HCOME: A Tool-Supported Methodology for Engineering Living Ontologies

Konstantinos Kotis, George A. Vouros, and Jerónimo Padilla Alonso

Dept. of Information & Communications Systems Engineering,
University of the Aegean, Karlovassi, Samos, 83100, Greece
{kkot, georgev, pgeron}@aegean.gr

Abstract. The fast emergent areas of the Semantic Web and knowledge management push researchers to new efforts concerning ontology engineering. The development of ontologies must be seen as a dynamic process that in most of the cases starts with an initial rough ontology that is later revised, refined, enriched, populated and filled in with details. Ontology evolution has to be supported through the entire ontology lifecycle, resulting to a living ontology. The aim of this paper is to present the Human-Centered Ontology Engineering Methodology (HCOME) for the development and evaluation of living ontologies in the context of communities of knowledge workers. The methodology aims to empower knowledge workers to continuously manage their formal conceptualizations in their day-to-day tasks. We conjecture that this methodology can only be effectively supported by eclectic human-centered ontology management environments, such as the HCONE and SharedHCONE.

1 Introduction

Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and for the evolution of the Semantic Web and its applications. We consider communities of knowledge workers that are involved in knowledge-intensive tasks within an organization, or World Wide Web users with common interests. Knowledge workers are unfamiliar with knowledge engineering principles and methods, and most of the times have little or no training on using ontology specification tools. In such a distributed setting ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge. However, it seems that there will always be the case that community members devise more than one ontologies for the same domain. For community members to explicate, maintain and evaluate the changing conceptualization of a domain, they must get powerful tools that will allow them to edit, review, update and maintain formal ontologies, on their own as well as in collaboration with colleagues [1].

Several methodologies have been proposed for the engineering of ontologies within a knowledge management setting. From the identification of goals and

C. Bussler et al. (Eds.): SWDB 2004, LNCS 3372, pp. 155 – 166, 2005.

requirements' specification, to the implementation, evaluation and maintenance of the conceptualisations, the ontology life cycle must be clearly defined and further supported by ontology development tools [2]. From the methodologies described in [3], [4], [5] and [6], the OnToKnowledge methodology supported by the OntoEdit ontology development tool, being the most well known one, starts from the initial stages of knowledge management projects (feasibility and requirements) and proceeds to the deployment and maintenance of an ontology-based knowledge management system [4], [5]. The OnToKnowledge methodological approach focuses on the application-driven development of ontologies, supporting the introduction of ontology based knowledge management systems [4], [5]. According to this approach, the maintenance of ontologies is primarily an organizational process driven by the knowledge engineer who gathers updates to the ontology and initiates the switchover to a new version of the ontology after thoroughly testing possible effects to the application [4],[5].

In contrast to the methodologies that are centered to the knowledge engineers, we propose the use of a human-centered approach to ontologies management [7], where the active participation of knowledge workers in the ontology life cycle is accentuated. Doing so, ontologies are developed and managed according to knowledge workers' abilities, are developed individually as well as conversationally, and put in the context of workers' experiences and working settings, as an integrated part of knowledge workers' "knowing" process [1], [8]. To leverage the role of knowledge workers by empowering them to participate actively in the ontology lifecycle, the human-centered approach entails the development of tools that provide greater opportunities for workers to manage and interact with their conceptualisations in a direct and continuous mode [7]. Although the final ontology is the product of knowledge worker's collaboration, knowledge engineers must join the discussion in order to further validate the final formal representation of the conceptualizations.

To further support our conjecture for the need of human-centered methodological approaches, let us consider the following ontology management scenarios in a living organization setting:

Scenario No 1: Involved in a knowledge retrieval process, a worker is searching for a specific piece of information about best practices concerning the design of a product type. The retrieval tool exploits the ontology concerning product designs, but the worker can neither find the terms that she thinks to be appropriate for querying the system, nor can she get the needed information by any combination of existing terms. She soon finds out that the definitions of some terms must be changed to reflect the information related to the new case at hand. The information is there, but cannot be reached, since the ontology does not reflect the up-to-date practice of the organization. Imagine now the same case happening for five workers per day in a fast changing domain. We suggest that workers must be empowered to shape their information space, working in collaboration with colleagues and knowledge engineers.

Scenario No 2: In a knowledge use process, a worker browses, recalls existing knowledge items, and process them for further use. During this process the worker may produce derivations that should be captured as new knowledge, indexed by new

terms, or by combinations of existing terms. Capturing derived knowledge is very important. Empowering this worker with the proper tools for describing her conceptions formally, incorporating them in organization's information repository, submitting and sharing this information with co-workers readily, accelerates much the knowledge processes.

Scenario No 3: In the day-to-day information creation and import tasks, workers are devising business documents, proposals, product reports, best practices, problem/fault reports, etc. Indexing such information using formal ontological commitments should be done in a seamless way by knowledge workers themselves, during authoring, allowing them to devise, expand and update their shared conceptualizations at the same time.

This paper emphasizes on the methodological implications to ontology engineering of the HCONE and SharedHCONE ontology engineering environments [7] that are oriented to the way people interact and shape their conceptualizations and to the way conceptualizations are formed as part of knowledge workers' day-to-day activities [1].

2 Management of Ontologies

As it is widely argued and shown in the above scenarios, ontologies explicate conceptualizations that are shaped and exploited by humans during *practice*. Being part of knowledge that people possess, ontologies evolve in communities as part of *knowing* [8].

Therefore, ontology management in the context of communities of knowledge workers involves the development, evaluation and exploitation of conceptualizations that emerge as part of practicing in their working contexts. In particular it involves:

- *The development of individual ontologies.* People develop their own conceptualizations that may either explicate (e.g. by formalizing concepts, by taking notes about their meaning or just by naming them) or not (by storing them in the background of their minds). In their day-to-day activities people develop their conceptualizations, either by improvising, by specializing/generalizing/aggregating existing concepts based on their experiences and on interaction with other community members, or by synthesizing existing conceptualizations.
- *The development of commonly agreed group ontologies*. Developing commonly agreed and understandable ontologies is a very difficult and resource-demanding task that requires members of the communities to work synergistically towards shaping the information they exploit. Working synergistically, workers map others' conceptualizations to their own and put them in the context of their own experiences. This leads to a conversation whose back-and-forth, as it is pointed in [8], not only results in exchanging knowledge but also in generating new knowledge.
- *The evaluation and exploitation of ontologies.* Exploitation and evaluation of ontologies as part of the day-to-day practice of communities can be considered only as part of knowing. Conceptualizations are put in practice or in the criticism

of community members who, as already pointed, have to compare them with their own conceptualizations and put them in the context of their own experiences. Evaluation can result in new meanings since concepts are seen under the light on new experiences and evolving contexts.

To empower knowledge workers to participate actively in the ontology engineering process in collaboration with colleagues and knowledge engineers, tools must enable them to improvise, to synthesize ontologies, to produce mappings/alignments between existing ontologies, and to collaboratively develop ontologies with their co-workers, in ways that are natural (according to their cognitive abilities, skills, knowledge, education, context of work and so on) for them, and so that the semantic validity of specifications is assured. Ultimately, this must happen in the background of the day-to-day knowledge intensive activities of workers, seamlessly to their working practices.

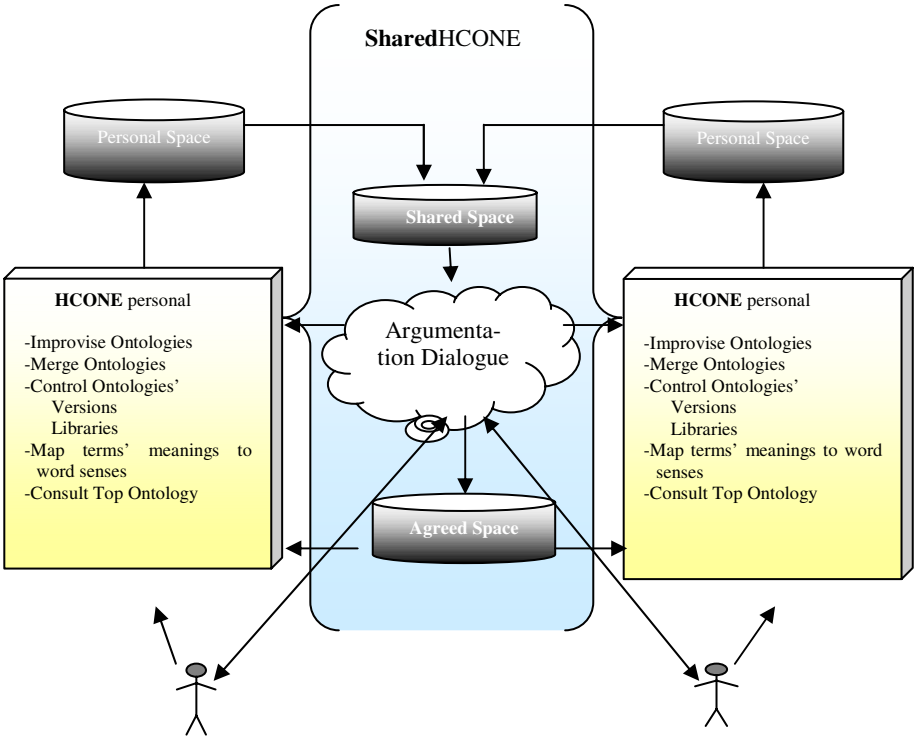

Fig. 1. HCONE decentralized model to ontology engineering

3 HCONE and SharedHCONE

In [7] we extensively describe HCONE and SharedHCONE tools that have been developed to support the above requirements.

HCONE (Human Centered ONtology Environment) follows a decentralized model to ontology engineering that is shown in Fig. 1. According to this model people can create their own ontologies stored in a personal space. Ontologies can be later publicized and shared among groups of workers that jointly contribute to ontologies development, with the aim to reach an agreement in conceptualizing their domain. During this process, workers may evolve ontologies by improvising in their personal space, map and synthesize their conceptualizations with the conceptualizations of their co-workers and discuss their arguments, objections and positions within the group. During collaboration, workers follow a structured argumentation process in which they may raise issues, propose solutions via stating positions, provide arguments for or against a position etc. Agreed ontologies are stored in a virtual space and can be further shared, evolved in workers' personal space and so on.

HCONE (Fig. 2.) is a modular environment, providing access to any integrated tool in any HCONE point. Doing so, workers are free to combine their own method for using the environment, following an eclectic way to ontology engineering. For instance, a worker may construct an ontology in her personal space while receiving comments on a previous version of the same ontology that has shared with co-workers. In the meantime, she is trying to comply with generic ontological commitments that the group has agreed to comply with, while in another slice of her work she is trying to merge her ontology with an ontology issued by a co-worker.

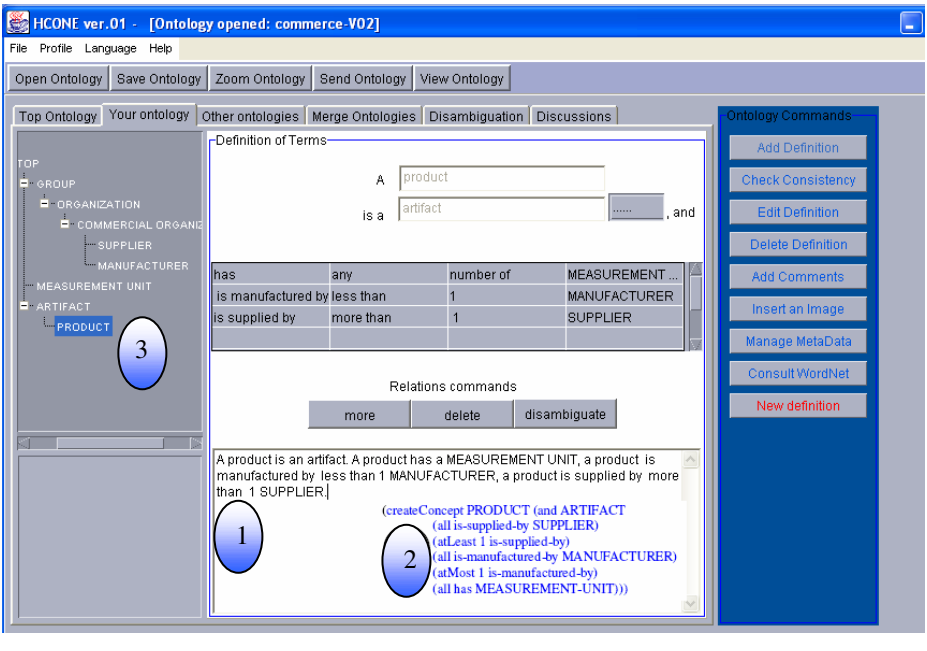

Fig. 2. HCONE support for the specification of the concept "Product": 1) natural language, 2) formal, 3) graphical representation

SharedHCONE (Fig. 4.) supports sharing ontologies to group members and supports group members' participation in structured conversations about conceptualizations. This is a built-in, rather than a patched-on facility, since it has been designed in order to support people to discuss ontological aspects and incorporate their suggestions / positions to specifications, rather than being a generic argumentation or discussion facility. The aim of the system is to support users to discuss upon an ontology and its versions, agree or disagree with a version, post new versions or get others' versions to their private space (HCONE), evaluate and exploit them, and so on. The users are able to post issues, arguments and positions (i.e. ontology versions) following a variation of the IBIS model (Issue-Based Information System), proposed by Kunz and Rittel [10]. Performing dialogue acts, users construct a discourse graph that is presented in the form of a threaded discussion. The discussion is based on three main abstractions, namely *issue*, *position* and *argument*. An issue represents a decision problem, the position is the statement that resolves the issue, and the argument either supports or objects position. These abstractions are related by predefined relationships, as it is shown in Fig. 3.

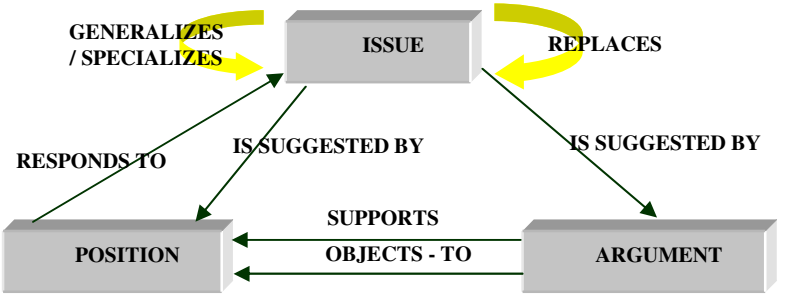

Fig. 3. The SharedHCONE discussion model

For a better understanding, we have replaced the term *position* of the IBIS model by the term *version*: Thus, making a position it is the same as posting a new version of an ontology. We have also limited the IBIS model to seven relationships between the abstractions mentioned: zero or more versions of an ontology may provide a solution for an issue raised. Each such version can be supported or objected by zero or more arguments. Also an issue can suggest a new version, or an issue can be the generalization or specialization of another issue. Furthermore, an argument can raise an issue. It is important to notice that an argument can be posted without having to support or reject an ontology version, and a version does not have to be an answer to an issue. These relationships support the modeling of the discussion in a more natural way. The user can compare any two versions of the same ontology using HCONE's version management functionality that is integrated to the system.

The SharedHCONE functionality:

- Enables criticism, identifying possible opportunities for members' collaboration Encourages feedback among community members
- Overcomes deadlocks within problematic situations that arise in ontology specification
- Supports evaluation of developed ontologies
- Provides an additional ontology versioning mechanism that records motivation behind changes (Fig. 4.)

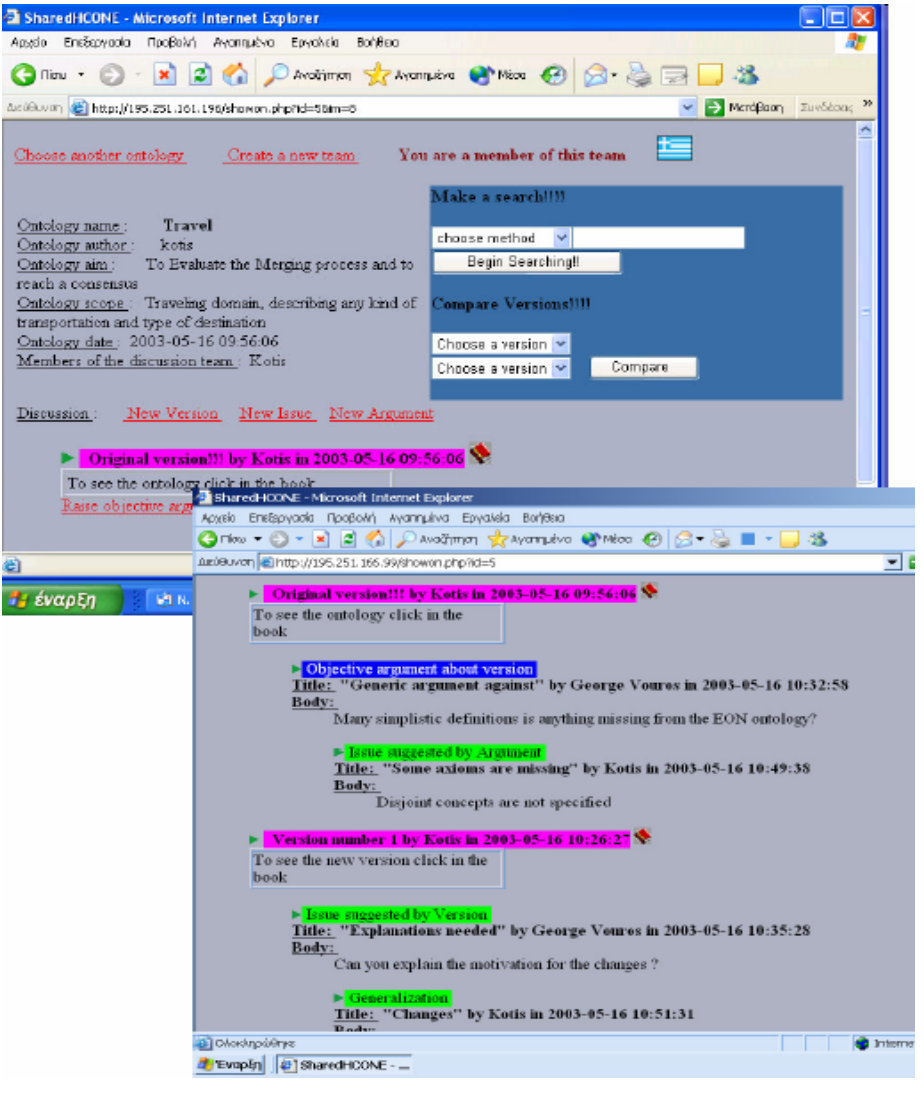

Fig. 4. Structured discussion upon ontology versions

Concluding the above, HCONE and SharedHCONE provides facilities for (a) users to improvise their conceptualizations, (b) consult generic ontologies that provide important semantic distinctions, (c) manage different versions of their ontologies, tracking the differences between the versions, (e) track the generalization/ specialization of an ontology during ontology development, (d) get proper consultation from machine exploitable/ readable lexicons by mapping concepts' meaning to word senses, (e) merge ontologies and further manipulate merged conceptualizations, and (f) share their ontologies with groups of co-workers, following a structured conversation towards agreeing in domain conceptualization.

4 HCOME: A Human-Centered Methodology to Ontology Development

As already pointed, the ultimate goal in ontology engineering is the development of commonly agreed and understandable ontologies for the effective management of knowledge in a community of knowledge workers. In order to reach this point of agreement in a community of people that share the same information needs, ontology management tasks must be integrated within the loop of information design and information exploitation [1].

Table 1. HCOME methodology phases to ontology development

Ontology life-cycle phases	Processes	Tasks
Specification	Define goals and scope, find knowledge sources	▪ discuss requirements (S) ▪ produce documents (S) ▪ identify collaborators, ▪ specify the scope, aim of the ontology (S)
Conceptualization	Acquire knowledge	▪ import from ontology libraries (P) ▪ consult generic top ontology (P) ▪ consult domain experts by discussion (S)
	Develop & Maintain Ontology	▪ improvise (P) ▪ manage conceptualizations (P) ▪ merge versions(P) ▪ compare own versions (P) ▪ generalize/specialize versions (P) ▪ add documentation (P)
Exploitation	Use ontology	▪ browse ontology (P) ▪ exploit in applications
	Evaluate ontology	▪ initiate arguments and criticism collaboratively (S) ▪ compare others' versions (S) ▪ browse/exploit agreed ontologies (S) ▪ manage the recorded discussions upon an ontology (S)

Table 1 summarizes the phases and tasks that knowledge workers perform in the HCOME methodology. These tasks are performed in a loop, until a consensus has been reached between knowledge workers. These tasks are either performed in worker's personal *space* (marked in Table 1 with a P), or they are performed in the *shared space* provided by a collaborative ontology engineering environment such as SharedHCONE (marked in Table 1 with an S). A worker can initiate any task in his personal or shared space, or take part to a shared task that has been initiated by other members of the community.

The initiating tasks of the methodology are included in the "Specification" phase of the ontology lifecycle, and they can be performed within the shared space in collaboration to other community members. The most important tasks are performed in the "Conceptualization" phase. The worker can choose any of the tasks supported by HCONE, or a combination of them, in order to develop an ontology. In the exploitation phase ontologies can be exploited and collaboratively evaluated: Users may raise new issues, form arguments for/against an issue or for/against a specific version of the ontology (i.e. a position) and form new positions (i.e. new ontology versions), feeding again the ontology development loop.

The following paragraphs discuss the major tasks of the HCOME methodological approach to ontology development, starting from the early stages of the ontology lifecycle. It must be strongly emphasized that the approach is iterative and continuous. To devise and maintain living ontologies in evolving domains and open environments, such as the Semantic Web, this iteration is rather necessary to keep on *"for ever"* until the aim for the development of ontologies is obsolete, i.e. until there is not reason for their existence [9].

As it is known, effectiveness and efficiency during the application of methodologies increase significantly through tool support [4]. HCOME is supported by the use of HCONE and SharedHCONE. Specifically, all the tasks of the HCOME phases in both, the personal and shared spaces are supported by the HCONE and the SharedHCONE tools.

4.1 Building Personal Ontologies

During the HCOME specification phase, a team of collaborators can agree on the aim and the scope of a new ontology following an argumentation dialogue in SharedHCONE. Having agreed on that, HCONE supports them to specify their conceptualizations (in their personal spaces), hiding low-level implementation details, enabling them to express subtle ontological distinctions, complying at the same time with formal constraints of specifications [7].

Specifically, HCONE, following the What-You-See-Is-What-You-Meant [11][7] knowledge editing paradigm, supports workers to specify their conceptualizations using the full expressive power of a description logic language, without dealing with low-level implementation details. While users specify the definition of a concept, they get feedback that reflects the definition of the corresponding concept in natural language. Typical tasks that workers may perform when defining a concept include concept and roles mapping to word senses through lexicon consultation, and checking for concepts' definition consistency.

Collaborators can store/manage different versions of their personal ontology, compare any ontology to other ontologies that are considered to be similar and merge relevant/similar ontologies. To support them to perform these tasks, HCONE provides seamless access to advanced services supported by description logics. These services include concepts' mapping to word senses, automatic concepts' classification, concepts' definitions consistency checks (e.g. between a concept and its subsumers) and detection of concepts' definitions differences. Feedback from these reasoning services is constantly provided during ontology development/ management and is of high significance.

Collaborators may also follow a deductive approach to concepts' specifications by elaborating a generic top ontology. In this case, concepts' definitions can be checked for their semantic validity against generic conceptualizations by means of the consistency checking mechanisms provided by the representation and reasoning system. In doing so, the construction of domain specific ontologies is speed-up and guided by the semantic distinctions and ontological principles of the generic ontologies consulted.

Critical to the ontology development process is the lexicons consultation task. Through lexicon consultation, collaborators are guided to the consensual definition of terms, guided to follow well-established norms and practices in the community they are exercising their practice (e.g. by consulting a terminological lexicon or a thesaurus) or in the wider context (e.g. by mapping their conceptions to the appropriate word senses in a lexical database). Lexicon consultation can be supported in any of the following three ways: (a) by mapping concepts definitions to word senses in a machine readable/exploitable lexicon through the concept's meaning mapping process, (b) by formally complying with generic ontological commitments of top level ontologies or (c) by simply consulting lexicons and other ontologies.

4.2 Exploiting and Sharing Ontologies

Having developed their personal ontologies, collaborators may use them within their work setting, and/or share them with colleagues in order to be further discussed, exploited and evaluated. In this context *the exploitation* of an ontology version that has been developed by a colleague is seen as part of the ontology development life-cycle since this process provides feedback for the conceptualizations developed. The need to achieve a common understanding about the working domain, push inevitably ontologies to the shared space [1].

The shared space supports people to devise ontologies conversationally, reaching a consensus on a domain conceptualisation, discuss ontological aspects and incorporate their suggestions into positions about concepts' specifications. The shared space tasks support contextualization of the built ontologies in communities' working practices and experiences, criticism and evaluation of the built artifacts, identification of possible opportunities for community members' collaboration, as well as overcoming deadlocks within problematic situations that arise in ontologies specification.

A shared space contains those ontologies that are conversationally constructed and for which the corresponding group has not reached an agreement. An *agreed space* is part of this shared space and contains those ontologies for which a group has reached an agreement. Any community member can post a new ontology to the shared space, specifying the subject, scope and aim for building this new shared ontology. At the same time, any collaborator can download to her personal space an ontology version developed by other community members, re-specify, change, enrich, compare, merge it with her own, exploit it, and send it back to the shared space, posting new issues, arguments and so forth.

To support the above, the typical process in SharedHCONE goes as follows: Having publicized an ontology, all community members receive a notification by e-mail. The body of this e-mail message provides details about this new ontology and points to the community members that they can become part of the discussion group within a number of days. Being members of the group, it is assumed that community members have already agreed on the importance of the shared ontology, and commit to take part in the upcoming discussion. Any group member can raise issues and arguments concerning the new ontology through an argumentation dialogue. Having all group members agreed on a specific version of the ontology, the ontology "moves" to the agreed space. For the seamless notification of community members about the discourse status, an e-mail notification manager sends each new discourse object to all community members via e-mail.

Users can intervene at any point in the discussion by performing any legal discourse act and can also inspect any ontology version. Inspecting an ontology, users may browse the ontology tree and get the natural language description of any concept. Furthermore, they can inspect the differences between two ontology versions through a formal comparison service. Following the threaded discussion and being able to inspect the differences between the different versions of the same ontology, people can track the rationale behind each version.

5 Conclusion

This paper presents the HCOME decentralized methodology for ontology engineering, which is supported by HCONE and SharedHCONE prototype systems. These systems support the personal and shared tasks of the HCOME ontology engineering methodology, respectively. Tools' key features, functionalities, technical details and screenshots can be found at http://www.samos.aegean.gr/icsd/kkot/HCONEweb and in [7]. Evaluation of the HCONE methodology has been carried out using these prototypes, in comparison to OntoEdit (http://www.ontoprise.de/products/ontoedit) and Protégé-2000 (protege.stanford.edu) supported methodology. The community for performing this evaluation was comprised by graduate students of our department. Their feedback was encouraging to continue our efforts. However, further evaluation of the methodology is needed in different working settings.

References

1. Vouros A. G.: Technological Issues towards Knowledge Powered Organizations. Knowledge Management Journal, Vol 7, No. 1 (2003)
2. Giboin A., Gandon F., Corby O., and Dieng R.: Assessment of Ontology-based Tools: Systemizing the Scenario Approach. In Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002, pages 63-73. Siguenza, Spain, 30th September (2002)
3. Fernandez-Lopez M.: Overview of methodologies for building ontologies. Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (1999)
4. Sure Y.: A Tool-supported Methodology for Ontology-based Knowledge Management. ISMIS 2002, Methodologies for Intelligent Systems (2002)
5. Staab S, Studer R., Schnurr H., and Sure Y.: Knowledge Processes and Ontologies. IEEE Intelligent Systems, pages 26-34, January/February (2001)
6. Fernandez-Lopez M.: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. ONTOWEB Consortium for Ontology-based information exchange for knowledge management and electronic commerce. IST-2000-29243, Deliverable 1.4 (2002)
7. Kotis K., Vouros G.: Human Centered Ontology Management with HCONE. Proceedings of the IJCAI-03 Workshop on Ontologies and Distributed Systems (2003)
8. Cook S. D. N., and Brown, J. S.: Bridging epistemologies: The generative dance between organizational knowledge and organizational knowing. Organizational Science, 10:381-400 (1999)
9. Stojanovic L. and Motik B.: Ontology Evolution within Ontology Editors. In Proceedings of OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002, pages 53-62. Siguenza, Spain, 30th September (2002)
10. Kunz W. and Rittel H. W. J.: Issues as elements of information systems. Technical Report S-78-2, Institut fur Gundlagen Der Planung I.A, Universitat Stuttgart (1970)
11. Power R., Scott D., and Evans R.: What You See Is What You Meant: direct knowledge editing with natural language feedback. ITRI Technical Report No. 97-03, University of Brighton (1997)