K-link: A Peer-to-Peer Solution for Organizational Knowledge Management

Giuseppe Pirro'[1], Domenico Talia[1,3], and Massimo Ruffolo[2,3]

[1] Department of Electronics, Computer Science and Systems, University of Calabria
Via Pietro Bucci, Cubo 41C, I-87036 Rende (CS), Italy
{gpirro,talia}@deis.unical.it
[2] ICAR-CNR, Via Pietro Bucci, Cubo 41C, I-87036 Rende, Italy
ruffolo@icar.cnr.it
[3] EXEURA s.rl, Rende (CS), Italy

Abstract. In the latest years knowledge management received more and more attention as a source of competitive advantage for enterprises and organizations, therefore becomes important to understand how computer science solutions should be designed to efficiently manage knowledge. Most of the current knowledge management systems use technological architectures that are in contradiction with the social processes concerning the creation of new knowledge, slowing down organizational innovation. Actually, most of those systems use centralized architectures filtering knowledge from any form of personal and contextual interpretation. Recently a new paradigm supporting cooperative and dynamic aspects of knowledge management (KM) has been proposed: Distributed Knowledge Management (DKM). In particular, peer to peer (P2P) architectures seem to naturally fulfil the requirements of this new model. Nevertheless, current P2P architectures suffer from heavy limitations due to the lack of semantic supports for handling knowledge. To overcome these limitations, the scientific community is appraising the possibility of using ontologies as a semantic support in KM processes. This paper presents an ontology based P2P system for DKM named *K-link*. The system design and its implementation are described. Moreover an ad hoc ontology framework for supporting organizational KM is also presented.

1 Introduction

In the nineties a new organizational paradigm has been proposed [1]. This paradigm points out knowledge as a key resource for organizations and aims at establishing paths to be followed for better exploiting organizational knowledge. Earlier organizational models [2] saw the organization like a box with the aim to maximize the output given an input or like something that can be scientifically and rigorously managed. With Simon's theories about the bounded rationality [3]the theme of the KM becomes more important and the role of the organization in the KM processes notably changes. The organization becomes a way to connect the knowledge of many subjects into a more complete understanding of the

S. Joseph et al. (Eds.): AP2PC 2006, LNAI 4461, pp. 86–97, 2008.

reality. The technologies' role also changes: they become a way to increase people's rationality by allowing memory and computational increase and knowledge exchange. Along the years several other theories about knowledge have been proposed. A generally accepted classification proposed by Polanyi [4,5] and extended by Nonaka [6] identifies from one side: "tacit knowledge" as the knowledge resulting from personal learning within an organization. Form the other side, the "explicit knowledge" is a generally shared and publicly accessible form of knowledge. This kind of knowledge is typically stored. Explicit knowledge can also be classified on the basis of the following forms: "structured" (available in database), "semi-structured" (generally available in Web sites: HTML pages, XML documents, etc.) and "unstructured" (available as textual documents: project documents, procedures, white papers, templates, etc.). More recently, new importance has been given to social processes and to the communities of practice as sources of knowledge. Through communities of practice the individual could learn from the community, but also the same community can innovate and create new knowledge. "Community is the social dimension of the practice", which is the ideal context for creating knowledge, a virtual space where the individuals learn, putting personal experiences and social competences in relation and having the possibility of acting on the same social competences through sharing [7]. According to this vision, organizations must become a community of communities, offering spaces for the creation of autonomous communities, that are connected one another . The different types of KM solutions are related to the image of the corresponding social interactions in KM processes. According to this consideration, technological systems for KM can be classified and inserted in a scheme according to the adopted social model. Therefore, on one hand we have centralized systems that are practically identified with the Enterprise Knowledge Portal (EKP) and, on the other hand, we have distributed KM systems. In this paper we describe *K-link*, a JXTA based P2P system coherent with the vision of distributed knowledge management (DKM). K-link provide users with an environment allowing a work activities management in a semantic way. For example projects users are involved in, are in K-link represented by workspaces. A workspace is a community of peers endowed with a set of tools (such a Calendar, File Sharing, etc.) enabling distributed and real-time interactions. For example the File Sharing tool allows users to search documents inside a workspace on a semantic basis (selecting an ontology concept and forwarding the query to the network, all the documents classified by the workspace members under that concept will be given back). Indeed, P2P systems seem particularly suitable to implement the two core principles of DKM, namely the principle of autonomy - communities of knowledge should be granted the highest possible degree of semantic autonomy to manage their local knowledge - and the principle of coordination - the collaboration between autonomous entities must be achieved through a process of semantic coordination, rather than through a process of semantic homogenization-. In K-link, each knowledge entity (or Knowledge Link Node - KLN) is represented by a peer, and the two principles mentioned above are implemented in a direct way:

– each KLN (peer) is endowed with a set of services for creating and organizing its own local knowledge (autonomy), and
– ii) defining cooperating structures (peer groups) and an ad hoc ontology framework in order to achieve semantic interoperability (e.g., classifying and searching documents on a semantic basis).

The paper is organized as follows. Section 2 presents the K-link architecture and its implementation in JXTA explaining how it provides a useful support to DKM. Section 3 presents the ontology framework supporting the system, developed by a Disjunctive Programming Language (DLP+). Section 4 presents the implemented system. Finally,sections 5 and 6 outline some related works in the DKM field, draw some conclusions and sketch future work.

2 K-link Architecture

K-link is a P2P system which allows a collection of subjects (individuals, groups, or whole organizations) to manage and search for knowledge on a semantic basis through the ontology support. Because of its decentralized model, K-link doesn't impose a single model to classify knowledge. Furthermore, through K-link it is possible to integrate knowledge stored in pre-existing systems (such as servers, knowledge bases, etc.) if based on our ontology framework. In the next sections we describe the K-link high-layer and low-layer architecture. For both layers we analyze the main components and the services that allow the system to support users and communities in managing distributed organizational knowledge. The following figure shows the system architecture.

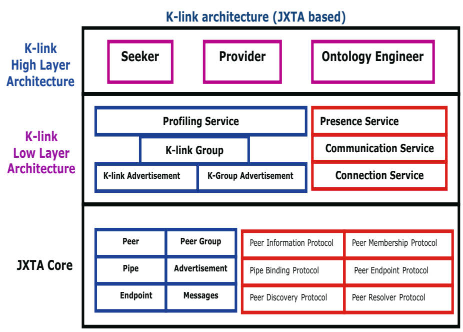

Fig. 1. K-link low level architecture

2.1 K-link Low-Level Architecture

The K-link low-layer architecture contains services built by the JXTA framework. JXTA (http://www.jxta.org) is a set of protocols, through which P2P ad-hoc networks can be carried out. JXTA also allows the autonomous peer group creation. By the peer group abstraction, the communities of practice can be formed. In a specific group it will be possible to create a set of services only for the members of the group. Moreover, this framework allows various device set (server, PC, PDA, etc.) to communicate within a peer group independently from the transport protocol. The common format for exchanging information inside JXTA is the advertisement. An Advertisement is an XML document containing information about each network resource (peer, peer group, services, etc.). A certain resource (to be used), must be first "announced" to the network. In JXTA there is a wide number of protocols; each of them devoted to an activity. For example, the Pipe Binding Protocol handles the creation of pipes, namely communications channels. Furthermore, the Peer Rendezvous Protocol handles the message exchanging within the network of JXTA rendezvous peers . A Rendezvous peer, or "super peer", represents a generic peer provided with message routing capabilities. From this point of view, JXTA allows to overcome the limits of pure P2P architectures.

2.2 K-link Services

K-link services created on the top of the JXTA architecture, allow KLN to implement all the requirements of DKM as discussed before. In the following we give a brief description of each service.

- **Profiling Service:** This service is close in meaning with the profile concept. A profile represents a semantic description of each KLN within a K-link group. Each profile contains useful information about the KLN (name, email address, skills, etc.). Moreover, the Profiling Service guarantees the profile consistency with respect to any change. If any change is detected the profile will be republished within the group.
- **Connection Service:** The Connection Service is responsible for all the operations concerning the JXTA network setting up. This service creates the super group K-link within the JXTA NetPeerGroup. This service also deals with the creation of the other subgroup of the super group. All the services (Discovery Service, Rendezvous Service etc.) will be inherited from this group. Moreover, this service provides all the operations for joining the K-link groups.
- **Communication Service:** The Communication Service deals with the information exchanging operations through the Jnutella protocol. Jnutella is a JXTA-based implementation of Gnutella. It is composed by a set of descriptors designed using the JXTA messages. Moreover, Jnutella adds two new kinds of descriptors (i.e. profile request and profile reply) for exchanging profile information. Jnutella messages are not forwarded to all KLNs, but only to a subset of those (contained in a buddylist) the KLN has chosen to deal with. In this way, the network traffic is reduced and messages will not propagate blindly.

- **Presence service:** The Presence Service provides a mechanism for exchanging presence information. Through this service, each KLN lookups the current status of the other KLNs (contained in its buddylist). In fact, this service on a regular basis manages message exchanging in a ping/pong style.

2.3 K-link High-Level Architecture

This layer implements the following K-link roles: *seeker*, *provider* and *ontology engineer*. The architecture of each KLN is shown in figure 2.

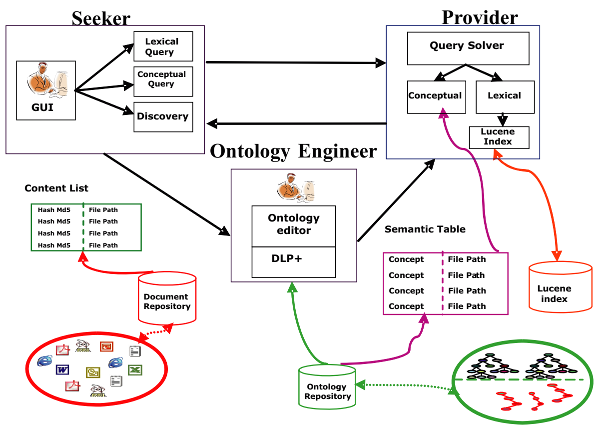

Fig. 2. K-link high level architecture

2.4 K-link Node

A K-link node (KLN) represents a knowledge entity that able to produce, store and exchange knowledge. In general, a KLN could be a PC, a server, or also a community of entities that exchange information in a P2P way. A KLN will be able to play the role of Seeker when it searches for knowledge, or the role of Provider when it publishes its own knowledge. Furthermore, a third role (ontology engineer) leads the processes of ontology personalization. In K-link system ontologies are represented by DLP+ [9]. In the following we illustrate the main components of this layer.

2.5 K-link Main Components

- **Document Repository:** A Document Repository is the place where each KLN stores its own knowledge. Thus, this repository can be viewed as a private space in which each KLN holds its own documents and data according to a local scheme (e.g., a file system, a database, etc.). This repository contains knowledge that through other system components will be annotated, indexed, and shared. It is important to guarantee the consistence of that repository since it will assure the "quality" of the answers to the others KLNs.

- **Ontology Repository:** This repository contains the DLP+ implementation of the ontology framework described in section 3. The DLP+ ontology language is an extension of the Disjunctive Logic Programming (DLP) that extends Datalog allowing disjunction in the rules' head. The presence of disjunction in the heads of the rules makes DLP inherently non-monotonic. The ontology repository contains also the contexts (i.e. personal concepts networks) created from the Upper Ontology. A context represents a vision about a piece of knowledge. From this point of view, the application gives autonomy to a user to specialize its context as her/him prefers. The only tie is that the basic concept (the root of the hierarchy) must be a core ontology concept. Documents stored in the document repository will be semantically annotated to the context concepts, allowing to assign a semantic meaning to their contents.
- **Lucene Index:** This repository contains the Lucene index allowing to search for information inside user documents. This index is permanently stored on the disk. Nevertheless, Lucene (http://lucene.apache.org) doesn't provide any mechanism for checking index consistency. In fact, in case of a document removal or changing the index should be updated manually. In the K-link system the index consistency is assured through a cyclical control checking of the contained documents.
- **Content List:** This list contains pairs *(Hash Md5, File Path)* used for managing the Lucene index consistency. Through the cyclical control of such list, the system can know which files currently owns. The file paths are obtained from the document repository, while md5 hashes are obtained trough the JXTA API. From the modifications of the Md5 value, file changes will be noticed and the system can proceed to the new indexization or deletion inside the index.
- **Semantic Table:** This table stores permanently the file-to-concepts associations on disk as pairs *(Concept Id, File Path)*. In the current K-link implementation it is possible to associate one file to more than one concept by creating different views for it. The role played by the semantic table is essential for conceptual querying. In fact, by specifying the name of a concept and issuing a remote query, all the contents classified from other KLNs under that concept will be returned.

2.6 K-link Main Roles

Each KLN can play three roles: seeker, provider and ontology engineer. Here we describe them in detail.

1. **Seeker.** A KLN plays the seeker role when it searches for knowledge inside a group. Moreover, the Seeker role includes the mechanisms allowing the discover of both KLNs and K-link groups. A KLN can issue two kinds of query (conceptual and keyword based) forwarding requests to its "buddylist" that contains the others KLNs chosen to deal with. For each query, a

handling process will be activated. with the aim to gather the responses and to send back them. The seeker component of the receiver can group these results on query type and KLN sender basis. When the results are shown, a KLN can activate a download request handled by the JXTA CMS service (http://cms.jxta.org).
2. **Provider.** In the Provider role, a KLN should deal with queries coming from other KLNs. Also in this case, for each query (identified a timestamp) a handling process will be activated. This process queries the local indexes for the requested information. If any result is found, the KLN sends back replies to the remote peers. The handling process will be different depending on the query type.
   - Keyword query: for this type of query, the KLN looks at its local Lucene index, obtaining the documents containing the requested keyword. The results composed by pairs (file path, content id), will be sent back to the requester and then locally handled through its query handler process. The content id allows the requester peer to identify the last file version.
   - Conceptual query: When a conceptual query is issued, a KLN in the provider role will check its own semantic table giving back all the documents associated to the concept expressed in the query. Also in this case it is important to send back the content id in order to obtain the last file version.
3. **Ontology engineer.** In the Ontology Engineer role, a KLN peer can deal with the context creation operations. A context represents a proper view on a part of the domain of interest. In fact, each KLN can specialize this part of knowledge on a personal basis rising from real use cases. Moreover, this role deals with all the core ontology modification operations. In this version of K-link, those operations are carried out manually, but in a future release they will be handled through a distributed mechanism.

3 The K-link Ontology Scenario

Recently KM tools using ontologies as semantic support for describing contents have been designed. Mainly two classes of tools can be identified:

- Tools that through standardized ontologies aim at working out all the comprehension problems deriving from their use in dynamic environments.
- Tools that don't accept any layer of standardization, and allow users to define ontologies (in this case would be more appropriate to speak of contexts) according to the needs and the abilities of everyone. In this case, all the processes of meaning negotiation are totally trusted to automatic mechanisms with no (or almost) involvement from the users.

In an organizational environment both those classes of tools cannot be properly used. An intermediate solution provided with a time-changing basic ontology could represent an effective trade-off between the demand of common models and the demand of individuals (people or organizations) of defining local concepts in their models. Our approach, through the basic ontology and the user's feedbacks tries to build a shared conceptualization. Moreover, K-link goes beyond the classification of information, taking into account all the resources and the cognitive legacy of an organization (business processes, human resources, etc.) through their accurate definition. Through an intensive use of Ontologies, the work of organizing, understanding and looking for information will result more accurate, simple and efficient. The proposed ontology-based framework [8] is organized as in two layer as shown in figure 3. The first ontology layer contains the Upper Ontology (UO) that contains concepts characterizing the organizational background knowledge. These concepts are used for annotating Core Organizational Knowledge Entities (COKE) contained in the second ontology layer. Our framework gives an abstract representation of COKE allowing handling of semantic knowledge objects (e.g. semantic search and retrieval, semantic process management, etc.). In particular, the framework provides a uniform abstract representation of static (concepts) and dynamic (processes) organizational knowledge handled by users. All peers are initially provided with this framework in which the upper ontology contains basic organizational knowledge. Each peer can extend the upper ontology with own relevant concepts. In the following the framework structure is explained more in detail.

3.1 The Upper Ontology

The UO defines concepts characterizing the typical organizational knowledge background. It specifies explicitly organizational topic (i.e. declarative knowledge concerning the concepts characterizing an application domain). For example an IT enterprise background is founded on concepts coming from computer science such as databases, programming languages, architectures, etc. The UO provides COKE ontologies with concepts to formally annotate their contents.

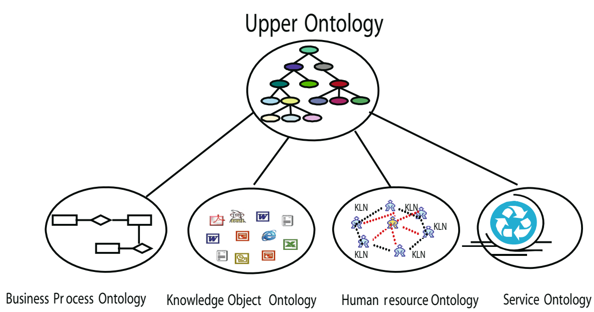

Fig. 3. K-link ontology framework

3.2 The COKE Ontologies

COKE Ontologies contain the formal representation of human resources and their organization in groups, processes and their activities, knowledge objects constituting elements produced or used in business processes, services in term of instruments used during business process execution. The **Human Resource Ontolog**y defines individuals working in the organization (knowledge workers) and social groups they are involved in. For each individual, a profile in term of skills, topics of interest, organizational role, group membership is defined. For each group (community of practice, project team, organizational group, etc.), a group profile in term of topics of interest, required services is defined. The **Business Processes Ontology** contains procedural knowledge related to the managerial and operational processes. The business process ontology exploits an interesting capability of DLP+ language allowing the expression of taxonomic and non-taxonomic relations between classes enabling the representation of process meta-model, process schemas and process instances. Therefore, each process is described in terms of a three layer structure. The meta-model layer allows the definition of process elements (i.e. activities, sub-processes, transition states and conditions, involved actors, treated topics). The schema layer allows the definition of a single process in term of process elements. The instance layer allows the definition and acquisition of process instances. The **Knowledge Objects** Ontology maps the structure of logical objects (e. g. textual documents, web pages, process activities, blog and chat sessions, e-mail, etc.). These are used in the business processes and handled by the human resources through services. Semantic knowledge objects management and handling (e.g. search and retrieval) is facilitated by the annotation on the UO concepts. The **Services Ontology** identifies the tools by which knowledge objects are created, acquired, stored and retrieved. The execution of a query on the UO can be executed using a specific tool able to retrieve all the elements related with a specific concept. Element can be filtered to obtain a specific COKE related to the query. For example, a query result can contain info about people knowing a given concept or knowledge objects related to some concepts.

4 K-link in a Nutshell

Figure 4 shows the main GUI from which a user can create new workspaces, open existing workspaces, deleting existing workspaces, and also deal with all the Personal Knowledge Management tasks (local document indexing, semantic and full text search, etc.). Furthermore is also possible to invite contacts to the created workspaces.

The Figure 4 shows also the contacts view trough which is possible to search and add new contact to our contact list. Moreover from the contact view is also possible to start a one to one chat session.

Figure 5 shows the K-link *Personal Knowledge Management* environment.

From the *Personal Knowledge Management* GUI is possible to query local knowledge. Currently there are two kinds of queries, the first (keyword based)

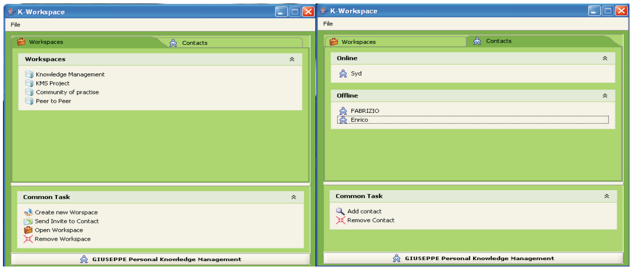

Fig. 4. K-link main GUI

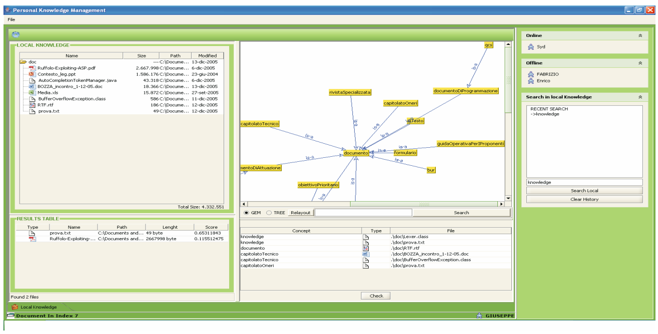

Fig. 5. K-link Personal Knowledge Management GUI

uses the Lucene index and show the results in a *JTable* as can be viewed in the left side of the figure. From this table is possible to open directly the content. Is also possible to associate file to concepts by dragging the file on the left hand side to the ontology representation in the center. The figure shows also the the semantic table represented also in this case by a *JTable*. From this table is possible to erase semantic associations and to check the semantic table consistency with respect to the ontology (e.g in the case of a concept deletion).

5 Related Works

To the best of our knowledge, recently only two P2P KM systems have been proposed: SWAP and KEEx. SWAP (Semantic Web and Peer to Peer) is a research project started in 2002, aiming at combining ontologies and P2P for KM purposes. SWAP allows local KM by a component called LR (Local node repository), which gathers knowledge from several source and represents it in RDF-Schema. Moreover, SWAP allows to search for knowledge using a language called SeRQL an evolution of RQL [10]. KEEx [7] is a P2P architecture that aims to combine both semantic and P2P technologies. This system implemented in JXTA allows a set of K-nodes to exchange information on a semantic basis.

Semantic in KEEx is achieved by the notion of context. A context in KEEx, represents a personal point of view about reality and is represented using a proprietary language called CTXML [12]. KEEx lets the user completely free about context creation without providing him with any organizational background. For the reasons discussed in section 3 we argue that although this approach is innovative, due to the presence of an automatic mapping algorithm which aim is to find correspondences between concepts present in different contexts, this complete autonomy could turn into isolation.

6 Conclusions and Research Issues

In this paper we argued that technological and social architectures must be consistent for supporting KM processes. Along the paper sections we discussed as P2P systems naturally fulfill this requirement. We designed a P2P architecture called K-link and implemented it by using JXTA, Lucene and other technologies. K-link addresses all the main needs emerging in an organizational distributed KM scenario through an ad-hoc ontology framework. Nevertheless, a number of research issues related to the mapping of the distributed knowledge into technological requirements emerged. Two of them are:

- Mapping algorithm: In order to solve the semantic interoperability problem we are designing an algorithm that aims at discovering the potential semantic relations among concepts in different contexts in a semi-automatic way. This algorithm uses all the expressiveness of the DLP+ and *Wordnet* as source of knowledge about the word.
- Automatic classifier: Based on the rules definition related to the concepts of Upper Ontology we are developing a semiautomatic classification system supported by the Hylex [11] system.

References

1. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company. How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York, USA (1995)
2. Taylor, F.W.: The Principles of Scientific Management. Harper and Row, New York, USA (1911)
3. Simon, H.A.: Theories of bounded rationality. In: McGuire, C.B., Radner, R. (eds.) Decision and organization: A volume in honor of Jacob Marschak (Ch. 8), Amsterdam, Olanda (1972)
4. Polanyi, M.: The tacit dimension. Routledge and Kegan Paul, London, England (1966)
5. Polanyi, M.: Tacit Knowledge. Ch.7 In: Prusak, L. (ed.) in Knowledge in Organizations, Butterworth-Heinemann, Boston, USA (1997)
6. Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation. In: Organization Science, vol. 5 (1994)

7. Bonifacio, M., Nori, M., Bouquet, P., Busetta, P., Danieli, A., Don, A., Mameli, G.: KEEx: A Peer-to-Peer Tool for Distributed Knowledge Management. In: Proc. P2PKM 2004, Boston, MA (2004)
8. Ruffolo, M., Gualtieri, A.: An Ontology-Based Framework for Representing Organizational Knowledge. In: Proc. I-Know - International Conference on Knowledge Management, Graz, Austria (June 2005)
9. Eiter, T., Leone, N., Mateis, C., Pfeifer, G., Scarcello, F.: A Deductive System for Non-Monotonic Reasoning. In: Fuhrbach, U., Dix, J., Nerode, A. (eds.) LPNMR 1997. LNCS, vol. 1265, pp. 364–375. Springer, Heidelberg (1997)
10. SWAP:Ontology-based Knowledge Management with Peer-to-Peer Technology Marc Ehrig Christoph Tempich, Jeen Broekstra Frank van Harmelen, Marta Sabou, Ronny Siebes, Steffen Staab, Heiner Stuckenschmidt
11. Ruffolo, M., Leone, N., Manna, M., Sacc, D., Zavatto, A.: Esploiting ASP for Semantic Information Extraction. In: The ASP 2005 workshop - Answer Set Programming: Advances in Theory and Implementation, University of Bath, Bath, UK, July 27th–29th (2005)
12. Bouquet, P., Serafini, L., Zanobini, S.: Semantic coordination: A new approach and an application. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 130–145. Springer, Heidelberg (2003)