Journal of Universal Knowledge Management, vol. 1, no. 3 (2006), 199-216
submitted: 5/10/06, accepted: 15/11/06, appeared: 28/12/06 © J.UKM

Consensus Building in Collaborative Ontology Engineering Processes

Stelios Karapiperis
(University of Piraeus, Greece
skarapas@yahoo.gr)

Dimitris Apostolou
(University of Piraeus, Greece
dapost@unipi.gr)

Abstract: Ontology development is time and money consuming as well as an error-prone process; the need for an embedded mechanism that evaluates quality and acceptance of the resultant collaborative ontology is apparent. Existing tools and methodologies lack consensus building mechanisms that must be employed in order for a team to cooperate and agree on the design and deployment process of a shared one. In this paper we describe a collaborative methodology for ontology development that supports a team to reach consensus through iterative evaluations and improvements. In every cycle of the iterative process, the structure of the collaborative ontology is revised and evolved. Finally, the process terminates when the participants have no more critiques and objections. We illustrate the methodology by creating an ontology for an airline training centre using the PROTÉGÉ software tool.

Keywords: Semantic Web, Ontology Engineering
Categories: I.2.4, I.2.11

1 Introduction

Ontology [Uschold, 96] is the term used to describe the concepts and any relations between them that exist in a domain of interest which can be used as the infrastructure for facilitating the creation of a common and shared understanding. The structure of the ontology includes a set of explicit predefined concepts with their detailed characteristics and the inter-relations between the concepts that can be used to interpret a specific domain. With the use of this structure we aim to create a common vocabulary of terms and a representation of the knowledge that exists in a certain domain thus allowing us to store, retrieve and disseminate it. A shared representation of a domain’s knowledge can improve the quantity and quality in communication between and among humans and machines.

Currently available methodologies focus mostly on the centralized development of static ontologies by knowledge engineers and domain experts. In order for ontology to be truly a shared conceptualization of a domain’s knowledge, support is needed for the team to cooperate and reach consensus. In this paper, we describe a collaborative methodology based on the ontology development framework proposed in [Holsapple, 02] that supports domain experts (the team) to reach consensus through iterative evaluations and improvements.

According to [Holsapple, 02], the collaborative approach phases' to ontology design are:

1. *The preparatory phase* that defines the design criteria for the ontology for guiding and assessing its success;
2. *The anchoring phase* that includes the development of the initial ontology that will seed the next phase;
3. *The iterative improvement phase* that revises the ontology until consensus is reached through a consensus building technique and
4. *The application phase* that applies the final ontology structure to the case in order to demonstrate its uses.

Our proposed methodology complies with the above phases and starts with the definition of the criteria and the design of an initial ontology from the team coordinator, according to the general requirements of the team. In this phase we exploit the ontology building steps described in [Noy, 01]. The initial ontology is then depicted in the evaluation sheets and these sheets feed the first cycle of ontology questioning and evaluation from the participants through voting in a Nominal Group Technique (NGT) manner.

The NGT is a formal technique and an effective way to make pooled judgments and decisions in-groups that meet face-to-face. The first cycle ends when every item in the evaluation sheets reflects the participants' view through any necessary action (addition, deletion or substitution) [Porzel, 04]. The mutual agreed actions and the new structure is then depicted in new evaluation sheets and a new round starts again as described above until the participants have no more objections and everybody agrees that consensus has been reached about the resultant, shared ontology.

In this paper, the agreed ontology is formally depicted using the PROTÉGÉ tool that has the capability of translating the structure of the ontology into the formal Web Ontology Language (OWL). In 2002, the Web Ontology Working Group of W3C adopted OWL[1] as the official language for ontology creation, overcoming expression problems of the older W3C standard, the Resource Description Framework Schema (RDF-S)[2]. The structural units of the OWL are:

- *Classes* that explicitly describe concepts of the domain of interest,
- *Data type properties* that describe various attributes of the classes,
- *Object properties* that represent relations between classes and
- *Individuals* that are class instantiations, representing objects of the domain of interest.

Throughout our paper, we use examples from an ontology created to represent the knowledge of the training centre (TC) of an airline company.

2 Related Work

Our work is mainly related to three different research areas: Ontology engineering tools, ontology engineering methodologies, consensus building techniques.

[1] http://www.w3.org/TR/owl-features/

[2] www.w3.org/RDF

2.1 Ontology Engineering Tools

According to [Youn] there is a variety of ontology tools that support among others, the creation, deployment and maintenance of ontologies. These tools, which are listed below and described briefly, are PROTÉGÉ, OilEd, Apollo, RDFedt, OntoLingua, OntoEdit, WebODE, KAON, ICOM, DOE, WebOnto, Medius Visual Ontology Modeler, LinkFactory and K-Infinity.

- PROTÉGÉ is a graphical user interface tool that enables the user to create and maintain ontologies, entry forms and enter data. Besides the standard configuration, there are various plug-ins extensions in order to add tables, diagrams, queries etc. This tool provides a uniform GUI which is composed of overlapping tabs for compact presentation. For example, there is a Classes tab to define classes and the class hierarchy, a Properties tab for creating data type and object properties and a Forms tab for easy creation of entry forms. There is multi-user support capability for reading/editing the same database.
- A simple editor that allows the user to create and maintain ontologies is the OilEd tool. It has been built by the University of Manchester and its main intention is to provide a simple and free editor that supports, and stimulates interest in, DAML+OIL. It lacks capabilities for development large-scale ontologies or integration or versioning.
- Apollo is a user friendly application for ontology creation and its class's system is modelled according to the Open Knowledge Base Connectivity (OKBC) protocol for accessing Knowledge Bases stored in and allowing interoperability between different Knowledge Representation Systems. Every new ontology inherits the default ontology with primitive classes like: Boolean, integers, float, string etc. Apollo does not support multi-user capabilities or collaborative processing but it features strong type consistency checking.
- One way to build fast and easy complex and structured RDF (and RSS) documents is by using the RDFedt tool. It has very limited consistency check, it works only on windows platform and it does not support multi-user capabilities. This tool can support Dublin Core Element Set and RSS modules like aggregation, annotation etc.
- The OntoLingua tool provides a distributed collaborative environment with World Wide Web services in order to create, modify and use ontologies, like the Ontology Editor that allows these actions through the web browser. Among other features there is multi-user support via write-only locking and user access levels. This tool has a substantial user community around the world.
- OntoEdit is an ontology-engineering environment supporting the following three different phases of the ontology engineering cycle: initial kick-off, refinement and evaluation phase. It also keeps representation language as neutral as possible for the underling concepts, relations and axioms thus making possible easy transformation to all major ontology representation languages. In its professional edition it supports a collaborative working environment and ontology libraries.
- WebODE is based and created to provide technological support for methodology and it is an advanced ontological engineering workbench for all the major ontology related activities, for easy development of ontology-based applications

and finally for integrating ontologies in information systems. It has multi-user support by synchronization, authentication and access restrictions per user groups and it supports collaborative processing.

- KAON_is an open-source framework for easy building ontology-based business applications. It has an ontology editor module (OI-modeler) for ontology creation and maintenance and a second module, KAON Portal, for navigating and searching ontologies through Web browsers. It provides multi-user support by concurrent access control with transaction-oriented locking and rollback mechanisms.
- The purpose of the ICOM CASE tool is to provide a simple and free mean in order to demonstrate and stimulate interest in knowledge representation based technologies through ontologies. It is a standard Java 1.2 application and it operates both on Windows and Linux platforms. It supports consistency checking via the FaCT reasoner but it does not provide multi-user capabilities.
- Another simple ontology editor is the Differential Ontology Editor which is based in the methodology proposed by *Bruno Bachimont.* The DOE does not intent to be a full ontology environment and therefore it does not support many activities in ontology construction but rather a complement of other editors. It has neither multi-user nor collaborative capabilities.
- The WebOnto tool developed by the Knowledge Media Institute of the Open University in England and by using a graphical interface aims to be an easy-to-use yet powerful tool for creating and maintaining ontologies. It has collaborative capabilities and supports multi-users through global write-only locking with change notification.
- Medius Visual Ontology Modeler is an add-in to Rational Rose Enterprise Edition and therefore it's a UML-based ontology modelling tool and it operates only as Rational Rose plug-in. It supports collaborative and concurrently development through the network-based environment that it provides, but it has limited consistency checking capabilities.
- The LinkFactory tool, which was originally designed for very large medical ontologies and it has a 3-tier client-server architecture: the client application to manage the ontology, the server interface to cope with user requests and data layer for accessing the user and maintenance information and the ontology contents. It facilitates collaborative environment with author privileges and auditing specific to concept hierarchies.
- Finally, the K-Infinity tool developed by Intelligent Views is a knowledge editor and it provides broad support for object-oriented knowledge modelling. Through Knowledge Builder, the main component of K-Infinity tool, users can manage objects and relations as well as relations between objects. It has consistency check and collaborative features.

To sum up, a number of ontology engineering tools have features that allow for and facilitate concurrent development of the ontology by a group of experts-users. These tools that can support collaborative functions mainly provide locking mechanisms for each ontology resource (e.g. class/concept, individual/instance, property/relation) to ensure safe development conditions [Sure, 02].

2.2 Ontology Engineering Methodologies

Tempich et al. reports in [Tempich, 05] that most of the established methodologies for ontology engineering focus on ontologies developed in a centralized manner. Existing methodologies typically divide the whole process of ontology engineering into several steps in order to create a new ontology, or to reuse others or re-engineer them. On the other side, Holsapple et al. in [Holsapple, 02] focus on the collaborative aspect of the ontology engineering process, in a way that we have outlined in the introduction. Moreover, the DILIGENT methodology [Tempich, 05] provides support for argumentation by employing an issue-based argumentation approach. An integrated approach for distributed argumentation that permits ontology evolution is also HCOME [Kotis, 05].

2.3 Consensus Building Techniques

Consensus building or collaborative problem solving is a process that allows members of a team with interests in the problem or issue to work together developing a mutually acceptable solution. This is happening when there is no domain expert that can solve the problem and a group decision should be made or when we want to choose among different solutions. When everybody agrees and is satisfied with the final proposal then we say that we have reach consensus.

Consensus is a form of collaborative decision making in which every member of the team is freely and equally consulted, can vote and eventually accepts the final decision. The greatest benefit of collaboration is greater quality of solutions, because the produced solution incorporates the different perspectives of the team members rather than those of a single person, and thus is less subjective.

The Nominal Group Technique (NGT) is an effective way to generate a large quantity of new ideas or prioritize alternative solutions. It is designed to allow every party of the team to participate freely without influence from others and express his/her idea through a voting process. The whole process is transparent to everybody so all participants have a clear understanding of what is going on.

The steps to reach consensus using the NGT method are:

1. The facilitator presents or writes down the issue but does not propose solutions, so the participants should be assured that there are many alternatives.
2. The participants producing new ideas by brainstorming or prioritize alternative ones and they write down all their responses.
3. Each member in a round robin format shares one idea until all ideas have been listed.
4. The facilitator associates each idea to a number and writes them down on the flip chart and all of them are fully discussed and explained so everybody have a clear meaning of them.
5. Each member ranking the ideas by writing down the ranking mark and the associated number and the leader writes everything down.
6. All ideas receiving a rank are listed so all can view them. The higher the total for each idea, the higher the rank and each idea is designated accordingly.

7. The participants rank again the designated ideas and the final rankings are discussed so the team reach consensus.

The NGT method is relatively easy and inexpensive to apply but the facilitator's role is very important and s/he must keep everybody encouraged to participate. It also takes less time to produce results than the Delphi method without compromise to quality and quantity.

The Delphi method is another consensus building technique and it has been widely used to generate forecasts in technology, education and other fields. It was developed in order to make discussion between experts possible without permitting a certain social interactive behaviour as it happens during a normal group discussion and which may hamper opinion forming. This group decision process allows geographically dispersed experts to deal systemically and anonymously with a complex problem. It comprises a series of questionnaires sent electronically or via mail to a pre-selected group of experts. Anonymity, controlled feedback and statistical response characterize Delphi. The Delphi steps are:

1. The mediator forms a panel of experts to participate in the process. The experts remain anonymous.
2. S/he develops a questionnaire and sends it to all participants (mail or electronic mail).
3. When the questionnaires are returned, s/he performs an analysis of the responses.
4. If a consensus has not been reached, s/he provides a synthesized version of the responses back to the participants, and starts again from step 2 with a second questionnaire, otherwise s/he develops the final report.

The pros of Delphi include that the experts do not have to physically meet; the anonymity helps preventing domineering personalities while participants have more time to consider alternative solutions. The cons include that the results can be influenced by the mediator, are dependent on the level of expertise and wording of the questionnaire and the whole process is time and sometimes money consuming.

3 A consensus-based Ontology Engineering Approach

Our proposed collaborative methodology includes:

a) The definition of the ontology's design criteria.

b) The development of the initial version of the ontology that will feed the next phase (evaluation phase) while being aware and complying with the design criteria.

c) The iterative process of the evaluation of the initial ontology until everybody in the team of users agrees with the structure. In this phase, every cycle of the iterative evaluation will revise and evolve the structure of the ontology.

d) The application of the final ontology.

In the following we explain in more detail the four phases of the proposed methodology.

3.1 Phase 1: Definition of the design criteria

The design criteria will guide us throughout the whole process and will evaluate objectively the design of the ontology. As in [Uschold, 96] these criteria are clarity, coherence – consistency, extensibility, minimal ontological commitment and encoding bias. In our case the encoding bias criterion is not applicable because we will not use directly any formal representation language but via the PROTÉGÉ editing tool that automatically encodes our effort into OWL.

Clarity implies that any term we use in the structure should be clear and with no ambiguity to every party involved in the design. In other words, every class and every property should have the same semantic meaning to all participants, in order for everybody to communicate effectively, sharing the same vocabulary. Examples can be used to clarify any doubtful class or property, e.g. “seminar” and “course” is the term used to describe the action of an instructor and “instructor” has the same semantic meaning like “teacher” or “professor” (let the latter excuse us).

Coherence – consistency means that the ontology must have no contradictory statements, even if natural language is to be used. In the formal axioms, logical consistency must be applied as well. For example, the statement that the class “Instructor” and the class “Trainee” must have no individual in common or there must be no “Instructor” that is also “Trainee”, implies consistency that should be followed. This statement in particular was a predefined requirement stated by the users.

Extensibility of the ontology implies that the underling classes should permit any changes as extensions for customization without the need of revising existing definitions. For example, after six months of use the team that designed the ontology decided that they wanted to extend the structure and store one more class, “Administrators” which specialises the “Employee” class.

Minimal ontological commitment is the balance and the compromise that the designer has to take between a general model and a very detailed one, resulting in a vocabulary that might contain needless and redundant classes and properties. The structure of the ontology must reflect only the intended domain and make as few claims as possible without restricting future extensions.

3.2 Phase 2: Designing the initial ontology

To develop the initial version of the ontology we opted for the main ontology building steps described in [Noy, 01]. Before starting designing the initial ontology for our case study, we had an extensive discussion with the participants of the team about their work and about how a new knowledge organisation schema, the ontology, could cope with the knowledge that flows within the Training Centre. This briefing was aiming to prepare the participants for the whole venture, to leverage their active involvement as well as to familiarise them with the possible applications and benefits of the ontology.

Members of the team were carefully selected in order to complement each other and represent diverse viewpoints. With this in mind we chose the director of the TC, which has an integrated perception of the business workflow inside the TC and other individuals from different ranks, positions and experiences. The number of participants was chosen to be 7. Our experience has shown that in order to gain quantitative and qualitative results the participants should be not more than 9 or10, as the development might lead to fruitless and time consuming efforts, whereas less than 4 participants would probably produce a biased and limited ontology. In table 1 we present the basic characteristics of the team members.

RANK	POSITION	EXPERIENCE
Director	Supervising the product of the TC and defining the long term targets.	16 yrs
Managers (2)	Coordinate and evaluate the instructors' work, Control the supporting material (manuals, TC facilities, computers, classrooms and other)	11 and 9 yrs respectively
Supervisor	Schedule and control the instructors' plan	6 yrs
Instructors (3)	Carry out the TC plan	7, 6 and 4 yrs

Table 1: Basic characteristics of the team members

Knowledge management with ontology-based systems adds value to knowledge and to knowledge managers as well. The vision of an ontology-based knowledge management system and the associated benefits were shared among the participants. This had a positive impact on the participants' motivation to participate in the collaborative effort to develop the ontology. Suggestions about the domain and the scope that the ontology will cover (step 1), the enumeration of the important terms (step 3) and the definitions of the class properties, all came from participants' contributions.

3.2.1 Step 1: Determine the domain and scope of the ontology

What is the domain that the ontology will cover?
The domain of the ontology will be the activities of the airline TC**.** The main activity of the TC is delivery of seminars by certified instructors. Knowledge related to the seminars, the instructors, the trainees and the classrooms will be the domain of the ontology.
For what purpose are we going to use the ontology?
We are going to use the ontology in order to organise, store, retrieve and disseminate the knowledge that exists in the TC concerning the domain that we already defined. Users of this knowledge will be any person – employee of the TC.
Does the ontology contain enough information to answer these competency questions?
One way to define the domain [Uschold, 96] and the level of detail for the properties is through competency questions which will be used to define the search space and specify the requirements for the ontology. The competency questions listed below are indicative:

- Is it possible for the TC to accommodate a conference for 200 persons?
- Is the TC capable of having Computer Based Training or distance learning seminars or ad hoc?

- The director of the TC wants to evaluate the efficiency of the centre if the system informs him / her for the frequency of visits of the trainees, does he/she knows this info?
- Is there a log seminar file to figure out major problems?
- Which training methods does the TC have?
- Is it possible for a trainee to log on to the Internet to download e-mail?
- Can I check which instructors accomplished seminar “Check-in ver.1” last year in order for them to follow up “Check-in ver.2”?
- Which seminars are there in this year’s schedule?
- Which seminars ran last December and who facilitated them?
- How many individuals have attend seminar “Lost Baggage” and when?
- Which seminars has employee “Georgiou” attended in the past?
- Are there any seminars with common topics that can be merged?
- Are there any preconditions to attend seminar “Safety on Board”?

3.2.2 Step 2: Re-using existing ontologies

This step is not applicable in our project since we were not able to locate any related existing data schemata or ontologies.

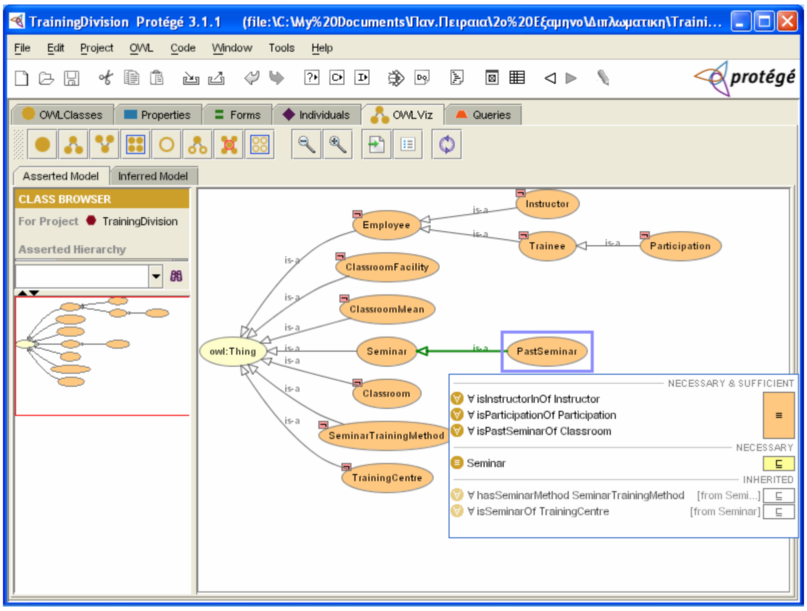
3.2.3 Step 3: Enumerate the terms of the ontology

In this section we declare all the terms that describe important concepts and their characteristics. Important terms of the TC knowledge base are: *instructors, trainees*, personal data, e.g. *first-last name*, *sex*, *birthday*, *marital status*, *address*, *phone no* and firm data, e.g. *ID, date of hire, specialty, station*, classroom data, e.g. *location, capacity, training means, facilities,* in which classrooms took place the seminars, seminars data, e.g. *title, duration, topics, max. number, training methods, start & finish date, log file, instructor & trainees of the seminar, degree*. We also want to store information about the TC itself like *infrastructure* and *facilities* available and range of *seminars* offered.

3.2.4 Step 4: Class definition and Class hierarchy

To develop the class hierarchy we followed the top-down approach starting with the most general classes and subsequent specializations of them.

In figure 1, we present the asserted hierarchy of the initial ontology using the OWL Viz plug-in in the PROTÉGÉ tool. First level classes (concepts) are *Training Centre, Employee, Seminar, Classroom, Classroom means* and *Facilities, Training methods*. Subclasses are *Instructor, Trainee, Past Seminar* and sub-class of sub-class Trainee is *Participation*.


TrainingDivision Protégé 3.1.1 (file:\C:\My%20Documents\Παν.Πειραια\2ο%20Εξαμηνο\Διπλωματικη\Traini...
File Edit Project OWL Code Window Tools Help
protégé
OWLClasses Properties Forms Individuals OWLViz Queries
Asserted Model Inferred Model
CLASS BROWSER
For Project TrainingDivision
Asserted Hierarchy
owl:Thing
Employee
Instructor
Trainee
Participation
ClassroomFacility
ClassroomMean
Seminar
PastSeminar
Classroom
SeminarTrainingMethod
TrainingCentre
is-a

NECESSARY & SUFFICIENT

∀ isInstructorInOf Instructor
∀ isParticipationOf Participation
∀ isPastSeminarOf Classroom

NECESSARY

Seminar

INHERITED

∀ hasSeminarMethod SeminarTrainingMethod [from Semi...]
∀ isSeminarOf TrainingCentre [from Seminar]

Figure 1: OWLViz displaying the Asserted Hierarchy for Training Centre

3.2.5 Step 5: Determine the data type and the object properties of classes

Terms identified in step 3 and not used in step 4 can be data type properties of the classes defined in step 4, e.g. *first-last name*, *sex*, *birthday*, *marital status*, *address*, *phone no*, *employee ID*, *date of hire*, *specialty*, *station*, *classroom location*, *capacity*, *training means*, *facilities*, *seminar title*, *duration*, *topics*, *max. number*, *training methods*, *start & finish date*, *log file*. It should be noted that all subclasses of a class inherit the data type properties of that class.

Object properties are: "*Trainee has participated into Seminar*", which relates class Participation with class Past Seminar, "*Instructor of the Seminar*" relates class Instructor with class Past Seminar, "*Classroom of the Seminar*" relates Classroom with Past Seminar, "*Training Centre performs Seminar*" relates Training Centre with Seminar, "*Training Centre has Classroom*" relates Training Centre with Classroom, "*Training Centre has Instructor*" relates Training Centre with Instructor.

3.2.6 Step 6: Determine the restrictions of the data type and the object properties

Data type properties may have different restrictions describing value type, number of values, allowed values and other. For example, *first-last name*, *address* and *seminar* title are string type values while *date of hire*, *birthday*, *start & finish date* are date values.

The relation "*Trainee has attended Seminar*" has multiple values referring to multiple Seminars that the Trainee has attended; these values are individuals of the class "*Past Seminar*". The relation "*Instructor of the Seminar*" has multiple values referring to multiple Seminars that the Instructor has instructed and the values are individuals of the class “*Accomplished Seminar*”.

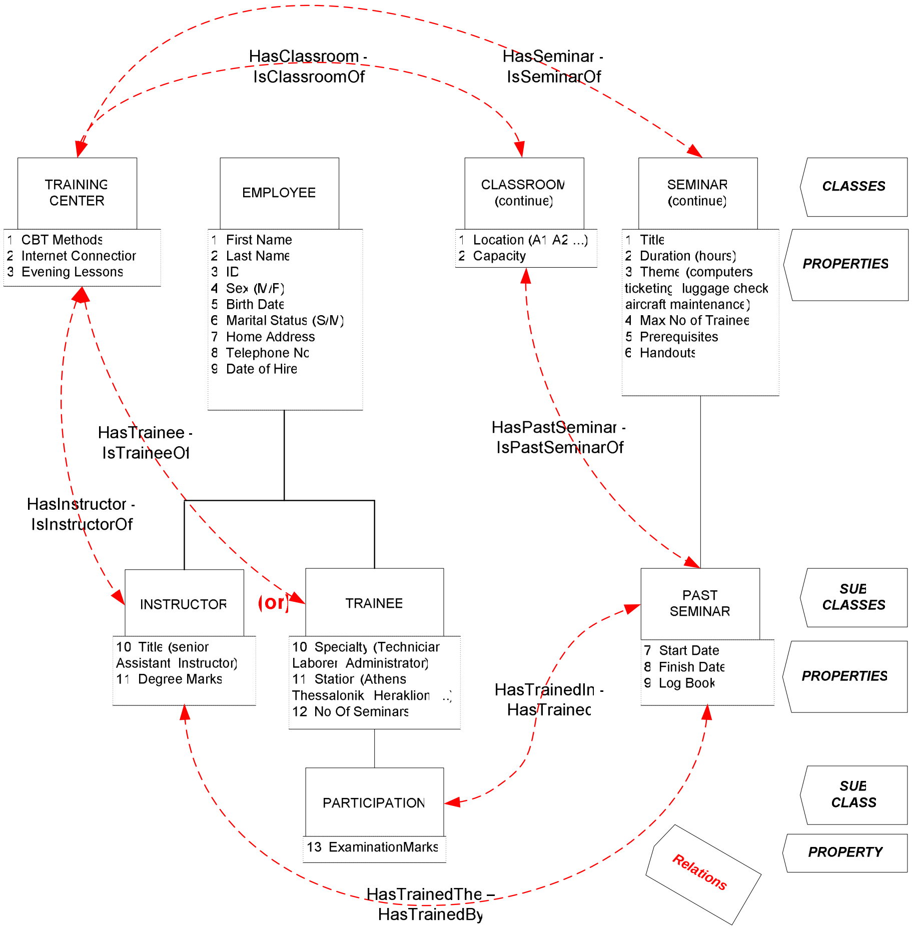


Figure 2: Part of the Training Centre Initial Ontology

3.2.7 Step 7: Creation of individuals

A number of individuals were entered using PROTÉGÉ but full deployment of the project will take place after the next phase when the initial ontology structure will have evolved based on each participant’s opinion.

Figure 2 presents the main part of the Training Centre initial Ontology structure. At the first level we see the general classes, e.g. Training Centre, Employee, Classroom and Seminar, at the second level the subclasses, e.g. Instructor, Trainee, Past Seminar and at the third the sub-sub-classes, e.g. Participation, as we defined them in step 4 above. Each parent class is connected with a straight line with its descendant class(es). Below every class there is a dotted parallelogram specialising and declaring each class' features (data type properties). Finally, the red dashed arrows mark the relations between classes (object properties) with their names on every one of them. The top name concerns the relation between the class on the left-hand side and the class on the right-hand side and the second name below concerns the inverse relation between the right and the left classes.

3.3 Phase 3: Initial structure evolution according to Nominal Group Technique

In the previous phase we created an initial version of the ontology, according to information and requirements that we collected from the case study participants. In this phase we are going to actively involve the participants in the evolution of the initial version of the ontology by asking them to evaluate it and finally reach consensus and agree upon the final version.

For evaluating ontologies, there exist a number of specialized checkers, validators and parsers such as "Validating RDF Parser", "DAML Validator", "DAML+OIL Ontology Checker" and many more [Suárez-Figueroa, 03]. An attempt to integrate such tools in an environment was made by Gómez-Pérez and Guardino with the ODEClean tool which is integrated into the WebODE tool. However, these tools are focused on validating the syntax of the language and not inconsistencies and redundancies in the structure of the ontology that may exist.

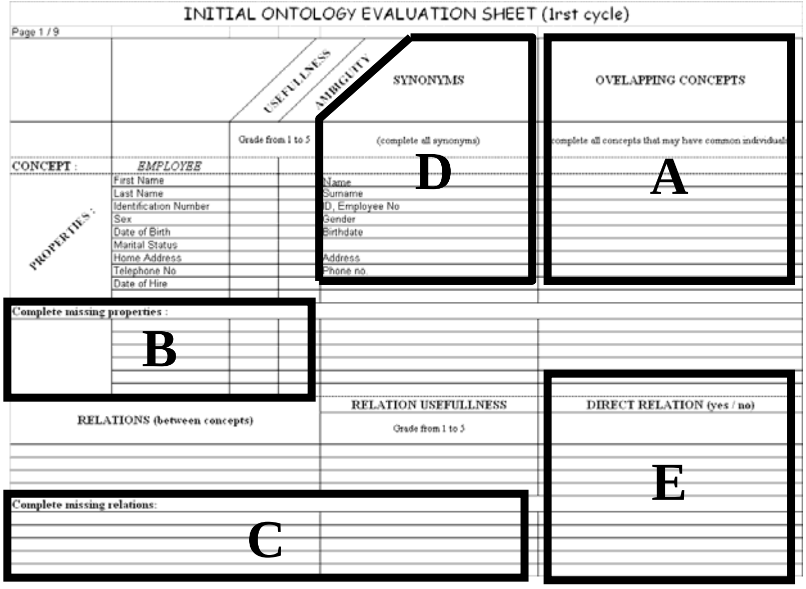
To address this issue in our work, we follow the Porzel and Malaga approach [Porzel, 04] that proposes evaluating the ontology structure on three levels:

(i) the scope of vocabulary-concepts,
(ii) the relation between classes and subclasses (taxonomy), and
(iii) the adequacy of non-taxonomic relations (semantic relations between classes).

We add one more evaluation level, which is checking the implied class overlapping that happens in OWL with the declaration of classes that might have common individuals (see area A of figure 3).

The actions that the designer can take for evaluating the ontology structure according to the aforementioned levels are additions, deletions and substitutions respectively. The following actions that have been depicted in the Evaluation Sheet (ES) part of which is shown in figure 3, should be completed by the participants in every evaluation cycle:

1. Addition of new element (class or property or semantic relation). The participants can enter their proposed new classes. Missing properties about the existing classes are entered in area B of figure 3 with their corresponding marks and semantic relations are entered in area C of figure 3. The SYNONYMS column in area D is added in order to broaden the vocabulary.
2. Deletion of an element (class or property or semantic relation).
3. Substitution a property's name when it's not clear to the participants and there is ambiguity or substitution a semantic relation when it does not hold directly between two classes (see area E of figure 3).



INITIAL ONTOLOGY EVALUATION SHEET (1rst cycle)

Page 1 / 9

		USEFULLNESS	AMBIGUITY	SYNONYMS	OVELAPPING CONCEPTS
		Grade from 1 to 5		(complete all synonyms)	complete all concepts that may have common individuals
CONCEPT :	*EMPLOYEE*				
PROPERTIES :	First Name			Name	
	Last Name			Surname	
	Identification Number			ID, Employee No	
	Sex			Gender	
	Date of Birth			Birthdate	
	Marital Status				
	Home Address			Address	
	Telephone No			Phone no.	
	Date of Hire				

Complete missing properties :

RELATIONS (between concepts)	RELATION USEFULLNESS Grade from 1 to 5	DIRECT RELATION (yes / no)

Complete missing relations:

Figure 3: Extract of the Evaluation Sheet

The first two activities will be triggered by the ranking results in columns "USEFULNESS" and "RELATION USEFULNESS" for existing properties and relations respectively, while the third activity will be triggered by the ranking results in columns "AMBIGUITY" and "DIRECT RELATION".

For reaching consensus among participants, we employ the NGT consensus building technique. The initial version of the ontology feeds the iterative evaluation process until no objections regarding the ontology structure exist. This way, the resultant ontology is the product of a collaborative approach where everybody has participated, committed and finally agreed upon.

In our pilot case, before starting the evaluation and consensus building processes, we briefed the participants on what is going to happen in this phase and what their role in this phase is. We also encouraged participation by everyone and suggested practical methods and procedures (e.g. usage of flipcharts) that can help the team to work better in a short amount of time.

We next describe the way participants must enter their ranking marks in the Evaluation Sheets, which is actually an action formulation aiming to transform the initial ontology into the final collaborative version, according to the NGT technique. Table 2 presents the Likert scale for columns "USEFULNESS" and "RELATION USEFULNESS" of the Evaluation Sheets.

Grade	1	2	3	4	5
Text	Disagree		Neither Agree - Nor Disagree	Agree	
	Entirely	Not Entirely		Not Entirely	Entirely

Table 2: Likert scale for columns "Usefulness" and "Relation Usefulness"

The results of the rankings in the above columns are averaged across participants. If the average shows:

- Disagree (grade 1 or 2): the item does not pass and have to be deleted from the structure.
- Neither Agree – Nor Disagree (grade 3): the item contains ambiguity and needs further explanation. The leader of the voting explains in detail this particular item to the participants and they vote again on a modified scale without grade 3, in order to decide explicitly whether they agree or not about this item.
- Agree (grade 4 or 5): the item remains in the structure because the majority of the participants approve it.

For example, suppose the participants were seven and their grades were 3, 5, 5, 4, 3, 4, and 5. The average of these numbers is 4 that correspond to the "Agree" grade, meaning that the majority of the participants decided that this item is useful and therefore they want to keep it as is.

Each member of the group votes for every item in the ES only once and the results can not be cancelled afterwards.

Table 3 presents the Likert scale for column "AMBIGUITY". The scale is again from 1 to 5 and it concerns the classes and their properties.

Grade	1	2	3	4	5
Text	No	Some	Indifferently	Enough	Too much

Table 3: Likert scale for column "Ambiguity"

We calculate again, as we did before, the average of the grades and the semantics of them are as follows:

- Grades 1 or 2: the item does not contain any ambiguity and does not require any further process.
- Grade 3: the item contains ambiguity and needs further explanation. The leader of the voting explains the item again in detail and the participants vote again on a modified scale without grade 3, in order to classify it into the other two categories.
- Grade 4 or 5: the term does contain serious ambiguity and it requires substitution with a new one that does not imply multiple meanings. The leader explains again and the participants agree in a new one.

For the evaluation of the initial ontology, we differentiate between two cases:

(a) the participants grade the classes, the properties and the relations of the initial structure that the initial ontology has, and

(b) they propose new items in the structure (brainstorming) and grade them in the empty ES.

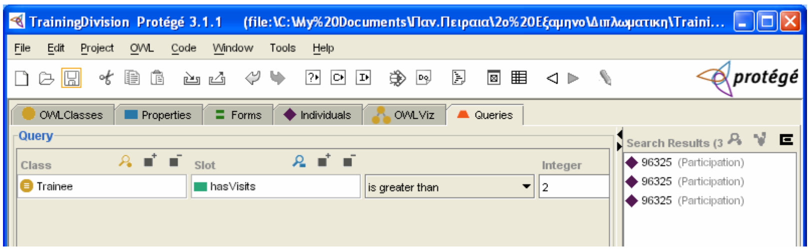
In the first case, the participants vote all together and the results are counted according to the way we just described above (adapted NGT method). In the second case, every new item is presented to the participants and they vote for it, as it happens in the classical NGT method. All other evaluation cycles but the first follows the classical NGT with brainstorming, presentation and voting. The process terminates when there are no more new propositions (additions, deletions, or substitutions). The fact that there are no more propositions implies that there is consensus in the team and acceptance of the resultant collaborative ontology; everybody has been committed to the "product" and the ontology reflects their mutual perspectives.

In our pilot case, the participants filled in the first cycle ES and graded the initial structure as well as their new issues (classes or properties) according to the adapted and the classical NGT method respectively. The whole procedure took place in front of the whole team, real time, as NGT dictates. The results demonstrated minor changes, additions mostly, like insert new properties in the class Employee about number of children, general evaluation index and education level. The relations between classes were all direct with no ambiguities and the structure became more semantically rich with some proposed synonyms. We incorporated these changes into the structure, modify the evaluation sheets accordingly and brought back the new structure for a new cycle of evaluation. The first cycle lasted for about 2.5 hours and it was obvious that the usefulness of the ontology structure became apparent to every participant. The second cycle of evaluation took less than half an hour, did not reveal any more changes and so the process ended into two cycles only.

3.4 Phase 4: Ontology Application

The following answers to indicative, simple competency questions are retrieved using the "Queries" PROTÉGÉ plug-in tool (and not an external reasoner).

QUERY: Let us know if there are trainees that have frequency visit rate more than 2 in order to evaluate the effectiveness of the TC.
Answer [Fig.4]: Employee with ID 96352 has visited – trained in the TC 3 times.


TrainingDivision Protégé 3.1.1 (file:\C:\My%20Documents\Παν.Πειραια\2o%20Εξαμηνο\Διπλωματικη\Traini...
File
Edit
Project
OWL
Code
Window
Tools
Help
protégé
OWLClasses
Properties
Forms
Individuals
OWLViz
Queries

Query

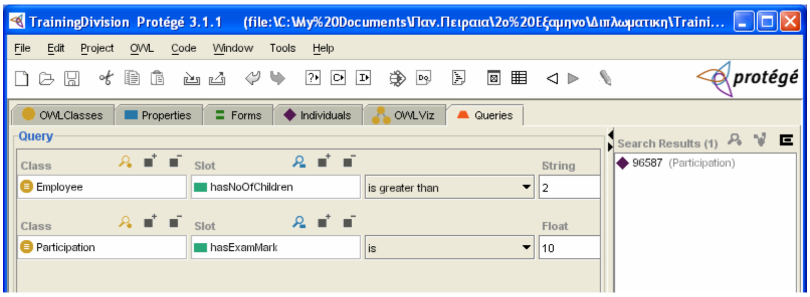
Class	Slot		Integer
Trainee	hasVisits	is greater than	2

Search Results (3

- 96325 (Participation)
- 96325 (Participation)
- 96325 (Participation)

Figure 4: IDs with Frequency visit rate > 2

QUERY: Find out trainees with more than two children and mark examination equal to 10 to honour them?
Answer [Fig. 5]: Employee with ID 96587 deserves fully honours.

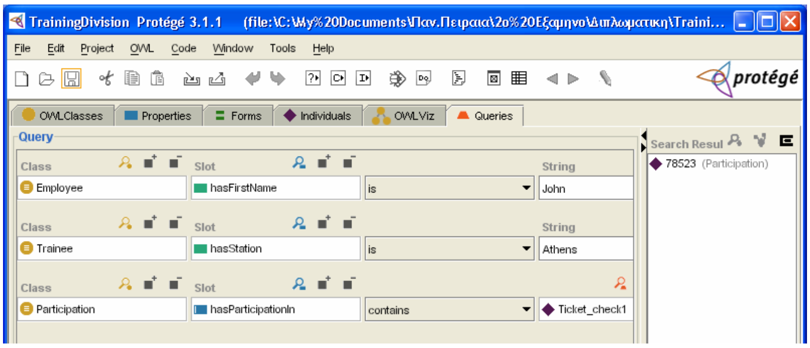

TrainingDivision Protégé 3.1.1 (file:\C:\My%20Documents\Παν.Πειραια\2o%20Εξαμηνο\Διπλωματικη\Traini...
File Edit Project OWL Code Window Tools Help
protégé
OWLClasses Properties Forms Individuals OWLViz Queries
Query
Search Results (1)

Class: Employee — Slot: hasNoOfChildren — is greater than — String: 2
Class: Participation — Slot: hasExamMark — is — Float: 10

96587 (Participation)

Figure 5: Employee 96587 fulfils criteria for praise

QUERY: How many individuals have trained so far in the seminar "Ticket Check 1" that belongs to the station of "Athens" and their First Name is "John"?
Answer [Fig. 6]: Only one employee fulfils the above criteria and his ID is 78523.


TrainingDivision Protégé 3.1.1 (file:\C:\My%20Documents\Παν.Πειραια\2o%20Εξαμηνο\Διπλωματικη\Traini...
File Edit Project OWL Code Window Tools Help
protégé
OWLClasses Properties Forms Individuals OWLViz Queries
Query
Search Resul

Class: Employee — Slot: hasFirstName — is — String: John
Class: Trainee — Slot: hasStation — is — String: Athens
Class: Participation — Slot: hasParticipationIn — contains — Ticket_check1

78523 (Participation)

Figure 6: Employee 78523 lives in Athens, has trained in seminar "Ticket Check 1" and his first name is John

4 Conclusions and Future Work

In this work we presented a methodology for collaborative ontology creation with an embedded mechanism that evaluates the quality and acceptance of the resultant ontology by a group of participants. We also implemented this methodology in the case of the training centre of an airline to demonstrate the application of this methodology into the real world.

The proposed methodology starts with the deployment of an initial version of the ontology, created by the coordinator, based on the participants' requirements. This initial version is being iteratively evaluated by the participants and is finally evolved into the final version. Our approach ensures that all participants agree and accept the resulting ontology, being a product of a joint team effort.

The same methodology can be used on a regular basis (quarterly or biannual) for future evolution of the ontology structure with newer requirements. These requirements may concern new knowledge that is produced from the interaction with the ontology. With this manner, we can embody latest changes in our ontology in order to keep it updated.

Important factors for the quality and quantity of the ontology’s structural content are the degree of participation and the variety of domain knowledge that each member of the team has. The first factor can be leveraged by the facilitator, which plays an important role in the whole process and must encourage active participation from all team members. The latter factor implies that the members of the team must be picked carefully to complement each other and represent diverse domains, experiences, rank positions, ages and viewpoints in general so each one’s contribution results in a maximum totally outcome (rich content).

The collaborative ontology approach to ontology engineering may require more time and effort to deployment as opposed to other approaches (inspirational, inductive, deductive, synthetic) due to the iterative cycles of the consensus building mechanism but the payback is deemed to be better in the long term. This is because the resultant ontology is a joint effort reflecting multiple individuals’ viewpoints instead of a single’s viewpoint that happens in the others approaches. However, the collaborative ontology has by default the acceptance of the participants and thus pulls together everybody to make use of it.

Although the NGT method adopted in our approach is designed so that every participant is freely and equally consulted, the fact that it requires the physical presence of all participants possibly leaves room for revealing some of the negative aspects of face-to-face groupwork such as social pressures of conformity, inappropriate influences (domination of time, topic or opinion), tendency of group members to rely on other to do most of the work, tendency to produce compromised results, etc. We believe that a future groupware system supporting the different steps of the proposed collaborative ontology engineering method can help overcome some of the aforementioned limitations and improve the efficiency and effectiveness of the process by speeding-up the process and improving the quality of results, respectively.

References

[Holsapple, 02] C.W. Holsapple, K.D. Joshi, Collaborative Approach in Ontology Design, Communications of the ACM, Vol. 45 / No 2, 2002

[Kotis, 05] K. Kotis, G.A. Vouros, J.P. Alonso, HCOME: A Tool-Supported Methodology for Engineering Living Ontologies, Lecture Notes in Computer Science, Vol. 3372, 2005, pp. 155-166

[Noy, 01] N. Noy, D.L. McGuiness, Ontology Development 101: A Guide to Creating Your First Ontology, 2001

[Porzel, 04] R. Porzel, R. Malaka, A Task-based Approach for Ontology Evaluation, ECAI-2004 Workshop on Ontology Learning and Population, August 2004, Valencia, Spain

[Suárez-Figueroa, 03] M.C. Suárez-Figueroa, A. Gómez-Pérez, Results of Taxonomic Evaluation of RDF(S) and DAML+OIL ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms import services, In: York Sure, Oscar Corcho (Eds.): EON2003, Evaluation of Ontology-based Tools, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools held at the 2nd International Semantic Web Conference, ISWC 2003, October 2003, Florida, USA, pp. 13-26

[Sure, 02] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke, OntoEdit: Collaborative Ontology Development for the Semantic Web, In: Horrocks I, Hendler JA (Eds.): First International Semantic Web Conference, ISWC 2002, Sardenia, Italy. Remarks of the conference in Computer Science Lecture Notes in Computer Science 2342, Springer-Verlag, Berlin, Germany, pp. 221-235

[Tempich, 05] C. Tempich, H.S. Pinto, Y. Sure, S. Staab, An Argumentation Ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT), A. Gomez-Perez and J. Euzenat (Eds.): ESWC 2005, Lecture Notes in Computer Science 3532, pp. 241–256, 2005

[Uschold, 96] M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, Knowledge Engineering Review, Vol. 11 / No 2, 1996, pp. 93-155

[Youn] S. Youn, A. Arora, P. Chandrasekhar, P. Jayanty, A. Mestry, S. Sethi, Survey about Ontology Development Tools for Ontology-based Knowledge Management, http://www-scf.usc.edu/~csci586/projects/ (access date: 25/11/2005)