Knowl Inf Syst (2006) 10(1): 109–131
DOI 10.1007/s10115-005-0227-4

Knowledge and Information Systems

REGULAR PAPER

Konstantinos Kotis · George A. Vouros

Human-centered ontology engineering: The HCOME methodology

Received: 7 October 2004 / Revised: 23 March 2005 / Accepted: 9 April 2005 /
Published online: 9 September 2005

Abstract The fast emergent and continuously evolving areas of the Semantic Web and Knowledge Management make the incorporation of ontology engineering tasks in knowledge-empowered organizations and in the World Wide Web more than necessary. In such environments, the development and evolution of ontologies must be seen as a dynamic process that has to be supported through the entire ontology life cycle, resulting to living ontologies. The aim of this paper is to present the *H*uman-*C*entered *O*ntology Engineering *Me*thodology (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 activities and shape their information space by being actively involved in the ontology life cycle. The paper also demonstrates the *H*uman *C*entered *ON*tology Engineering *E*nvironment, HCONE, which can effectively support this methodology.

Keywords Semantic web · Ontology engineering · Human-centered computing · Knowledge management

1 Introduction

Ontologies are formal conceptualizations of domains, describing the meaning of (abstract and concrete) domain aspects by means of concepts and their interrelations [1]. Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and evolution of the Semantic Web. Ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through

K. Kotis (✉) · G. A. Vouros
Department of Information and Communications Systems Engineering, University of the Aegean, Samos, Karlovassi, 83200 Greece
E-mail: {kkot, georgev}@aegean.gr

practice and interaction among community members, binding the knowledge processes of creating, importing, capturing, retrieving and using knowledge [19].

The incorporation of ontology engineering tasks in knowledge-empowered organizations can prove to be a hindrance if not done in a way that is seamless to the day-to-day activities of the organization members [24]. Traditionally, knowledge engineers develop the ontologies that are required by knowledge workers, and provide these ontologies for exploitation. However, due to constant updating, changing and evolution of ontologies, there must be a close collaboration between knowledge engineers and knowledge workers,[1] requiring the active and decisive involvement of the latter in all stages of the ontology engineering processes [20].

In conjunction to the above, the proliferation of the Semantic Web aims at explicating the meaning of the worldwide available information. Ontologies constitute the backbone of this effort, providing machine-exploitable semantic information for the Knowledge Web. But how could we possibly populate the Semantic Web, enabling information providers to attach semantic information to every published information item, shaping also their consensual conceptualisations? Current efforts aim to support the automatic and manual structuring of web pages using ontologies, in conjunction to ontology learning and enrichment. In conjunction to these approaches, we believe that people must be empowered to manage their ontologies shaping their *common* information space.

Several methodologies [4, 6, 15, 18, 21, 22] have been proposed for the engineering of ontologies within a knowledge management setting. From feasibility analysis and identification of goals, to requirements' specification, implementation, evaluation and maintenance of ontologies, the ontology life cycle must be clearly defined and further supported by ontology engineering tools [21]. Early ontology engineering methodologies have been developed to support knowledge engineering tasks, starting from the feasibility analysis and requirements specification of a knowledge-based system and proceeding to its deployment. Most recent methodological approaches [4, 21] focus on the application-driven engineering of ontologies, supporting the introduction of ontology-based knowledge management systems. According to all these approaches, the development, maintenance and evolution of ontologies is primarily a 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. Only very recent ontology engineering methodologies concern about the role of knowledge workers and about their active involvement in the ontology life cycle [15]. However, no methodology fully supports the decisive involvement of knowledge workers in the ontology life cycle. Specifically, existing ontology engineering methodologies:

1. Emphasize on the role of knowledge engineers (or of a centralized structure governed by a control board, whose tasks is to collect and analyse change requests to a single-shared ontology) in the ontology life cycle.
2. Do not support the seamless integration of ontology engineering tasks to the day-to-day activities of the knowledge workers. This should involve an

[1] A knowledge worker is any member of an information production–exploitation community. Such community may involve workers within an organization, or World Wide Web users with common interests.

in-principle approach to deal with specific needs and constraints of knowledge workers.

3. Are not fully supported by tools that empower knowledge workers to actively participate in the ontology life cycle.

In contrast to the existing methodologies, we propose a human-centered approach to the engineering of ontologies, called HCOME 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 are put in the context of workers' experiences and working settings, as an integrated part of workers' 'knowing' process [2, 24]. Leveraging the role of knowledge workers by empowering them to participate actively in the ontology life cycle, the human-centered approach entails the development of tools that provide greater opportunities for them to manage and interact with their conceptualisations in a direct and continuous mode.

To further support our conjecture for the need of human-centered methodological approaches let us consider the following 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, possibly working in collaboration with knowledge engineers.
- *Scenario No 2:* In a knowledge use process, a worker browses, recalls existing knowledge items, and processes 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 conceptualisations at the same time.

Summarizing the above, this paper emphasizes on accentuating the role of knowledge workers in shaping their information through their active and decisive involvement in ontology engineering tasks. Section 2 of this paper presents issues concerning the engineering of ontologies in the context of communities of knowledge workers, and Sect. 3 points on generic requirements for the human-centered engineering of ontologies. Section 4 presents the Human Centered Ontology Engineering Methodology (HCOME), and Sect. 5 presents the overall architecture and functionality of a human-centred environment for ontology engineering (HCONE), which have been designed and implemented to support the deployment of the HCOME. Section 6 presents related work and results concerning a preliminary evaluation of HCOME. Finally, Sect. 7 concludes the paper.

2 Engineering 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, conceptualizations evolve in communities as part of *knowing* [2]. 'Knowing' is about interactions with the world as well with members of communities, with the aim to create new knowledge. Personal knowledge is created through practice, whereas group knowledge is created through interaction between community members.

Therefore, the manipulation of conceptualizations in the context of communities of knowledge workers involves their development, evaluation, exploitation and their continuous evolution as part of *knowing*, as humans achieve their goals in their working contexts. In particular it involves:

- *The development of personal conceptualizations.* People develop their own conceptualizations that may either make explicit (e.g. by formalizing terms in a special language, 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 conceptualizations.* Developing commonly agreed and understandable domain conceptualizations 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 [2], not only results in exchanging knowledge but also in generating new knowledge.
- *The evaluation and exploitation of conceptualizations.* Exploitation and evaluation of conceptualizations as part of the day-to-day practice of communities can be considered only as part of knowing. Conceptualizations are put in practice and 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 (e.g. exploit them in a real working setting). Evaluation can result in new meanings since concepts are seen under the light of new experiences and evolving contexts.

Impediments for knowledge workers to participate actively in formalizing their conceptualizations include their unfamiliarity with formal representation languages and knowledge engineering principles/methods, as well as with methods and techniques for constructing and synthesizing ontologies. Existing ontology engineering environments and methodologies have been designed and implemented for the knowledge engineer, concentrating mostly on the ontology development process. This leads communities of knowledge workers to develop semantically poor thesauruses for their domains, or even abandoning the trial for semantically annotating their resources, since in most of the cases they are not willing to pay the costs implied by employing knowledge engineering resources.

Currently, the engineering of ontologies follows methodologies that are analogous to the user-centered software engineering approaches: Knowledge engineers with the participation of domain experts (artefact's users) gather the requirements for the ontologies, and either by using a knowledge representation language or by using an ontology engineering environment, design and implement prototype ontologies. These ontologies are commented by experts to be further improved so as to asymptotically represent a consensual conceptualisation of the domain at hand. This contrasts to the old-fashioned technology-centered design approach in which developers gather requirements, design and implement an artefact and leave users to cope with what they have produced. However, even with the involvement of users, the approach to ontologies engineering is still mostly machine-oriented: Knowledge engineers deal with these artefacts at the symbol level, mediating between domain conceptualisations and their formal representations, which cannot be further manipulated or even (in some cases) be inspected by domain experts. This leads to a machine-oriented, knowledge engineer-centered ontology engineering approach. It relies heavily on de-contextualized principles of engineering ontologies, with formal means for capturing subtle domain aspects, but does not deal with the way people develop their conceptualisations in the context of their day-to-day activities, individually or conversationally with colleagues.

For knowledge workers to be empowered to engineer ontologies, they must participate actively in the ontology engineering processes in collaboration with knowledge engineers. In particular, they must be enabled to develop, evaluate and further evolve ontologies individually and conversationally with their co-workers, according to their cognitive abilities, skills, knowledge base and context of work. Ideally, this must happen in the background of the day-to-day knowledge intensive activities of workers, in a seamless way to their working practices, and so that the semantic validity of specifications is assured.

3 The human centered engineering of ontologies

Taking into account the above mentioned issues, constraints and considerations to the engineering of ontologies, and to:

(a) support the active and decisive involvement of knowledge workers in all stages of the ontology life cycle, and
(b) further empower people to engineer their ontologies shaping their information space in ways that are seamless to their day-to-day working activities.

We propose a human-centered approach to the engineering of ontologies. This is an in-principle approach to devising ontology engineering methodologies and construction supporting tools, that emphasizes at providing greater 'opportunities' for the members of knowledge management organizational units to develop their conceptualizations during their day-to-day activities [7].

Human-centered computing (HCC) intents on helping us maximizing the value of computing in society, towards a new age of human and machine symbiosis [7]. Joint research efforts on cognition, ergonomics, psychology, social sciences and information technologies contribute to the development of human-centered systems, leading to a new world of applications and services. In this new era, systems' designers have to really understand the use of systems in conjunction with the *cognitive abilities*, *contextual constraints* and *activities of users*.

It has been argued that the human-centered development of systems must take the triple of 'people-machine-context' as the unit of analysis [7]. This principle involves studying people capacities and goals, computational mechanisms and interface capabilities, all within their appropriate contexts. Context involves organizations, activities, practices, norms and constraints that are inherent in these activities, policies, procedures, as well as devices and media that people use for communicating and doing their work.

Viewing people activities for developing conceptualizations of domains within the context of organizational structures, we shall not focus (at least in the first place) on object-centered representation formalisms or on other technological/engineering issues, but put special emphasis on the physical way of interacting with these conceptualizations and on the way conceptualizations are formed by means of people interacting among themselves and practicing.

People in organizations have complimentary knowledge and share common interests and aims. Although domain or decision tasks may be well distributed among them and there may be power relations among them, knowledge workers share a common conceptualization that provides the basis for their consensual domain conceptualization. However, due to their complimentary knowledge, the domain conceptualization is considered to be distributed among them. Therefore, we consider that knowledge workers contribute to ontology specifications on an equal basis. On the other hand, concerning the cognitive and perceptual capabilities of knowledge workers, we must emphasize that they are not familiar with knowledge representation languages, ontology engineering principles and practices. Therefore, they must manipulate ontologies at a level of abstraction that is more 'convenient' for them, interacting with their conceptualizations in a direct mode.

To realize the full potential of human-centered ontology engineering we must develop ontology engineering environments that would empower *specific* persons in *specific* contexts, following well-established ontology engineering methodologies for the design of human-centered information systems (e.g. [26]). However, rather than examining a specific case for engineering ontologies, we point on important generic issues for the human-centered engineering of ontologies within organization structures. The list of issues is by no means complete. However it points on what we consider to be the most important issues that need to be considered within an ontology engineering methodology, based on the characteristics of knowledge workers that the above analysis reveals:

1. *Allow an eclectic way to the development of ontologies.* Members of communities must be allowed to follow any approach or combination of approaches for the development of ontologies, which better fits their practice, their working

norms and constraints: They may improvise by integrating concepts in a conceptual system, provide concepts with informal definitions attaching information items to them, compare, merge and refine/generalize existing ontologies.

2. *Emphasize on the need for a natural and consistent way to interact with conceptualizations.* As already pointed, the major issue for human-centered ontology engineering is that people must interact with their conceptualizations at a level of detail that is more convenient for them. Therefore, low-level implementation details of formal ontologies must be hidden from workers who do not understand knowledge representation formalisms' terminology. People must be given the power to express subtle ontological distinctions in a way that is natural to them but satisfies the formal constraints of the specifications too. This further implies that knowledge workers must be supported to develop well-formed ontologies (i.e. ontologies without inconsistencies among the defined concepts, with coherency, and well organized), as well as to compare and map existing ontologies.
3. *Provide the means for exchanging, using and evaluating ontologies conversationally.* As already pointed, shaping information synergistically is necessary, since knowledge is distributed among workers. To support conversations between individuals, a methodology must enable further criticism of the developed artefacts, encourage feedback from community members, putting ontologies in the context of knowledge workers' experiences and practice. Workers may deploy and evaluate ontologies during their day-to-day activities, raising arguments for and against specific conceptualization aspects, suggest changes, additions and refinements, and propose new ontology versions. Conversation facilities must provide the means for detecting new opportunities for collaboration, as well as for getting out of deadlocks within problematic situations that may arise during collaboration.
4. *Consider mapping of concepts' definitions to other ontologies and/or lexical resources.* The aim is to uncover the human-intended semantics [23] of the specifications for clarification and communication purposes. This supports the bridging of different perspectives about the domain and provides a critical feedback on the preciseness of specifications. Concepts' meaning mapping is important for the development of commonly agreed conceptualizations, especially in communities where people from different disciplines use the same term with different meanings or use different terms for the same concept. Furthermore, for people to get a feedback on whether the developed specifications reflect the intended meaning of terms, or to further constraint the meaning of terms, they need to map the term definitions to word senses is a precise way.

Given the above issues, we must emphasize that to empower knowledge workers to actively participate in the ontology life cycle following a human-centered methodology, the whole life cycle must be supported by appropriate tools.

4 HCOME: A human-centered methodology to ontology engineering

In this section we describe the HCOME methodology to ontology engineering that points to the way knowledge workers develop their conceptualisations continuously through the phases of requirements specification to that of ontologies' deployment, development and maintenance, exploitation and evaluation.

Table 1 The HCOME methodology phases to ontology engineering

Ontology life cycle phases	Goals	Tasks
Specification	Define aim/scope/ requirements/teams	Discuss requirements (S)
		Produce documents (S)
		Identify collaborators (S)
Conceptualisation	Acquire knowledge	Import from ontology libraries **(P)**
		Consult generic top ontology **(P)**
		Consult domain experts by discussion (S)
	Develop and maintain ontology	Improvise **(P)**
		Manage conceptualisations **(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 (S)
		Compare others' versions (S)
		Browse/exploit agreed ontologies (S)
		Manage the recorded discussions upon an ontology (S)
		Propose new ontology versions by incorporating suggested changes (S)

4.1 Description of the HCOME approach

HCOME provides clear distinction between the phases of the ontology life cycle, goals that should be achieved in each phase and tasks that can be performed so as to achieve these goals. These are summarized in Table 1.

As it is shown, these tasks are performed iteratively, until a consensus has been reached between knowledge workers. Tasks are performed by workers either individually or conversationally. In the first case, we consider that tasks are performed in the personal *space* of workers (marked in Table 1 with the letter 'P'). In the latter case, tasks are performed in an information space that a group of knowledge workers shares, i.e. in a *shared space* (marked in Table 1 with the letter 'S'). A worker can initiate any ontology engineering task in his personal or shared space, or take part to a task that has been initiated by other members of the community.

The following paragraphs discuss the major tasks of HCOME in relation to the issues described in Sect. 3.

4.1.1 Specification phase

During the HCOME specification phase, knowledge workers are joining groups that are about to develop shared ontologies. Having identified themselves within a group, during this initial phase of ontology engineering, workers are discussing requirements, produce specification documents, and agree on the aim and the scope of the new ontology. This phase may start from a worker that has already formed a conceptualisation of specific domain aspects and needs the contribution of colleagues to further develop the conceptualisation.

The 'Specification' phase of the ontology life cycle is performed within the shared space and includes:

a. The specification of the scope and aim of the ontology. This is essential in order workers to have an agreed initial reference of the way they understand the domain and the way they want initially to model it, according to their information needs.
b. An argumentation dialogue between the members of the team to obtain commonly agreed specification requirements.
c. Recording of the agreed specifications in appropriate forms and documents.

4.1.2 Conceptualisation phase

Having agreed on the scope and aim of the ontology to be developed, workers in their personal space can follow any approach or combination of approaches to the development of ontologies: They may improvise by integrating concepts in a conceptual system, provide concepts with informal definitions attaching information items to them, compare, merge and refine/generalize existing ontologies. Since the consultation of other well-known and/or widely acknowledged resources is critical to the ontology development process, the collaborators may perform this task before sharing their conceptualisations with others. Collaborators should be able to create, store, maintain, compare, merge and manage different versions of ontologies.

The 'conceptualization' phase includes the following tasks:

a. The import of existing ontologies, for the reuse of conceptualisations.
b. The consultation of generic top ontologies, thesauruses and domain resources, for better understanding and clarification of the domain conceptualisations, uncovering the informal human-intended semantics of the ontology specifications.
c. The improvising of ontologies, allowing the from-scratch development of formal ontologies based on workers' perception on the domain.
d. The mapping, merging and management of multiple versions of ontologies, supporting reuse and evolution.
e. The comparison of different versions of an ontology, for tracking ontologies' evolution and for identifying ontologies that can possibly be merged.
f. Attaching to ontology concepts information items with further comments, examples and specification details.

4.1.3 Exploitation phase

The need to achieve a common understanding about the working domain, push inevitably ontologies developed in personal spaces to the shared space [24]. Shared ontologies can be used within workers setting, in the context of specific ontology-driven applications and tasks, and exploited and evaluated conversationally. The exploitation of an ontology version that has been developed by a colleague is seen as part of the ontology life cycle because it provides feedback for the conceptualizations developed. The evaluation and further development of personal ontologies is achieved via a structured conversation, and criticism on the ontology versions posted in the shared space. The recording of this conversation enables the tracking of changes and rationale behind ontology versions, supporting the decisions on conceptualising the domain in the final ontology.

The 'Exploitation' phase includes:

a. The inspection of agreed or shared ontologies either by individuals in their personal space or by collaborators in the shared space, for reviewing, evaluating and criticizing the specified conceptualisations.
b. The comparison of shared versions of an ontology, for identifying the differences between them.
c. The posting of arguments upon versions of ontologies for supporting workers decisions, for or against specifications.

5 HCONE support to the HCOME

*H*uman *C*entered *ON*tology Engineering *E*nvironment is a prototype environment [8] that has been designed and implemented taking into account the above-pointed generic issues for the human-centered engineering of ontologies, with the aim to support HCOME. The overall architecture of HCONE is shown in Fig. 1.

Major tasks that are provided to people include the construction of ontologies in a personal space, publication of ontologies in a shared space, formation of groups for the conversational development of ontologies, and participation in argumentation dialogues for reaching consensual conceptualizations. Agreed ontologies are stored in a virtual space and can be further evolved in workers' personal space, become shared and so on.

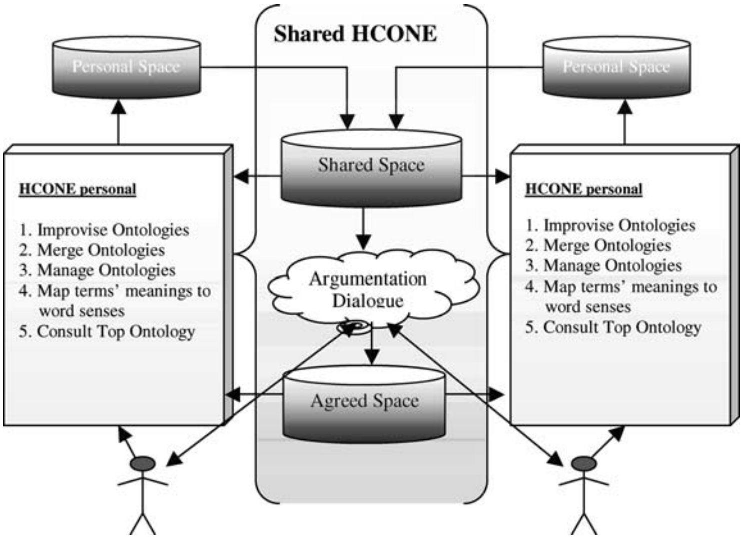


Fig. 1 HCONE architecture

HCONE is a modular and integrated environment, providing access to any functionality in any HCONE point. Doing so, workers are free to combine their own method using the environment, following an eclectic way to ontology development. 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 been shared with co-workers. In the meantime, she is trying to comply with generic ontological commitments that the group has agreed to comply with, whereas in another slice of her work she is trying to merge her ontology with an ontology issued by a co-worker.

Towards these targets, HCONE provides facilities for users to:

(a) decide on the scope and aims of ontologies;
(b) improvise domain conceptualizations by constructing personal ontologies;
(c) consult generic ontologies that provide important semantic distinctions and organization principles;
(d) manage different versions of their ontologies, tracking the differences between the versions;
(e) get proper consultation from machine exploitable/readable lexicons by mapping concepts' meaning to word senses;
(f) merge ontologies and further manipulate merged conceptualizations;
(g) share their ontologies with groups of co-workers; and
(h) follow argumentation dialogues towards consensual domain conceptualizations.

A very important built-in HCONE functionality, which, due to its importance, is separately discussed in the next section, concerns sharing ontologies with group members and supporting group members' participation in structured conversations about conceptualizations. This is a built-in, rather than a generic argumentation or threaded patched-on discourse facility, since it has been designed to support people to discuss ontological aspects and incorporate their suggestions/positions to specifications. This functionality, with the distinct name SharedHCONE, supports contextualization of the built ontologies, criticism and evaluation of the built artefacts, identification of possible opportunities for community members' collaboration, and helps workers overcome deadlocks within problematic situations that arise in ontologies specification.

Later in this article, we provide a description of each HCONE functionality in relation to the three phases of the HCOME approach to ontology engineering.

5.1 Specification phase

The HCOME specification phase is supported by SharedHCONE. Among others, SharedHCONE provides support to the tasks of identifying a group of collaborators and of specifying conversationally the scope and aims of the ontology. SharedHCONE realizes a shared space where any group member can post a new ontology, specifying the subject, scope and aim for constructing this new ontology. Having publicized an ontology, all community members receive an e-mail message that provides details about the new ontology, as these are specified by the initiator of the discussion, and calls community members to become members of the discussion group. Being members of the group, it is assumed that community members have already agreed on the importance of the new ontology, and commit to take part in the upcoming discussion.

Following an argumentation dialogue during the specification phase, workers, aiming to reach an agreement on the aim and scope of an ontology, are discussing requirements, produce specification documents, raise issues concerning the scope and the aim of the ontology, make proposals, state arguments for and against these proposals. The argumentation model of SharedHCONE is further explained in the paragraphs that follow.

5.2 Conceptualization phase

Having specified the aim and scope of an ontology, collaborators can either construct their own version of that ontology in their personal space using HCONE, and then post it to the shared space for further discussion and evaluation using SharedHCONE. Furthermore, workers can download shared or agreed ontology versions from the SharedHCONE shared space to their personal space, to elaborate and evaluate them, and produce new versions. These versions can be posted again to the shared space for further discussion and evaluation.

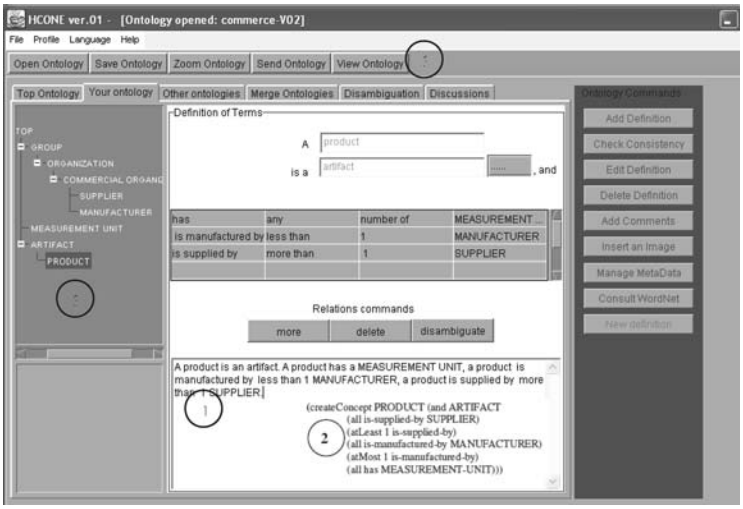
To support the conceptualization phase, HCONE provides functionalities for constructing and refining ontologies, for merging existing ontologies and for consulting generic lexicons and ontologies.

5.2.1 Constructing and refining ontologies

HCONE supports people to interact with their conceptualizations hiding low-level implementation details, enabling them to express subtle ontological distinctions, complying at the same time with formal constraints of specifications. Formal specification of ontologies in HCONE is done in the NeoClassic Description Logic [14].

In particular, workers communicate their conceptualizations using the full expressive power of NeoClassic, without dealing with low-level implementation details. As is it shown in Fig. 2, this is achieved by following canned natural language dialogues based on a slight variation of the What-You-See-Is-What-You-Meant (WYSIWYM) [17] knowledge-editing paradigm. Although users specify the definition of a concept, they get a feedback text that reflects the definition of the corresponding concept (part 1 of Fig. 2) whereas in the background a formal Description Logic definition is constructed, which can be further converted to standard schemes. Typical tasks that users may perform when defining a concept include concept and roles mapping to word senses through lexicon consultation (described in the next paragraphs), checking for concepts' definition consistency, and deletion/addition of relations with other concepts.

Additionally, users can store and manage in their personal space different versions of an ontology. The versioning mechanism is not currently very sophisticated. It requires from users to assign specific suffixes on ontology file names to indicate that these are versions of a source initial ontology. However, a more advanced version management mechanism, which will be presented subsequently,



A product
is a artifact ....., and

has	any	number of	MEASUREMENT ...
is manufactured by	less than	1	MANUFACTURER
is supplied by	more than	1	SUPPLIER

A product is an artifact. A product has a MEASUREMENT UNIT, a product is manufactured by less than 1 MANUFACTURER, a product is supplied by more than 1 SUPPLIER

```
(createConcept PRODUCT (and ARTIFACT
        (all is-supplied-by SUPPLIER)
        (atLeast 1 is-supplied-by)
        (all is-manufactured-by MANUFACTURER)
        (atMost 1 is-manufactured-by)
        (all has MEASUREMENT-UNIT)))
```

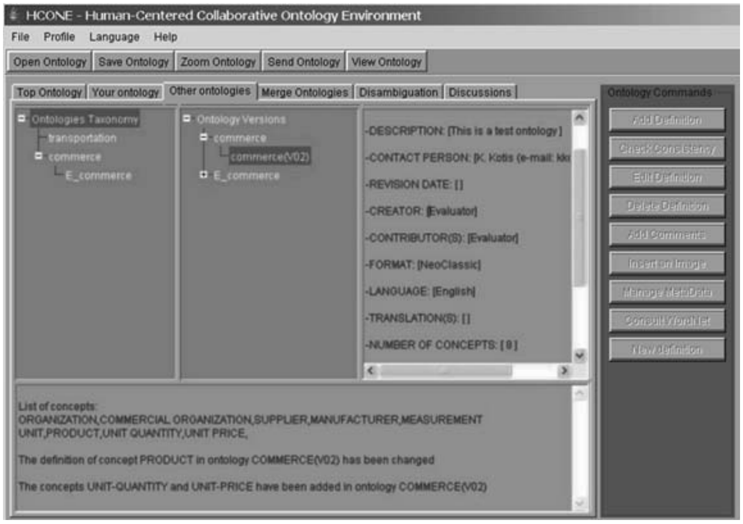
Fig. 2 HCONE concept specification: (1) natural language, (2) formal and (3) graphical representation

is provided by SharedHCONE. Additional functionality on ontology comparison enables users to identify differences between ontology versions. The comparison functionality provides the means for comparing any pair of ontologies, indicating similarities and differences for concept /role definitions and pointing on concepts and roles that are specified only in one of the ontologies compared.

To support users to construct and compare ontologies, HCONE provides seamless access to reasoning services provided by Description Logics. These services include 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 to users during ontology construction/ management and is of high significance. For instance, as Fig. 3 shows, although tracking the differences between two versions of the same ontology, as well as during merging, reasoning services identify semantically equivalent definitions, allowing proper handling of versions and disallowing semantic errors.

5.2.2 Consultation of generic ontologies

Knowledge workers may 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, since it is guided by the semantic distinctions and ontological principles of the generic ontologies consulted. HCONE, in its current status, provides the option of importing any generic ontology, given that it has been implemented in NeoClassic. We have contacted experiments with a generic ontology for terminological



-DESCRIPTION: [This is a test ontology]

-CONTACT PERSON: [K. Kotis (e-mail: kk

-REVISION DATE: []

-CREATOR: [Evaluator]

-CONTRIBUTOR(S): [Evaluator]

-FORMAT: [NeoClassic]

-LANGUAGE: [English]

-TRANSLATION(S): []

-NUMBER OF CONCEPTS: [8]

List of concepts:
ORGANIZATION,COMMERCIAL ORGANIZATION,SUPPLIER,MANUFACTURER,MEASUREMENT UNIT,PRODUCT,UNIT QUANTITY,UNIT PRICE,

The definition of concept PRODUCT in ontology COMMERCE(V02) has been changed

The concepts UNIT-QUANTITY and UNIT-PRICE have been added in ontology COMMERCE(V02)

Fig. 3 HCONE versions management

resources that incorporates EuroWordNet and SIMPLE semantic commitments [25].

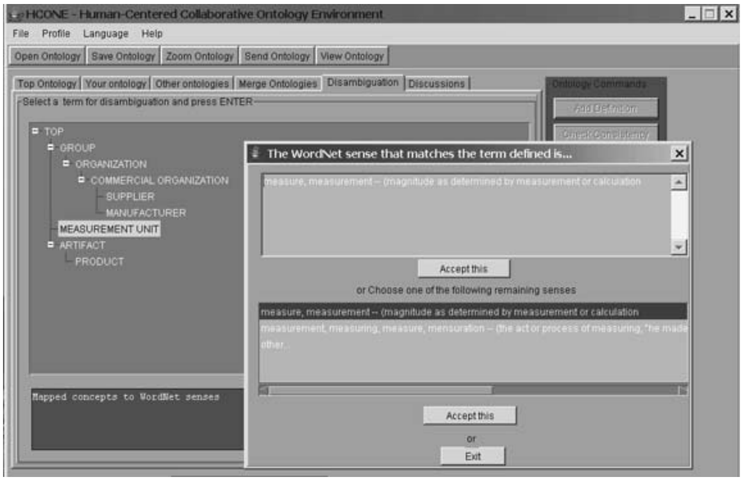
5.2.3 Consultation of lexicons

Critical to the ontology specification process is the lexicons consultation process. Through lexicon consultation, users 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). In HCONE, lexicon consultation can be supported in any of the following two ways: (a) by mapping concepts definitions to word senses of a machine-readable/exploitable lexicon, (b) by simply consulting lexicons and other resources.

As already pointed, the mapping of concepts to word senses is critical to uncover the human-oriented-intended semantics of concepts specifications, clarifying humans intentions and enabling them to bridge different perspectives in conceptualizing a domain.

To support the mapping of concepts to word senses, HCONE uses the Latent Semantic Indexing (LSI) technique [3]. LSI is a vector space technique for information retrieval and indexing. LSI has been chosen since it provides high precision in mapping concepts meaning to word senses, which, based on our experiments, can be up to 70% [9].

LSI takes a large matrix of term-document association data and constructs a semantic space where terms and documents that are closely associated are placed near one another. Currently, HCONE exploits the WordNet [12] lexical database to match descriptions of terms provided by the user with word senses in WordNet. LSI in this case constructs a semantic space of terms and word senses.



HCONE - Human-Centered Collaborative Ontology Environment

Select a term for disambiguation and press ENTER

The WordNet sense that matches the term defined is...

measure, measurement -- (magnitude as determined by measurement or calculation

or Choose one of the following remaining senses

measure, measurement -- (magnitude as determined by measurement or calculation

measurement, measuring, measure, mensuration -- (the act or process of measuring; "he made other...

Mapped concepts to WordNet senses

Fig. 4 Mapping the meaning of the concept MEASUREMENT UNIT to WordNet senses

As Fig. 4 shows, word senses that closely match the lexicalization and meaning of a given concept are shown to the user in an ordered way (best matches first). In case no WordNet senses are found that correspond to user's specifications, then the user is asked to provide a word sense that is close to his specifications. Further experiments [9] have recently shown that a semiautomatic method can be applied to minimize user involvement in validating mappings to WordNet, keeping the precision still in high percentages.

It must be noticed that beyond concepts' meaning mapping, concepts' relations (i.e. description logic role names) are also mapped to word senses. Currently, the results of role name mapping are exploited only for the generation of the feedback text during concepts' definition.

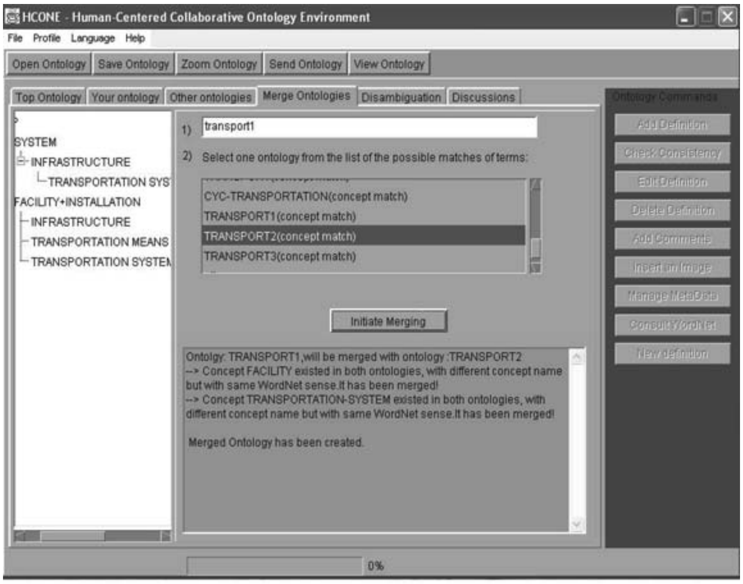
5.2.4 Merging ontologies

In conjunction to the formal definition of concepts, having mapped terms to word senses, HCONE has gathered the informal intended meaning of terms. Having done that, concepts in different ontologies can be associated among themselves, resulting in effective merging of ontologies.

The *merging* process takes for input two ontologies from the same domain and produces a new ontology [9]. The merging functionality exploits all the available types of knowledge available: Linguistic, structural, domain knowledge, and also some matching heuristics. Recently, due to the importance of this task in the ontology engineering life cycle, we have developed methods for the semiautomated mapping and merging of ontologies [9].

Figure 5 shows the integrated merging functionality of HCONE, presenting the results of the merging process to the end user as a feedback report that describes the merging actions that the system has performed.

The resulted merged ontology is stored in users' personal space and can be further manipulated according to users' preferences.



1) transport1

2) Select one ontology from the list of the possible matches of terms:

```
Ontolgy: TRANSPORT1,will be merged with ontology :TRANSPORT2
--> Concept FACILITY existed in both ontologies, with different concept name
but with same WordNet sense.It has been merged!
--> Concept TRANSPORTATION-SYSTEM existed in both ontologies, with
different concept name but with same WordNet sense.It has been merged!

 Merged Ontology has been created.
```

Fig. 5 The integrated to HCONE merge functionality

5.3 Exploitation phase

Having constructed a new version of an ontology in his personal space, a collaborator may post it to the shared space using the HCONE *post* functionality. To manage ontologies in the shared space, SharedHCONE provides facilities that assist collaborators to produce an agreed ontology, which will be uploaded to the SharedHCONE agreed space.

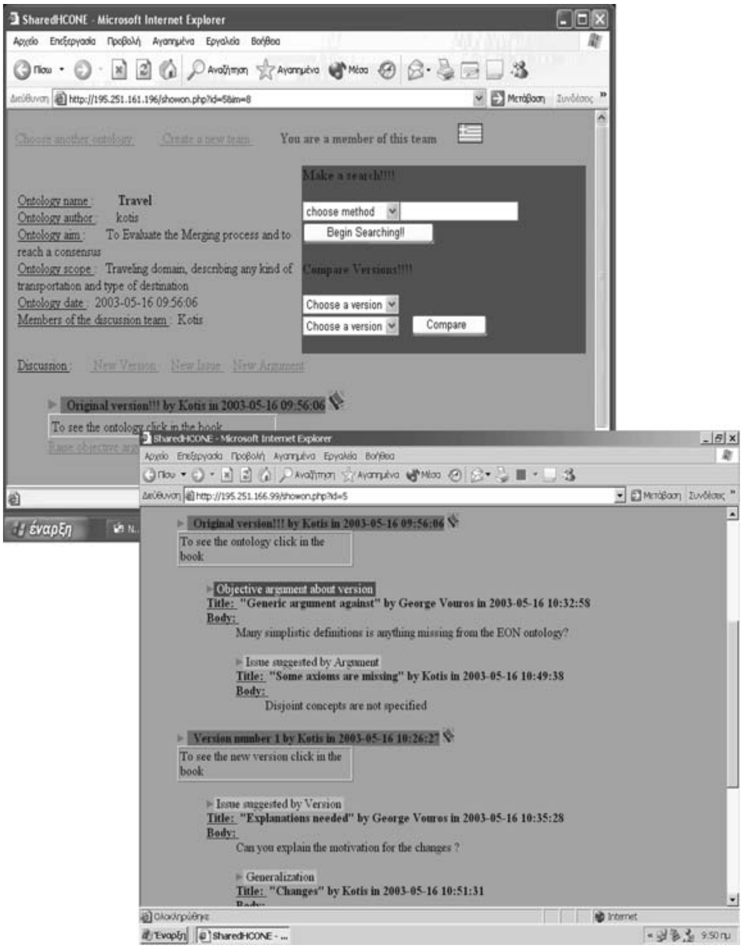
5.3.1 Sharing ontologies

SharedHCONE supports contextualization of the constructed ontologies in organizational working practices and workers experiences, criticism and evaluation of the built artefacts, identification of possible opportunities for collaboration among workers, as well as overcoming deadlocks within problematic situations that arise in conceptualizing a domain.

Through an argumentation dialogue, any group member can raise issues, propose solutions and post arguments concerning an ontology that has been shared (Fig. 6). Specifically, the argumentation dialogue follows a version of the IBIS model [10], which is based on three main abstractions.

- *Issue*, representing a problem to be solved;
- *Position*, which can be either a solution to an issue raised, or a new version of the discussed conceptualisation;
- *Argument*, which can be either a ‘supporting argument’, speaking in favour of a position, or an ‘objecting argument’, speaking against a participant’s position.

Objects (issues, positions, arguments) created can be linked according to the generic model shown in Fig. 7. The users are able to post issues, arguments and


SharedHCONE - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα Μέσα
Διεύθυνση http://195.251.161.196/showon.php?id=5&im=8
Μετάβαση Συνδέσεις
Choose another ontology
Create a new team
Make a search!!!!
choose method
Begin Searching!!
Compare Versions!!!!
Choose a version
Choose a version
Compare
Discussion :
New Version
New Issue
New Argument
Raise objective ar
έναρξη
SharedHCONE - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα Μέσα
Διεύθυνση http://195.251.166.99/showon.php?id=5
Μετάβαση Συνδέσεις
Ολοκληρώθηκε
Internet
Έναρξη
SharedHCONE - ...
9:50 πμ

You are a member of this team

Ontology name : Travel
Ontology author : kotis
Ontology aim : To Evaluate the Merging process and to reach a consensus
Ontology scope : Traveling domain, describing any kind of transportation and type of destination
Ontology date : 2003-05-16 09:56:06
Members of the discussion team : Kotis

Original version!!! by Kotis in 2003-05-16 09:56:06
To see the ontology click in the book

Original version!!! by Kotis in 2003-05-16 09:56:06
To see the ontology click in the book

Objective argument about version
Title: "Generic argument against" by George Vouros in 2003-05-16 10:32:58
Body:
Many simplistic definitions is anything missing from the EON ontology?

Issue suggested by Argument
Title: "Some axioms are missing" by Kotis in 2003-05-16 10:49:38
Body:
Disjoint concepts are not specified

Version number 1 by Kotis in 2003-05-16 10:26:27
To see the new version click in the book

Issue suggested by Version
Title: "Explanations needed" by George Vouros in 2003-05-16 10:35:28
Body:
Can you explain the motivation for the changes ?

Generalization
Title: "Changes" by Kotis in 2003-05-16 10:51:31
Body:

Fig. 6 Structured discussion upon ontology versions

positions following a variation of the IBIS model. Performing dialogue acts, users construct a discourse graph that is presented in the form of a threaded discussion. According to the SharedHCONE argumentation model, an issue represents a decision problem. A position is a statement that resolves an issue and can be a posting of a new version of an ontology. The argument either supports or objects a position. These abstractions are related by predefined relationships, as it is shown in Fig. 7. We have also limited the IBIS model to seven relationships between the abstractions mentioned: zero or more positions or 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 position/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 a position or ontology version, and a

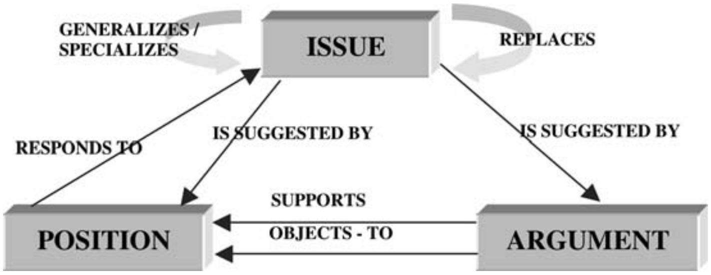
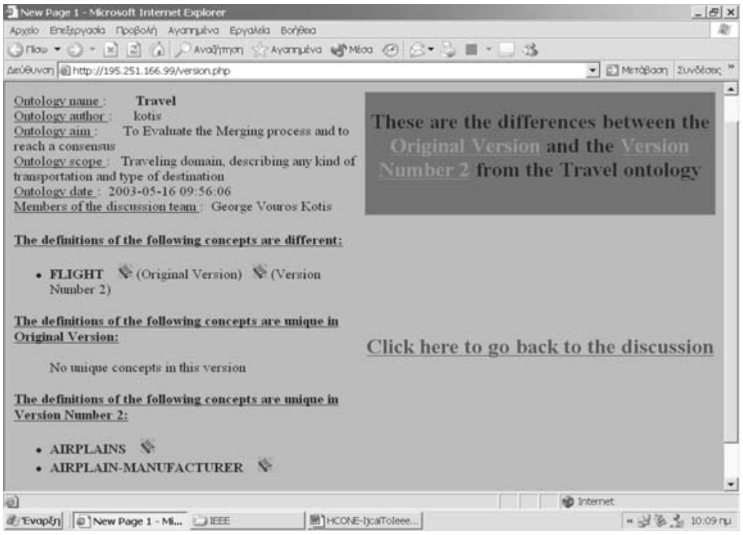


Fig. 7 SharedHCONE discussion model


New Page 1 - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα Μέσα
Διεύθυνση http://195.251.166.99/version.php
Μετάβαση Συνδέσεις
Internet
Έναρξη New Page 1 - Mi... IEEE HCONE-IjcaiToleee...
10:09 πμ

Ontology name : Travel
Ontology author : kotis
Ontology aim : To Evaluate the Merging process and to reach a consensus
Ontology scope : Traveling domain, describing any kind of transportation and type of destination
Ontology date : 2003-05-16 09:56:06
Members of the discussion team : George Vouros Kotis

These are the differences between the Original Version and the Version Number 2 from the Travel ontology

The definitions of the following concepts are different:

- **FLIGHT** (Original Version) (Version Number 2)

The definitions of the following concepts are unique in Original Version:

No unique concepts in this version

Click here to go back to the discussion

The definitions of the following concepts are unique in Version Number 2:

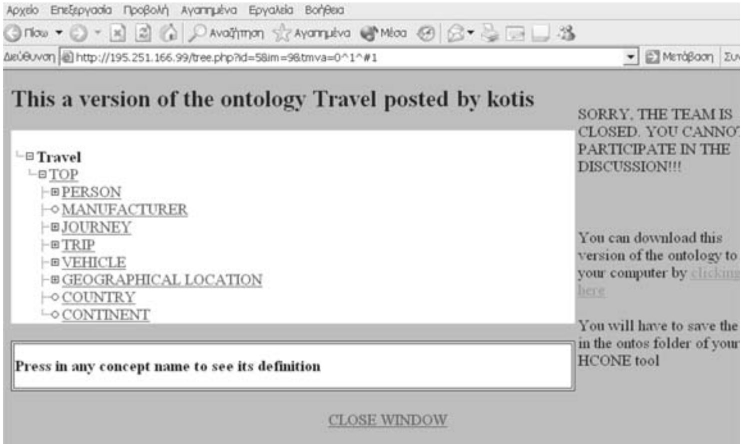
- **AIRPLAINS**
- **AIRPLAIN-MANUFACTURER**

Fig. 8 Ontology versions comparison

position or version does not have to be an answer to an issue. These relationships support the modelling of the discussion in a more natural way.

For the seamless notification of community members about the discourse status, the 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 and download in their personal space any ontology version posted at any point in the discussion using the corresponding SharedHCONE functionality (Fig. 8).

Inspecting an ontology, as Fig. 9 shows, 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 that is based on the same techniques as the ones included in personal HCONE comparison functionality. Figure 8 depicts the results provided by the comparison service.



http://195.251.166.99/tree.php?id=5&im=9&tmva=0^1^#1

This a version of the ontology Travel posted by kotis

- ⊟ Travel
  - ⊟ TOP
    - ⊞ PERSON
    - ○ MANUFACTURER
    - ⊞ JOURNEY
    - ⊞ TRIP
    - ⊞ VEHICLE
    - ⊞ GEOGRAPHICAL LOCATION
    - ○ COUNTRY
    - ○ CONTINENT

Press in any concept name to see its definition

SORRY, THE TEAM IS CLOSED. YOU CANNO PARTICIPATE IN THE DISCUSSION!!!

You can download this version of the ontology to your computer by clicking here

You will have to save the in the ontos folder of your HCONE tool

CLOSE WINDOW

Fig. 9 Browse and inspect ontology versions

Following the threaded discussion and being able to inspect the differences between two versions of the same ontology, people can track the rationale behind each version and more important, behind the agreed ontology.

Concluding the above, workers, being able to download ontologies in their personal space and further study and exploit them in their working setting, and being supported to conversationally review ontologies, are decisively and actively involved in the evaluation and evolution of ontologies.

6 Related work and evaluation

Work on ontology engineering methodologies has been extensively discussed and compared in [11, 13, 16]. It must be pointed that only very recently ontology engineering methodologies did mature, incorporating tasks concerning reusability and maintenance.

Lopez [11] uses the IEEE software engineering framework to compare methodologies. This comparison motivated the proposal of the METHONTOLOGY methodology. This is a well-known second-generation methodology which has been considerably influenced by software engineering methodologies. The methodology proposes an evolving prototyping ontology life cycle, proposing reusability, version management and project management activities. Although METHONTOLOGY proposes that ontology engineering must be done at the knowledge level rather than at the symbol/implementation level, there are no tool-supported activities that would enable this to happen. Furthermore, OntoWeb [13] extensively describes and compares existing methodologies based also in the IEEE software engineering framework, taking into account the construction strategies used, ontology engineering processes, methodologies' use, and technological support. In [16] authors establish a framework that specifies the activities that compose an ontology engineering process. This framework is more appropriate than the IEEE software engineering framework used in [11], because it fits more appropriately in the ontology engineering process. In particular [16] identify the need to consider issues that are related to distributed, dynamic and uncontrolled environments such as the Web or modern knowledge-based organizations in which ontologies are being built and deployed. Based on this work, a recent methodological proposal to ontology engineering called DILIGENT [15] moves towards specifying activities for engineering distributed, loosely controlled, and evolving ontologies.

In contrast to other methodologies to ontology engineering, the DILIGENT methodology is very close to HCOME. We would say that both efforts move towards the third-generation of ontology engineering methodologies. Specifically, both methodologies consider distributed settings and thus emphasize on issues concerning collaboration and argumentation. In addition, both methodologies consider evolving ontologies, pointing on the importance of argumentation dialogues, versions management and merging of ontologies. Finally, they both consider that engineers must somehow minimize their involvement (control) during the engineering of ontologies, passing control to domain experts. However the two approaches tackle this issue differently: DILIGENT is clearly depended on decisions of a control board, whereas HCOME is a tool supported methodology that empowers knowledge workers to be actively and decisively participate in the ontology life cycle. In the following paragraphs we point on specific points in which HCOME differs to DILIGENT.

Although DILIGENT users are allowed to follow an eclectic way to the management of ontologies, the completion of some tasks is partially dependent on the decisions of a board of experts. For instance, knowledge workers cannot use any updated version of an ontology for exploitation, evaluation or further refinement, unless the board decides that the changes made in the shared ontology are valid. Although this eases the management of ontology versions, there is no evidence on how DILIGENT provides the power to knowledge workers to develop and evaluate multiple shared versions of an ontology.

In addition, although the exchange and discussion on ontologies is facilitated by DILIGENT, users are constrained in this process since they are not allowed to inspect, conversationally exploit and evaluate ontologies that have been contributed by others. This prohibits the direct manipulation of conceptualisations developed by co-workers, hindering the bridging of different perspectives and conversationally overcoming deadlocks in conceptualizing the domain.

To our knowledge, apart from HCOME, there is no other tool-supported methodology that places knowledge workers at the center of the ontology life cycle, accentuating their role and active involvement in every task of the ontology management processes. In particular, HCONE, the *H*uman *C*entered *ON*tology Engineering *E*nvironment supports all phases and tasks of HCOME, providing functionalities for ontology requirements specification, conceptualization, exploitation and evaluation.

An extensive description and comparison of existing ontology engineering tools have been carried out in latest reviews (e.g. [13]). Although other tools consider some of the key issues presented in Sect. 3 for the human-centered engineering of ontologies, only HCONE provides access in a wide range of necessary functionalities. The major emphasis of HCONE is the minimization of the upfront knowledge engineering effort that a worker should put to accomplish a particular HCOME task. For instance, authoring an ontology from scratch and merging it with an existing version of another ontology is a task that can be performed in almost every ontology engineering environment, but the way these tasks have been realized in HCONE differ significantly from their realization in other tools: HCONE balances the degree of users' involvement in ontology engineering with a formal representation language and advanced reasoning services, guaranteeing validity of specifications. An important functionality of HCONE is the Shared-HCONE that concerns the support for conversational development and evaluation of ontologies. This functionality, as already pointed, supports essential tasks of HCOME such as requirements specification, evolution and exploitation of ontologies, as well as tracking the rationale behind different ontology versions.

The HCOME methodology has been evaluated as part of a graduate project in the Information and Communications Systems Engineering Department of the University of the Aegean. Although the evaluation of the methodology itself is biased by the HCONE tool and the way it supports the methodological tasks, the questionnaires and discussions with the students reveals the importance to consider the issues for human-centered engineering of ontologies. Students that took part in the evaluation had already gained background knowledge on ontologies in the frame of their MSc course. Although it was their first time that they had to develop an ontology, they found it easy to follow the tasks of HCOME, which where distinctively and fully provided by HCONE. The requirements specification for the ontology to be developed was given to them at the start of their project. Students developed their own conceptualisations in their personal space. Their participation in the exploitation phase was of critical importance for students with less developed skills on ontology engineering, since they received consultation from more skilled students, and in some cases they received developed versions of an ontology as a starting point to their own development process. Participation of their supervisor in this phase, playing the role of a knowledge engineer, gave them hints and comments on their conceptualizations, adding extra value to the task of collaborative construction and evaluation of ontologies. During the development process, the comparison and merging tasks saved them a lot of time due to their value in reusing information. This, in conjunction to the conversational development of ontologies, has helped them to understand and bridge the gaps between different conceptualizations. This is essential for co-workers to reach an agreed conceptualisation.

7 Concluding remarks

In this paper we have conjectured that the engineering of ontologies must be seen as a dynamic and collaborative process that in most of the cases starts with an initial specification of requirements and/or an initial rough ontology, which is later revised, refined and filled in with the details by knowledge workers.

Ontology evolution has to be supported through the entire life cycle, resulting to a living ontology [20] with the decisive and active participation of knowledge workers in collaboration with knowledge engineers. Towards this aim we have presented a new tool supported approach to the engineering of ontologies called HCOME. This has been designed and evaluated in the AI Lab of the Information and Communication Systems Eng. Department of the University of the Aegean.

The Human Centered Ontology Engineering Environment (HCONE) that supports the HCOME approach provides a modular environment that supports an eclectic approach to ontology engineering, focusing on these functionalities to a greater extent than other tools and empowering knowledge workers to develop their formal conceptualizations either individually or conversationally (with co-workers). According to Gibbon et al. [5] there must be a balance between usage and technology, and between formality and informality. HCONE, following the human-centered approach, emphasizes on integrating ontology engineering environments with knowledge workers practices considerably, enabling knowledge workers to interact directly with their conceptualizations at a high-level of abstraction.

Feedback from the evaluation conducted was encouraging to continue our efforts. Further evaluation of the methodology is needed in wider and more varied work settings.

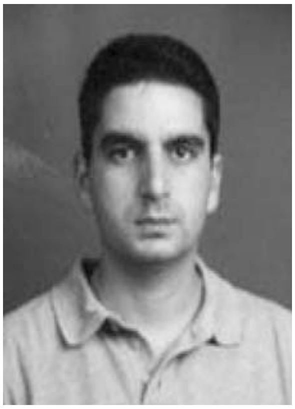
References

1. Chandrasekaran B, Josephson JR, Benjamins VR (1999) What are ontologies, and why do we need them? IEEE Intell Syst 14(1):20–26
2. Cook SDN, Brown JS (1999) Bridging epistemologies: The generative dance between organizational knowledge and organizational knowing. Organ. Sci 10:381–400
3. Deerwester S, Dumais TS, Furnas WG, Landauer KT, Harshman R (1990) Indexing by Latent Semantic Analysis. J Am Soc Inform Sci 46(6):391–407
4. Fernández-López M, Gómez-Pérez A, Pazos-Sierra A, Pazos-Sierra J (1999) Building a Chemical ontology using METHONTOLOGY and the ontology design environment. IEEE Intell Syst 14(1):37–46
5. Giboin A, Gandon F, Corby O, Dieng R (2002) 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, pp 63–73
6. Grüninger M, Fox MS (1995) Methodology for the design and evaluation of ontologies. In: Proceedings of IJCAI'95, workshop on basic ontological issues in knowledge sharing, Montreal Canada
7. Hoffman R, Hays P, Ford K (2002) The triples rule. IEEE Intell Syst 17(3):62–65
8. Kotis K, Vouros AG (2003) Human centered ontology management with HCONE. In: Proceedings of ODS workshop, IJCAI-03, Acapulco
9. Kotis K, Vouros AG, Stergiou K (2004) Capturing semantics towards automatic coordination of domain ontologies. LNAI, Vol. 3192, Springer-Verlag
10. Kunz W, Rittel HWJ (1970) Issues as elements of information systems. Technical Report S-78-2, Institut fur Gundlagen Der Planung I.A, Universitat Stuttgart
11. Lopez MF (1999) Overview of methodologies for building ontologies. In: Proceedings of the IJCAI-99 workshop on ontologies and problem-solving methods
12. Miller G, Beckwith R, Fellbaum D, Gross D, Miller K (1995) Five papers on WordNet. CSL Report 43. Cognitive Science Laboratory, Princeton University
13. ONTOWEB (2002) Consortium for ontology-based information exchange for knowledge management and electronic commerce. IST-2000-29243, Work package 1, Deliverable 1.3: A survey on Ontology Tools
14. Patel-Schneider PF, Abrahams M, Resnick AL, McGuinness LD, Borgida A (1996) Neoclassic reference manual: Version 1.0. Artificial Intelligence Principles Research Department, AT&T Bell Labs
15. Pinto S, Martins J (2004) Ontologies: How can they be built? Knowledge and Inform Syst 6:441–464
16. Pinto S, Staab S, Tempich C (2004) DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely controlled and evolvInG Engineering of oNTologies. ECAI2004, Spain

17. Power R, Scott D, Evans R (1997) What you see is what you meant: direct knowledge editing with natural language feedback. ITRI Technical Report No. 97-03, University of Brighton
18. Schnurr HP, Sure Y, Studer R (2000) On-to-knowledge methodology – Baseline version. In: Executive summary, On-To-Knowledge EU-IST-1999-10132 Project Deliverable D15
19. Staab S, Studer R, Schnurr H, Sure Y (2001) Knowledge processes and ontologies. IEEE Intell Syst 16(1):26–34
20. Stojanovic L, Motik B (2002) Ontology evolution within ontology editors. In: Proceedings of ontoweb-sig3 workshop at the 13th international conference on knowledge engineering and knowledge management EKAW 2002. Siguenza, Spain, pp 53–62
21. Sure Y (2002) A tool-supported methodology for ontology-based knowledge management. In: ISMIS 2002, Methodologies for Intelligent Systems
22. Uschold M (2003) Where are the semantics in the Semantic Web? AI Mag 24:25–36
23. Uschold M, King M (1995) Towards a methodology for building ontologies. In: Workshop on basic ontological issues in knowledge sharing
24. Vouros AG (2003) Technological issues towards knowledge powered organizations, Know. Manage J 7:114–127
25. Vouros AG, Eumeridou E (2002) Simple and EuroWordNet: towards the prometheus ontological framework. Terminology 8:245–278
26. Zahng J, Patel VL, Johnson KA, Smith JW, Malin J (2002) Designing human-centered distributed information systems. IEEE Intell Syst 5:42–47

Author Biographies

George VOUROS (B.Sc. Ph.D.) holds a B.Sc. in Mathematics, and a Ph.D. in Artificial Intelligence all from the University of Athens, Greece. Currently he is a Professor and Head of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece, Director of the AI Lab and head of the Intelligent and Cooperative Systems Group (InCoSys). He has done research in the areas of Expert Systems, Knowledge management, Collaborative Systems, Ontologies, and Agent-based Systems. His published scientific work includes more than 80 book chapters, journal and national and international conference papers in the above-mentioned themes. He has served as program chair and chair and member of organizing committees of national and international conferences on related topics.



Konstantinos KOTIS (B.Sc. Ph.D.) holds a B.Sc. in Computation from the University of Manchester, UK (1995), and a Ph.D. in Information Management from University of the Aegean, Greece (May, 2005). Currently, he is a member of the Intelligent and Cooperative Systems Group (InCoSys) and director of the Information Technology Department of the Prefecture of Samos, Greece. His research and published work concerns Knowledge management, Ontology Engineering and Semantic Web. He has lectured in several IT seminars and has served as member of program committees in international workshops.