of staff communication have become overused: the workers have to browse through hundreds of emails per day, risking to overlook important information. Yet, sometimes the knowledge shared between employees is not accessible to others: only a small percentage of most people’s output winds up on a common platform. So the rapid expansion of Web 2.0 and social networks inspired many researchers to use the same techniques in working environments. McAfee also illustrates one of the most important benefits of making staff interact in a social network: it increases the number of weak links among people. Two people who have a weak link with each other, i.e. have not a long standing friendship, probably have not many friends in common, so that each one may likely use the other as a bridge to get in contact with new people and new ideas. Furthermore, McAfee identifies six SLATES components for Enterprise 2.0 technologies: Search (providing the same search mode to look for all content produced in the enterprise environment), Links (showing links between documents, blog posts and wiki articles that concern the same topics or similar topics), Authoring (enabling persons to share content by writing blog posts, comments, articles, etc.), Tags (offering a new and collaborative way to categorize content by exploiting folksonomies), Extensions (supporting users with recommendations, suggestions, etc.) and Signals (providing tools to automatically receive updates from news feeds and to aggregate them).

At present, the Enterprise 2.0 environments are not defined by the use of single applications but by the integration of several applications coming from Web 2.0 world (e.g. Wikis, Blogs, Social Network Tools, etc.), traditional enterprise world (e.g. CRM, ERP, CMS), Internet (e.g. LinkedIn, Facebook, etc.) and so on. So, in order to enable the knowledge creation and sharing scenarios inside Enterprise 2.0, exploiting the above technological components, we need an approach to model, represent and harmonize data produced by different applications. Our proposal consists in the definition of a framework based on Semantic Web techniques that aims to support the creation of Enterprise 2.0 environments [6]. In the last years, since many Semantic Web community efforts have been directed toward finding a way to use ontologies [5] [12] to express concepts in the Web 2.0 environment, this stands to reason that the same techniques could be applied also to the Enterprise 2.0. The Semantic Web techniques provide an answer to the request for the evolution from a chaotic state generated by the production of great volumes of data coming in online activities to a state characterized by information sharing, cooperation and collaboration. Modeling data by using data models (e.g. RDF) and established vocabularies (e.g. FOAF, SIOC, SKOS, etc.) provides a useful framework to distill communal knowledge out of a chaotic mess of information [2].

In section 2, we describe the proposed framework, which in section 3, is instantiated for a plausible scenario. Conclusions
and future works are considered in section 4.

II. A SEMANTIC WEB FRAMEWORK FOR ENTERPRISE 2.0

In this section we describe a Framework, based on Semantic Web technologies, that can be instantiated in order to support the SLATES components of the Enterprise 2.0 environments. The main idea of the proposed Framework is to exploit RDF/RDFS/OWL languages, and their related methodologies and technologies to model, represent, harmonize and access data produced by different and heterogeneous applications. The main advantages of this approach are: (i) the exploitation of a unique data model (RDF) to simplify storage and retrieval operations, (ii) the use of RDFS/OWL to describe data and further inference semantically and to obtain new facts from the ones already stated and (iii) the possibility to link internally produced data to the Linked Data on the Web [11] [10]. The Framework we introduce is composed of a set of existing Semantic Web vocabularies used to model specific and relevant aspects of Enterprise 2.0 and some architectural solutions to support data representation, storage, retrieval and inference.

A. The Proposed Architectural Solutions

Let us start with the description of the Framework by illustrating the layered architectural sketch depicted in Fig. 1. The upper layer (Applications Layer) consists in the set of applications executing both traditional and Web 2.0 scenarios in the enterprise context. At this level, we can note the introduction of a new functional block, namely Personal Work Learning Environment (PWLE). The paradigm of PWLE is defined by Tony Karrer 1 as the set of methodologies and tools useful to support the daily role of the knowledge worker with the aim of acquiring, processing and sharing new information and knowledge.

We based on the Karrer’s idea of PWLE and transform it into an architectural solution. From the functional viewpoint, we look at the PWLE as a personal Web 2.0 environment that aims to simplify the execution of working and learning tasks by providing a centralized and personalized (for a single worker) entry point to all tools and data the knowledge workers need. The PWLE allows workers to manage their knowledge, e.g. profiles, contacts, bookmarks, tasks, news, etc., and can execute some complex scenarios like expert finding, interest group composition, content search, semantic mashup, relationships network navigation, link finding and so on.

On the bottom of the Applications Layer we find the Data Representation Layer in which the data produced at the upper layer are modeled with some existing Semantic Web vocabularies represented in RDF and harmonized together (overlapping between different data produced at the upper layer is possible) using RDFS, OWL, OWL2, etc.

The lower layer (Data Storage Layer) collects all data converted in RDF and is realized by the use of an RDF Storage System (e.g. Virtuoso, OWLIM, Jena-TDB, Talis Platform, etc.). Once all data and their semantics are integrated and stored in the same place we can execute queries (e.g. using SPARQL, RQL, etc.) to perform search operations, to apply rules (e.g. using SWRL, RuleML, Jena Rules, etc.) to infer new facts from the ones previously stated and to navigate the RDF graph in order to support sophisticated visual information retrieval tools.

Furthermore, two additional layers are provided: Adaptation Layer and Data Access Layer. The first one is needed in order to translate data represented in application-specific formats using a particular RDF vocabulary. The second one is realized using of a RDF Query and Rule Engine performing query execution and inference on data.

As for the design of the architecture formerly described in detail, two main critical tasks have been identified: (i) modeling enterprise data with existing vocabularies and converting them into RDF and (ii) defining a coherent and extensible integration model among all used vocabularies. The “modeling part” of the first problem is treated in more details in the following sub-section, while the “representation part” is commonly solved using tools like RDFGateway to transform relational data, XML documents, etc. into RDF triples, exporter plug-ins API[9] or custom solutions. Conversely, the second problem can be solved through a systematic harmonization of the different vocabularies [3] by using defined properties like owl:equivalentClass, rdfs:subPropertyOf, owl:equivalentProperty, rdfs:subClassOf, owl:sameAs, rdfs:seeAlso etc. and existing guidelines and tutorials [10]. From the implementation viewpoint, we are using (in our prototype) RDF Storage and Query System called TDB 2 by using Jena Framework 3. Jena provides the execution of SPARQL queries, OWL Lite reasoning and inference operations (by using a rule-based engine).

B. Selection and Use of Existing Semantic Web Vocabularies

We will only focus on a few critical aspects emerging from the Enterprise 2.0 vision: Folksonomies and Social Bookmarking, Collaboration, Document Management, Rating & Commenting, Human Resource Management and Social Networks.

1) Folksonomies and Social Bookmarking: The workers should be allowed to enter free-form tags to describe and categorize the content they are creating, editing or simply viewing. This categorization makes it easier for the content to be found by others looking for the same information through search engines and by using facilities like Tag Clouds. As long as each individual or team is given the tools to tag their own documents or wiki pages in order to make them easier to browse for themselves, then this practice will gain some traction throughout the organization. Once many people begin to tag things for their own benefit, a community Folksonomy

---

1http://elearningtech.blogspot.com/2007/05/personal-learning-environment-plc.html
2http://relfinder.dbpedia.org/app.swf
3http://www.intellidimension.com/
4http://openjena.org/wiki/TDB
5http://jena.sourceforge.net/

1088
Fig. 1. Semantic Web Layer.

will emerge that is no doubt be useful as well. It will help to spread knowledge across teams and departments. The workers can store, manage, and share their bookmarks to Web content and apply a series of tags with each bookmark to make it easier to find at a later point. Not only has social bookmarking made it much easier to manage catalogs of thousands of bookmarks, but it has also facilitated bookmarks sharing with colleagues via RSS feeds.

To model and represent data coming from Folksonomies the SCOT6 (Social Semantic Cloud of Tags) specification can be used. SCOT aims to describe the structure and the semantics of tagging data and to offer social interoperability for sharing and reusing tag data and representing social relations amongst individuals across different sources. The scot:Tag class and the scot:TagCloud class are respectively used to manage tags and group them into tag clouds. A tag (individual of type scot:Tag) can be associated to a specific resource by using the scot:tagOf property. SCOT also enables the modeling of some aspects regarding "who" uses a specific tag. In fact, the property scot:usedBy links a tag to a specific user. Often, especially in a Social Bookmarking environment, it is important to track data of the tagging operation and not only of the used tag. For instance, we would like to know all the resources tagged with "OWL" and how many people have tagged a given resource with the "OWL" tag with respect to other tags (and their occurrences) attached to the same resource. In order to perform this task we need to exploit the Richard Newman’s Ontology7. In particular, the instances of the tags:Tagging make associations between tags (instances of scot:Tag), tagged resources (URLs) and users.

SCOT can be also integrated with the MOAT8 (Meaning Of A Tag) ontology that provides a mechanism to enrich data regarding tags. The MOAT ontology differs from previous tagging ontologies in that the tagging relationship includes a new concept: meaning. Each tag has a set of meanings, described by a URI (usually taken from lexical resources like DBpedia9) and a set of users who associate a certain meaning to a given tag. This aspect is very important to handle situations where the same word, representing a tag, has several meanings and where more than one tag have the same meaning. The information provided by MOAT can be used to improve the user’s experiences when disambiguation is needed.

2) Collaboration, Documents, Rating and Commenting: The starting point for most organizations, that look at investing in an Enterprise 2.0 strategy, is the creation of a Corporate Blog. Typically, a single blog maintained by one person is published on the corporate Intranet. Other employees can leave comments on posts and subscribe to the blog’s RSS feed. Successively, the blog is expanded to allow for multiple contributors. The third step in the blog evolution is to publish a blog on the company’s Web site that can be viewed by all customers and potential new customers alike, allowing to receive positive and negative feedbacks and find out the exact thoughts of the clients. Furthermore, Enterprise Wikis are becoming a very popular way of managing documents and information inside companies and are an important aspect of Enterprise 2.0. They allow true collaboration on documents as anyone enabled to access the page can edit it, making any relevant changes or posting updated content. Also Web Conferencing, and tools and services in this area are growing in popularity.

The SIOC10 (Semantically-Interlinked Online Communities) ontology is an RDF-based schema which describes the main concepts found in on-line communities. While there are many classes and properties in SIOC, the main notion is that sioc:Users create sioc:Posts that are contained in sioc:Forums that are hosted on sioc:Sites. One

---

6http://scot-project.org/scot/index.html
7http://www.holygoat.co.uk/owl/redwood/0.1/tags/
8http://moat-project.org/
9http://dbpedia.org
10http://sioc-project.org/
of the main SIOC properties of interest is sioc:hasTag defining a tag that a particular discussion post is associated to. The sioc:content and sioc:hasTarget property (with its inverse property sioc:reply_of). The first one is used to report the real content in order to perform better search operations. The second one is useful to track a discussion thread with posts and associated replies. An instance of sioc:Post can be tagged also by using the scot:hasTag property, or conversely by using the scot:tagOf property with domain scot:Tag and range sioc:Item (note that sioc:Post is a subclass of sioc:Item). Both the tagging solutions provide a way to synergically use SIOC and SCOT schemas. The information architecture provided by SIOC is useful to model data coming from Blogs, Wikis, Mailing Lists, Instant Messaging sessions, and so on. The great advantage of using SIOC is that we can organize on-line sessions data, produced by different applications, adopting the same model and the same representation in order to improve search operations and to foster knowledge-based applications. SIOC is also used in synergy with the Richard Newman’s Ontology given that tags:Tagging instances can be associated with sioc:User instances.

SIOC does not offer support to handle documents produced by traditional Document Management Systems. To fill this lack we could exploit a RDF representation of DC (Dublin Core Metadata Initiative). DC provides simple standards to facilitate the finding, sharing and management of information. The prefix dc and the prefix dcterms refer to the Dublin Core vocabulary. Note that dcterms:subject (used to apply tags to the documents) is a superclass of sioc:topic that is a superclass of scot:hasTag. So, if we perform a search for all resources in the domain of dcterms:subject we find both documents (described in DC) and items (described in SIOC).

Knowing “who” makes something in the Enterprise 2.0 environment is a very important aspect with respect to our goal. So, we need a way to link persons to the posts and documents they wrote. The property foaf:maker, coming from the FOAF vocabulary, allows us to establish a link for content and tags with people’s profiles (that will be modeled using FOAF as shown later).

The last issue regarding content description we would like to underline is the handling of Rating & Commenting. Making the company content available, by allowing users to rate and comment them and allowing customers to post feedbacks and opinions. This process creates a communication channel that can be used to get to know the customers better and helps build relationships where customers are satisfied that the company is glad to listen to what they have to say. Review RDF is a domain specific vocabulary used to describe the main properties of a review in RDF. The most important properties are createdOn, hasReview, maxRating, minRating, rating, reviewer and text. Rating is also useful to enable search engines to emphasize valued content and improve the reputation of workers/authors.

3) Human Resource Management and Social Networks: Modeling human resources information in the organization context is a very critical task, especially when we need to model internal and external relationships. Firstly, we need a schema to model a person and his/her relationships in a virtual environment. The needed schema is provided by the FOAF (Friend-of-a-Friend) ontology, that is useful to create machine-understandable information/metadata regarding people, groups, organizations and other related concepts. One of the most used properties of the FOAF ontology is the foaf:knows property, that represents a simple way to create social networks through the addition of “knows” relationships for each individual a person knows. The FOAF ontology has several relevant properties for representing, for instance, people interests or the artifacts they create. In the enterprise context we have to model, represent and store not only information about employees (e.g. workers, managers, etc.) but also information about external people like customers, consultants and suppliers. foaf:knows generates a social graph we can navigate. We can also state some rules like if “A knows B” and “B knows C” then assert “A can trust to C” in order to infer new facts (A and C are not previously connected in the social network) from a path made of foaf:knows properties.

Other important classes, we can find in FOAF, are foaf:Organization and Group respectively used to model an organization (e.g. a company, a university, etc.) and a group of persons (e.g. a project team, an interest group, etc.). In order to effectively build project teams, groups, etc. we have to know the employees’ personalities. FOAF provides the foaf:myersBriggs property that can be used to model the personality of an individual using the MBTI classification.

Unfortunately, FOAF does not cover all the relevant aspects regarding people within organizations. FOAF has to be complemented by the use of other specific vocabularies. Supposing we need something to model addresses, telephone numbers, etc. The answer to this request is vCard RDF. vCard is a file format standard for electronic business cards. vCards are often attached to e-mail messages, but can be exchanged in other ways, for instance on the Web and through the use of other software (e.g. Microsoft Outlook, etc.) in order to manage people’s contacts. The vCard vocabulary, as we are going to see in the next section, provides a great opportunity to exploit the mobile device applications in order to enrich Enterprise 2.0 data permitting new knowledge sharing scenarios.

Additionally, FOAF does not cover aspects like recruiting, staffing, system provisioning, benefits, payroll, assessments, background check, time and performance, that are supported by HR-XML. HR-XML also provides the properties useful.
to describe skills and competencies of each employee. In order to define a complete Semantic Web model for the Enterprise 2.0 we have to inject into FOAF new classes and properties that might represent the information existing in HR-XML. ResumeRDF\textsuperscript{17} is an ontology developed in order to express on the Semantic Web the information contained in a resume, such as business and academic experience, skills, publications, certifications, etc. For instance, ResumeRDF provides \texttt{cv:CV}, \texttt{cv:Person} and \texttt{cv:Skill} that are classes we can use to model skills and link them to FOAF profiles.

Furthermore, one of the most important innovation in the Enterprise 2.0 environment regards Social Networking. An important aspect of social network services is the way they allow people to connect with others, building a network of friends and acquaintances. Many of these connections may work well for companies, in fact employee’s friendship (existing or potential clients) could result, for instance, in a sales lead or in finding expertise to carry out a task. The Social Networks (but also Social Bookmarking, Rating and Commenting, etc.) become features used to perform the evolution of typical enterprise systems like Customer Relationship Management (CRM) or to enrich information coming from systems like Human Resource Management (HRM). CRM is a strategy focused on evolving customer relationships and is directed to understand, anticipate and respond to the needs of enterprise’s current and potential customers in order to increase the relationship value. The CRM database gives every salesperson access to the same customer information. With Enterprise 2.0, it is possible to enable the customers’ participation to communities in a wiki or a blog to share information about products with employees. Since everyone can participate, the community can monitor the information and clear misconceptions up that might have developed about a customer or product line. While, the main function of HRM Systems consists of tracking existing employee’s data which traditionally includes personal histories, skills, capabilities, accomplishments and salary. The use of social networking tools can be exploited to manage relationships between employees, customers, suppliers, consultants, etc.

The management of contact information is not sufficient, we need efficient and effective way to represent, store and retrieve relationship information in order to merge them with interests, skills and activities data. FOAF provides the \texttt{foaf:knows} to create networks of people. The most important limit of the FOAF approach is that we are not able to know the type of the relationship (e.g. colleagues, friends, etc.) between two persons. XFN\textsuperscript{18} (XHTML Friends Network) and Relationship\textsuperscript{19} vocabularies can be used to describe relationships between people and to enrich FOAF profiles.

Eventually, we would like to underline the existence of FOAFRealm\textsuperscript{20} that is another extension for FOAF. FOAF-Realm faces the problem of evaluating the friendship between two persons, to obtain trust information, by enriching the \texttt{foaf:knows} statement.

### III. A Sample Scenario: Searching for Contacts

In this section we will provide a specific instantiation of the proposed Framework. It sets out to describe how we can exploit the Semantic Web data modeling to easily find information about external people using the own network of relationships. In the sample scenario, “Francesco Orciuoli”, an employee of the “Z-Company”, asks the following question: Which of my colleagues met the X-Company R&D director?

We can start modeling the relationships between "Francesco Orciuoli" and his colleagues and other external relationships (imagine them as contacts). The following RDF code (written in Turtle\textsuperscript{21}) language could be produced by an internal Social Network tool using FOAF and Relationship specifications:

```turtle
sample:FrancescoOrciuoli
  a foaf:Person ;
  foaf:name "Francesco Orciuoli"ˆˆxsd:string ;
  <http://purl.org/vocab/relationship/coworkerOf> sample:MarcelloRosciano ,
  sample:MassimilianoMinei ,
  sample:FrancescoZurolo ;
  foaf:mbox <mailto:orciuoli@diima.unisa.it> .

sample:MassimilianoMinei
  a foaf:Person ;
  foaf:name "Massimiliano Minei"ˆˆxsd:string ;
  <http://purl.org/vocab/relationship/hasMet> sample:MarioRossi .

sample:FrancescoZurolo
  a foaf:Person ;
  foaf:name "Francesco Zurolo"ˆˆxsd:string ;
  <http://purl.org/vocab/relationship/hasMet> sample:GiorgioVerdi .

The \texttt{rel:colleagueOf} and \texttt{rel:hasMet} properties, coming from the Relationship vocabulary, are used to model the \texttt{colleague-relationship} between "Francesco Orciuoli" and his colleagues "Francesco Zurolo", "Massimiliano Minei" and "Marcello Rosciano", while the \texttt{meet-relationship} is between "Massimiliano Minei" and an external person called "Mario Rossi" and, finally, the \texttt{meet-relationship} is between "Francesco Zurolo" and "Giorgio Verdi".

Furthermore, let’s assume to have some information associated to the external contacts provided, together with the aforementioned \texttt{rel:hasMet} relationships, by a mobile device application able to scan the business cards of "Giorgio Verdi" and "Mario Rossi" when they met "Francesco Zurolo" and "Massimiliano Minei" (at a conference on "Adaptive Systems") and to connect with a service that executes data representation and storage operation using FOAF and ResumeRDF. Below an example is reported:

```turtle
sample:MarioRossi
  a foaf:Person ;
  foaf:mbox <mailto:mario.rossi@x-company.com> ;
  foaf:name "Mario Rossi"ˆˆxsd:string ;
  cv_rdfs:aboutPerson sample:MarioRossi_person ;
  cv_rdfs:hasWorkHistory
  a cv_rdfs:CV ;
  rdfs:label "Mario Rossi"ˆˆxsd:string ;
  cv_rdfs:aboutPerson sample:MarioRossi_person ;
  cv_rdfs:hasWorkHistory
```

\textsuperscript{17}http://hdis.cs.uga.edu/~aleman/efw2007/bojars_efw2007.html
\textsuperscript{18}http://gmpg.org/xfn/
\textsuperscript{19}http://vocabulary.org/relationship.html
\textsuperscript{20}http://www.foafrealm.org/
\textsuperscript{21}http://www.w3.org/TeamSubmission/turtle/
The above code lines describe the same person, "Mario Rossi", using FOAF and ResumeRDF. In particular, ResumeRDF is used to model aspects related to the working activity of "Mario Rossi", i.e. his curriculum (sample:sample:AttendedEvent) and his current job information (sample:CurrentEmployment). In order to respect the ResumeRDF specification we have to provide other profile information (sample:sample:AttendedEvent) that is redundant owing to the presence of the FOAF profile for "Mario Rossi". The property rdfs:seeAlso is used to link the ResumeRDF profile with the FOAF profile for "Mario Rossi". Furthermore, sample:sample:AttendedEvent is linked to sample:SampleEvent through the property cv_rdfs:aboutPerson and, finally, sample:sample:CurrentEmployment is linked to sample:CurrentEmployment by means of cv_rdfs:hasWorkHistory property. The last information we need concern the organization:

sample:CurrentEmployment
a cv_rdfs:Employment ;
rdfs:label "X-Company" ;
rdfs:seeAlso sample:X-Company .

Now, the needed data are all linked together and we are able to answer to the question asked by "Francesco Orciuoli". The corresponding SPARQL query is reported in the following code:

```sparql
SELECT ?co_name
WHERE {
  ?me cv_rdfs:Type foaf:Person .
  ?me foaf:email "mailkto:vicriciuoli@diima.unisa.it" .
  ?me rdfs:label "Mario Rossi" .
  ?me rdfs:hasMet ?ext .
  ?ext rdfs:seeAlso ?ext_f .
  ?ext_f cv_rdfs:Type foaf:Person .
  ?ext_p rdfs:hasWorkHistory ?history .
  ?history cv_rdfs:employedIn ?organization .
  ?history cv_rdfs:jobTitle "R&D Director" .
  ?organization cv_rdfs:Name "X-Company" .
  ?co foaf:name ?co_name .
}
```

Let us note that, in the previous query, we navigate the RDF graph by using patterns on the RDF triple in order to find the name of the colleague ("Massimiliano Meini") of "Francesco Orciuoli" who met the "R&D Director" of "X-Company". In particular, we use a simple text to identify the job title "R&D Director". A better solution could be to model job title, competencies, etc. using the SKOS (Simple Knowledge Organisation System) that provides a standard way to represent knowledge organization systems using RDF.

IV. CONCLUSIONS AND FUTURE WORKS

In this work we have illustrated a Framework based on Semantic Web technologies with the aim of supporting knowledge creation and sharing scenarios in the context of the Enterprise 2.0. We have also presented an architectural solution for integrating heterogeneous applications. The solution is based on the use of Semantic Web vocabularies to model, represent and harmonize data produced by different applications; on the use of a RDF Data Storage System and a RDF Query and Rule Engine; and, finally, on the definition of a functional block called Personal Work Learning Environment (PWLE) used, by single workers, to manage working and learning tasks by interacting with RDF Query and Rule Engine. In our proposal we mean to face the problem of integrating heterogeneous applications, in the emerging enterprise environments, both at user-level (PWLE) and data-level (Semantic Web vocabularies). From the architectural viewpoint, the most important advantage of our proposal is the possibility to exploit, in the Enterprise 2.0 context, data coming from new integrated applications only considering new schemas, harmonization statements, rules and queries. In this way, we could provide new attractive semantic-based information retrieval scenarios without architectural revolutions. Of course, other levels, like security, trust, service orchestration, etc. are to be considered in a complete application integration framework. The next steps of our research activity will consist in defining a detailed model for all the Enterprise 2.0 aspects and in developing a PWLE system.

REFERENCES

[9] U. Bojars, J. G. Breslin, From Online Community Data to RDF.

22)http://www.w3.org/TR/rdf-sparql-query/

23)http://www.w3.org/2004/02/skos/