Advanced ontology management system for personalised e-Learning

M. Gaeta, F. Orciuoli*, P. Ritrovato

Dipartimento di Ingegneria dell’Informazione e Matematica Applicata (DIIMA), University of Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy

A R T I C L E   I N F O

Article history:
Available online 29 January 2009

Keywords:
Knowledge representation
Knowledge modeling
Knowledge maintenance
Knowledge reuse
Distance learning

A B S T R A C T

The use of ontologies to model the knowledge of specific domains represents a key aspect for the integration of information coming from different sources, for supporting collaboration within virtual communities, for improving information retrieval, and more generally, it is important for reasoning on available knowledge. In the e-Learning field, ontologies can be used to model educational domains and to build, organize and update specific learning resources (i.e. learning objects, learner profiles, learning paths, etc.). One of the main problems of educational domains modeling is the lacking of expertise in the knowledge engineering field by the e-Learning actors. This paper presents an integrated approach to manage the life-cycle of ontologies, used to define personalised e-Learning experiences supporting blended learning activities, without any specific expertise in knowledge engineering.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Knowledge modeling represents a significant activity, that is particularly difficult to perform due to its complexity. Nowadays, in computer and information science, knowledge representation, reuse and sharing are facilitated by the explicit use of ontologies. In [1], an ontology is an “explicit specification of a conceptualization”. The term is borrowed from philosophy, where an ontology is a systematic account of existence. For artificial intelligence systems, what exists is what can be represented. Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among agents. Agents can be both software agents and/or human agents.

Recently, we have seen an explosion of interest in ontologies as artifacts to represent human knowledge and as critical components in knowledge management, Semantic Web, business-to-business applications, and several other application’s areas. Also in the e-Learning area there is a newly great interest in the exploitation of knowledge technologies.

Most of the current learning technology specifications are based on educational metadata: IEEE LOM [2] and ADL SCORM [3], for example, are the standards proposed for describing (and re-using) chunks of learning content annotated through metadata. Metadata is supposed to enable the reuse of these chunks by detailing the conditions of their initial deployment. However, the authors of [5,4] pointed out that such an approach failed to elicit cognitive behaviours and therefore the actual reuse. We believe that the use of ontologies in e-Learning can overcome these drawbacks.

In [6], some benefits from applying ontologies to e-Learning are well explained. The authors asserts that an ontology, formally and declaratively, represents the terminology of a specific domain, defining its essential knowledge. Ontologies are used to support semantic search, making possible to query multiple repositories and discover associations, between learning objects, that are not directly understandable. This is impossible or very complex with simple keyword- or metadata-based search supported by the current standards.

In this paper, we describe methodologies and techniques for supporting a community of experts in modeling educational domains (e.g. mathematics domain, English literature domain, etc.) through the management of convenient educational ontologies namely e-Learning ontologies and exploiting them in order to define and execute personalised e-Learning experiences within blended learning activities.

Blended learning is considered a learning approach defined by the effective combination of different modes of delivery and models of teaching and styles of learning [7]. Personalised e-Learning experiences represent a convenient way to complement face-to-face sessions within a whole blended learning experience. A personalised e-Learning experience could be very important when used for assessing the knowledge acquired by each individual learner during a face-to-face learning session and offering, in case of negative results, personalised remedial works able to fill the identified knowledge gaps with learning paths that best fits the needs, the cognitive state and the learning preferences of each individual learner. In the event that the personalised e-Learning experience can be built, packaged and deployed with an automatic process,
then the whole blended learning activity can become more effective and efficient.

Anyway, knowledge modeling through ontologies is a subjective process essentially, whereby different people that model the same domain, produce, in most cases, different ontologies depending on their sensitivity, their background, etc. In a distributed environment, such as communities of experts, harmonizing the work of all parties can be a relevant activity, in order to enrich and improve the available knowledge bases. For these reasons, we define a set of convenient techniques for versioning and harmonization of e-Learning ontologies. The current methodologies, developed in the e-Learning domain do not allow the integrated management of knowledge that meets all the requirements above mentioned. The proposed approach is conceived to allow the collaborative and shared management of the available knowledge, without having any specific proficiency in knowledge engineering taking into account aspects like ontology harmonization and ontology versioning. In particular, we focus on the collaboration approaches for ontology building and maintenance. More precisely, the most relevant contributions of our work are:

- The definition of convenient models to represent and exploit e-Learning ontologies in order to build and deliver personalized e-Learning experiences taking into account different cognitive states and learning preferences of learners.
- The definition of a set of tools for representing and managing e-Learning ontologies, within a community of teachers, tutors, mentors, etc. (without any expertise of knowledge engineering) through the features of an integrated framework.

The paper is organized as follows: Section 2 presents some related works; in Section 3 we describe our approach to build personalized e-Learning experiences through the use of ontologies; in Section 4 we describe the algorithms and the techniques to manage the life-cycle of e-Learning ontologies through our advanced ontology management system; in Section 5 a case study of an AOMS collaborative ontology construction session is presented. Finally, Section 6 concludes the work.

2. Background and related works

The importance of the knowledge modeling, to effectively organize the available e-Learning resources according to the particular needs of both teachers and students, has been advocated by many authors. Domain knowledge modeling is the basis for constructing domain concept structures and managing related course materials. Brase and Nejdl [8] have showed the increasing importance, in the e-Learning field, of knowledge modeling through metadata definition standards such as the learning objects metadata (LOM). However, these standards introduce the problem of incompatibility between disparate and heterogeneous metadata descriptions across domains, which might be avoided by using ontology as a conceptual backbone in an e-Learning scenario [9,6].

Various researchers (see for example [12]) have observed that e-Learning, even when properly designed and meta-tagged, will not realize full re-usability without the full benefits from the Semantic Web [10]. A number of systems have been developed to manage learning resources through the use of Semantic Web technologies.

A first example of these systems is the Edutella project [11]. It is an open source project based on RDF metadata, for P2P network users interested in the exchange of learning resources.

KGTutor, a knowledge grid based intelligent tutoring system [15], proposes a model for the construction of intelligent tutoring experiences in a more pleasant and effective way. The KGTutor is designed to provide better support to student centered distributed learning. Students’ characteristics, such as previous knowledge and learning styles, are used to choose, organize, and deliver the learning materials to individual students. During the learning progress, the system can also provide objective evaluations and customized suggestions for each student according to their learning performance.

Another representative system that uses Semantic Web techniques in an e-Learning environment is the Courseware WatchDog [14]. It is a module of the larger project PADLR (personalised access to distributed learning repositories) [13] based on a peer-to-peer approach to support personalised access to e-Learning resources. The purpose of this framework was the creation of a software architecture helping teaching staff and students in the search and management of learning objects. WatchDog is completely ontology-based and uses clustering techniques to create personalised ‘views’ of the learning objects. Moreover, it has some techniques for the management of the evolution of ontologies related to the educational content.

These works show that the most relevant difficulty in the knowledge modeling for e-Learning is related to the creation and maintenance of Semantic Web structures (such as ontologies) which can be exploited not only to organize learning objects and to state their inter-relationship but also to build personalised learning paths and to maintain up to date students cognitive states. In particular, in the e-Learning area, there is still a lack of methodologies and techniques that allow the effective management of ontologies in particular addressing the most actual ontology research issues such as Ontology Versioning, Ontology Harmonization and Collaborative Ontology Construction.

The issue of ontology versioning is related to the nature of ontologies that are not static piece of knowledge, but evolve over time. Changes in domain's concepts, adaptations to different tasks, or changes in the conceptualization require modifications of the ontology. Ontology Versioning is the management of the changes which can occur to the ontologies in their life-cycle, more precisely, it is the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology. The evolution of ontologies can cause interoperability problems, therefore it is important to track changes that may occur to the ontologies because of: (i) changes in the domain, (ii) changes in the shared conceptualizations or (iii) changes in the specifications [16,17]. Ontology Harmonization, instead, is the ability to harmonize two or more ontologies in a unique ontology in order to enrich the available knowledge base. It is strictly related to two main issues such as ontology matching [18] for the recognition of correspondences between ontologies and ontology merging [19] for the actual fusion of those ontologies. Ontology Versioning and Harmonization are still open problems in the Semantic Web area, due to the complexity of ontology management. Our approach does not want to solve all these problems, but it is focused on the definition of a set of methodology that can be used to effectively manage ontologies in the e-Learning domain. In [20], the authors present a consistent methodology for the Collaborative Ontology Construction articulated in four different phases: preparation, anchoring, iterative improvement and application. Furthermore, in [29] is illustrated a report about detailed interviews with members of ten different ontology engineering projects. Interviews were conducted either on the phone or in person. Analyzing the report, it emerges the lack and the request for user-friendly tools able to support collaborative ontology construction orchestrating single-user asynchronous tasks.

Usually, two main collaboration modes [22] are adopted by ontology editing tools to support collaborative ontology development. The first one is the model where everyone accesses the same version of an ontology and changes are immediately visible to
everyone (synchronous mode). In the second one, users can have their personal sandbox space and integrate their changes with the master version later (asynchronous mode). Several variants and mixed approaches are also used by existing ontology management tools.

The well known Protégé [21] tool adopts a client-server architecture allowing users to concurrently apply changes on a single ontology. Changes made by a user are immediately visible to the other users. APECKS [24] is mainly thought in order to support the collaborative analysis and definition of domain ontologies, meanwhile Tadzebao [25] supports the collaborative development enabling users interaction through texts, images and handwriting designs exchange. Some tools, like OntoRama [23], use a collaborative P2P network instead of a client-server system. Recently a new approach, based on Wiki engines [27], comes to age. Once few editing rules are given to authors, it is possible to obtain Wikis that are easily transformable into ontologies and vice-versa. Ontowiki [35] and Semantic MediaWiki [28] are some collaborative wiki-based ontology development tools.

Our idea consists in defining a technique for collaborative ontology construction based on principles of the methodology presented in [20], that takes care of the lacs and needs emphasized in the interviews of [29], that aims at overcoming the knowledge engineering expertise needed by the existent tools and that exploits the typical human–computer interactions provided by Web 2.0 applications [32].

3. Personalised e-Learning based on ontologies

In the following subsections, we focus on how to model educational domains in a machine-understandable way using ontologies and consequently, how these structures can be exploited in order to personalize e-Learning experiences using information coming from learner profiles and content coming from learning object repositories (in which learning object are annotated with semantic information through standard metadata schema). For the sake of simplicity we will consider that e-Learning experiences are represented by sequences of learning objects, but the approach is also suitable when e-Learning experiences are made up of complex flows of learning activities. Therefore, the main principles of our approach are: (i) modeling of educational domains by means of e-Learning ontologies; (ii) modeling of learner cognitive state and preferences (Student Model); (iii) annotation of Learning Objects with metadata and semantics connections between learning objects and ontologies elements (learning object model) and (iv) modeling of e-Learning experiences (e-Learning experience model).

3.1. Modeling of educational domains

An e-Learning ontology can be modeled with a graph in which nodes are relevant concepts (arguments, topics, etc.) within the educational domain of interest and edges are binary relations between two concepts. Our approach mainly foresees three kinds of relations: HasPart (in brief HP) that is an inclusion relation, IsRequiredBy (in brief IRB) that is an order relation and SuggestedOrder (in brief SO) that is a “weak” order relation. The restricted set of relations does not imply limits to the knowledge representation but it is a convenient method to improve the computational complexity of algorithms that have to navigate and process the graph. Let’s illustrate how to build an e-Learning ontology. Suppose we have to model the educational domain D, so we try to conceptualize the knowledge underlying D and find a set of terms representing relevant concepts in D. The result of the previous step is the list of terms \( T = C, C_1, C_2, C_3 \), where \( T \) is one of the plausible conceptualizations of D. In order to explain the semantics of HasPart relation we can refer to the ontology illustrated in Fig. 1 where the three HasPart relations HasPart\( (C, C_1) \), HasPart\( (C, C_2) \) and HasPart\( (C, C_3) \) means, in terms of e-Learning, that in order to learn concept C learners have to learn concepts \( C_1, C_2 \) and \( C_3 \) without considering a specific order.

In Fig. 1a, we note the existence of elements that are not concepts or relations. The new elements to introduce are the learning objects \( LO_1, LO_2 \) and \( LO_3 \). You can interpret the connection between a concept and a learning object, for instance \( C_1 \) and \( LO_1 \), as a HasResource (in brief HR) relation. The relation HasResource\( (C_1, LO_1) \) means that the educational content packaged in learning object \( LO_1 \) explains concept \( C_1 \). So, if we assume that our learning objective is \( C_1 \) then the correspondent assembled e-Learning experience is composed only of \( [LO_1] \), otherwise if the learning objective is \( C \) then the assembled e-Learning experience will be composed of one of the plausible permutation of \( [LO_1, LO_2, LO_3] \).

Now, consider the ontology shown in Fig. 1b. This ontology presents two IsRequiredBy relations, IsRequiredBy\( (C_1, C_2) \) and IsRequiredBy\( (C_2, C_3) \). The two relations mean that \( C_1 \) has to be necessarily learned before \( C_2 \) and \( C_2 \) has to be necessarily learned before \( C_3 \). In this case if \( C \) is the learning objective, learners have to learn the ordered sequence of concepts \( \{C_1, C_2, C_3\} \) and correspondingly they can join the e-Learning experience assembled by the ordered sequence of Learning Objects \( [LO_1, LO_2, LO_3] \). Alternative permutations like \( \{C_2, C_3, C_1\} \) will be invalid. The sequence of concepts useful to reach a pointed learning objective is called Learning Path, meanwhile the operation used to construct the concrete e-Learning experience assembling learning objects sequence is called Resource Binding.

Above, we have outlined the foundations of our modeling approach, now we want to refine the approach description. First of all, we state that the same learning object can explain more than one concept within the same ontology. In general, the relation HasResource can be represented by the function HasResource\( (C_1, C_2, \ldots, C_n, LO_1) \) meaning that \( LO_1 \) explains all concepts \( C_1, C_2, \ldots, C_n \). Otherwise, it is possible to have more than one learning object explaining the same concept. In general, we can have at the same time the relations HasResource\( (C_1, LO_1) \), HasResource\( (C_1, LO_2) \), HasResource\( (C_1, LO_3) \), etc.

Lastly, suppose you have a SuggestedOrder relation between concept \( C_1 \) and concept \( C_2 \) that is SuggestedOrder\( (C_1, C_2) \), this relation states that the modeler thinks that is preferable to explain concept \( C_1 \) before concept \( C_2 \), but this is not mandatory.

3.2. e-Learning ontologies representation

In order to foster knowledge sharing and systems interoperability, we need to represent e-Learning ontologies in a widely approved formal language, that is OWL [42]. First of all, you have to define only one base class called Concept and three binary rela-
tions (as Object Properties), from class Concept to class Concept, called HasPart, IsRequiredBy and SuggestedOrder. Now, the base schema is ready and you have to populate the ontology with instances (individuals) of class Concept and establish the relations between them.

Though you can use a knowledge representation tool like Protégé [21] in order to be supported in writing OWL ontologies we have defined an ontology editing tool (See Section 4.1) that does not require specific knowledge engineering skills. In the next section, we will show how e-Learning ontologies can be exploited, together with other information, in order to build personalised e-Learning experiences through a semi-automatic process.

3.3. Personalised e-Learning experiences

Before describing the steps performed by the process for building personalised e-Learning experiences, we must introduce the three models which the process relies on.

The Student Model is composed by a cognitive state and a preferences state. Cognitive state represents the knowledge reached by a learner at a given time and is composed of a list of concepts each with an associated score (values between 0 and 1). If the score for a given concept is greater than a fixed threshold then that concept is considered as known by the learner. Preferences state, for a given student, is a set of couples (property, value) used to represent in machine-understandable way the learning preferences of the student. For the sake of simplicity, we consider that properties are contained in the set (Learning Resource Type, Interactivity Level, Typical Learning Time, Difficulty, Language, Context). We assume also that there is one preferences state and one cognitive state for each learner.

The learning object model foresees that a metadata is associated to each learning object existing in the learning object repository. We assume that the metadata includes: (a) a list of explained concepts, and (b) a list of couples (property, value) where property is an element of the same set defined for the Student Model and value is an admissible value for that property. It is important to underline that the explained concepts list represents the semantic link (see HasResource relation in the previous subsection) between a Learning Object and an e-Learning ontology.

In the e-Learning experience model, e-Learning experiences are defined as: (a) a set of Target Concepts (TC) i.e. the set of high-level concepts to be transmitted to the learner (namely Learner Objective in the previous subsection), (b) a Learning Path (LP) i.e. an ordered sequence of atomic concepts that is necessary to transfer to a particular learner in order to let him learn TC (given the personalization on a particular learner, the sequence does not contain concepts already known by that learner) and (c) a Presentation (PR) i.e. an ordered list of learning objects that the learner has to use in order to acquire concepts included in the LP. LP can be automatically obtained starting from TC, while PR can be automatically constructed starting from LP and querying (using metadata information) the Learning Object repository. TC can be settled by the teacher or directly by the learner (in case of self-directed learning) and can be obtained by manually selecting concepts on ontologies or by selecting pre-defined groups of concepts.

Excluding the selection of TC and other customization parameters, the building process is fully automatic and realized through the execution of several algorithms. The most important are: Learning Path Generation Algorithm and Presentation Generation Algorithm. The first one determines the ordered sequence of atomic concepts needed to reach a satisfactory knowledge about selected TC on the basis of a reference e-Learning ontology, a set of TC and a given cognitive state. The second one selects and orders a set of learning objects, explaining all the concepts in the Learning Path, that best fits with the student model preferences state. The algorithm acts minimizing the number of learning objects (a single learning object could explain more than one concept) useful to cover the whole Learning Path.

The building process is repeated for each student enrolled to the e-Learning experience. Once the e-Learning experience is ready and packaged, then the enrolled student can start his/her execution phase. The student accesses the Presentation and executes the proposed activities studying the sequence of learning objects provided by the system until a milestone (assessment point within the Learning Path) is reached. At this point the student executes the assessment phase and sends the results back to the system. An algorithm, namely Student Model Update Algorithm, is invoked in order to update the student’s cognitive state on the basis of latest test results. The sequence of previously described algorithms is now re-executed considering the evolution of the student’s cognitive state. In this way a personalised remedial work is automatically provided. The on-line learning process ends, for a student, when his/her cognitive state includes all concepts of the Learning Path.

4. Advanced ontology management system

The educational domain modeling approach and the exploitation of e-Learning ontologies in the personalised e-Learning experience construction process have been implemented by Intelligent Web Teacher (IWT) platform [30,31]. In this section we describe how the Advanced Ontology Management System (AOMS) has been built to support the IWT platform in order to improve the management capabilities of its knowledge base (e-Learning ontologies). Once an ontology has been constructed and validated, through the AOMS tools, it is annotated with metadata, indexed and archived. IWT teachers can look up AOMS Repository for the desired ontology and import it in IWT. The ontology exchange format is OWL that also fosters the interoperability of AOMS with third party knowledge representation tools like, for instance, Protégé.

More precisely, we have placed the focus on a set of tools supporting ontology editing and collaborative ontology construction that can be used by domain experts (teachers or, in general, persons with special knowledge or skills in a specific matter like Physics, Italian art, Economics, etc.) with few (or any) competencies in knowledge engineering. Therefore, domain experts have not to interact with complex knowledge engineering environments but with user-friendly visual interfaces, such as the visual ontology editor (VOE). The most important AOMS tool is the coordination and cooperation tool (CCT) useful for the collaborative ontology construction. With a collaborative approach, the ontology construction will become an effort reflecting experiences, competencies and viewpoints of several persons (participants). The tangible results of the collaboration is to obtain a richer and more reliable represented knowledge thanks to the collective intelligence.

In the end, the AOMS includes other two main components: Ontology Merging Tool and the Semantic Wiki Engine, respectively used in order to aid the users to harmonize two or more ontologies modeling the same educational domain and to implements the consensus about a collaboratively constructed ontology.

4.1. Visual ontology editor

Most of the actors moving around e-Learning do not want to be engaged with formalisms that, though expressively powerful, are far from their habits and their common activities. Therefore, we construct an abstraction layer atop of OWL. The abstraction layer maps the e-Learning ontology model, defined in Section 3.1, on the OWL constructs and offers the user a simple visual language,
that can be used to graphically design e-Learning ontologies. The AOMS Visual Ontology Editor (VOE) provides a graphical web-accessible environment (Fig. 2) enabling the use of the visual language. Domain experts can define a new concept only by drag ovals into the design panel and create relations between concepts by drawing lines between two ovals. The tool enables users to add metadata for the whole ontology that can be stored in AOMS Repository and subsequently searched and retrieved by the metadata-based search engine. Within the context of AOMS, ontologies are stored using an internal representation format, but when they have to be exported towards IWT, they are translated in OWL. The idea is to exploit advantages of OWL (e.g. interoperability, etc.) hiding its complexity to the ontology designers that can reason only in terms of graphical representations. Third party editors and reasoners can easily work after the conversion in OWL format.

An extension has been supplied to the VOE: The ontology evolution tracker sub-system. Ontologies are not static and may evolve over time. In our approach, the changes applied to an ontology have impact mainly on learning objects linked to the ontologies through metadata, and on students' cognitive states that store all concepts known by each learner. Therefore, when an ontology evolves, and a new version of that ontology is adopted, several inconsistencies may appear into the system. The ontology evolution tracker sub-system listens to all update operations executed on the ontology by means of the VOE, and for each operation, a set of structured information are built, indexed and stored. These information help us to reconstruct the “history” of e-Learning ontologies and to resolve inconsistencies with the learning resources (i.e. learning profiles and learning objects) depending on them. For instance, thanks to the Ontology Evolution Tracker Sub-system, it is possible to re-align (in a semi-automatic way) Learner Profiles after the adoption of a new version of the reference ontology.

4.2. Coordination and cooperation tool

The AOMS coordination and cooperation tool (CCT) is realized as an extension of an open source platform named GENESIS [33], coming from the homonymous project funded by the European Commission. The coordination aspects are supported by a Workflow Engine and an Artifacts Management Module. GENESIS has been specialized for software development processes but is realized as a generic platform for the definition and the execution of coordination and cooperation processes that can be extended in order to support collaborative ontology construction processes.

The collaboration processes executed through CCT functionalities can be classified in Sequential and Parallel. Sequential processes contemplate that work is completed and then handed off from a participant to another participant. Thus, each participant is guided by what has been done before. A first release of the work result is provided by the coordinator that states also the guideline for the other participants. This approach has more benefits if the process starts from a complete draft of the result to be achieved and it needs to obtain an enriched artifact exploiting the collective intelligence expresses by a group of persons. Parallel processes foresee the presence of a coordinator in order to stimulate the participants and put together all the stuffs coming from each member of the collaboration group. Each participant yields separately its own artifact meeting the guidelines stated by the coordinator. The coordinator harmonizes all the received artifacts into a commonly approved schema. This approach is particularly advantageous when the coordinator divides the whole work into parts.

![Fig. 2. AOMS visual ontology editor.](image-url)
with few points of overlapping, assigns each part to a single participant and, lastly, collects and assembles all the partial results in a complete result. Workflow schemas representing the sequential mode collaborative ontology construction process and the parallel mode collaborative ontology construction process are depicted in Fig. 3 where the four new human-activity types have this meaning:

- **Start-up Activity**: this activity is needed in order to edit a first draft of the ontology used as an initial version to guide the collaboration process.
- **Refinement Activity**: this activity is used to modify a previous version of an ontology in order to obtain a more refined version to be stored into Artifacts Management Module. This activity defines a new version of the ontology.
- **Harmonization Activity**: this activity is executed when there is the necessity to harmonize two or more ontologies representing frames of the same domain. The harmonization task is performed by a semi-automatic Ontology Merging Tool that supports the user in the merging process. The result of this process can be refined by users using the Visual Ontology Editor.
- **Validation Activity**: this activity is executed in order to check the consensus of all participants to the collaboration process and to realize the last concurrent refinements. A specific version of an ontology is retrieved from the document repository and transformed into a Wiki space where ontology concepts are mapped as Wiki pages and relations are mapped as links between pages updatable by all participants. A special user (coordinator) moderates the collaboration activity (he/she could perform a rollback to cancel any modification to the Wiki space) and decides when the activity is finished.

The behavior of the CCT handling all the activity types is described using collaboration diagrams in Fig. 4. The CCT can be straightforwardly used in order to apply the methodology defined in [20] through the implementation of its four phases:

- **Preparation**: a domain expert (the coordinator) assembles a community of other domain experts. The community has the aim of building an e-Learning ontology, modeling a specific educational domain. The coordinator defines collaboration objectives (e.g., building a Physics Ontology, etc.), constraints (e.g. deadline, etc.), guidelines (e.g. modeling rules, tools tutorial, etc.) and notifies them to the community participants that could agree or not. The coordinator, exploiting the process definition features of CCT, selects the workflow schema (sequential or parallel) for collaboration tasks, customizes it in order to prepare a Refinement Activity for each participant. The workflow schema will be executed using the process execution features of CCT.
- **Anchoring**: the coordinator (both in sequential and parallel mode) provides an initial ontology version executing a Start-up Activity. The coordinator could exploit the reuse of an existing ontology (coming from the Artifacts Management Module, AOMS Repository or from external knowledge representation tools or archives) or edit the seed ontology from scratch.
- **Iterative improvement**: after the end of the Start-up Activity, the control passes to CCT that follows the selected workflow schema in order to start the first Refinement Activity (for sequential schema) or the set of parallel Refinement Activities (for parallel schema). CCT continues to execute the process guided by the workflow schema and events coming from users. In the case of sequential mode, the latest activity (before the Validation Activity) is represented by a Refinement Activity that provides an ontology version, obtained by the execution of all refinement steps, ready to be validated by all participants. The latest Refinement Activity of the sequential workflow schema can be assigned, by coordinator, to a participant or to himself/herself during the Preparation phase. In the case of parallel mode, the latest activity (before the Validation Activity) is represented by an Harmonization Activity in which the coordinator can merge all the results coming from previous parallel Refinement Activities. The Validation Activity is used to achieve the consensus of all participants about the e-Learning ontology obtained as artifact of the collaboration process. During the Validation Activity the participants can concurrently apply changes to the ontology and discuss about it. All changes can be rolled back from the coordinator that decides also when the activity ends. The ontology resulted by the Validation Activity is the official artifact of the whole collaborative ontology construction process.
- **Application**: the collaboratively produced ontology is, now, stored in the AOMS Repository and can be imported and exploited in order to produce a personalised e-Learning experience.

4.3. Ontology merging tool

The Ontology Merging Tool is based on a merging algorithm consisting of four steps. The first step simply consists of the selection
of two source ontologies. The second step is represented by the matching phase in which couples of concepts, belonging to the two ontologies, with a high similarity level (both lexical and semantical) are identified. A match is identified when two terms represent the same concept in two different ontologies. As mentioned above, the computation of similarity will be transparent to the user. The matching algorithm we proposed is based on the integration of different techniques that are suitable to be applied for e-Learning ontologies and without any pre-defined document corpus. In particular, we use:

- **String-based technique**: it is a similarity computation applied to all possible pairs of concepts \((C_1, C_2)\), \(C_1\) belonging to an ontology source \(O_1\) and \(C_2\) belonging to an ontology source \(O_2\). Similarity Metrics like q-Gram and Levenshtein Distance [34] and typical operations of Text Processing like Stemming are applied in order to establish a first set of matches between the two ontologies.
- **Graph analysis**: graph analysis techniques [37] are applied to the two ontologies, taking into account also the first set of matches, in order to enrich the set using information coming from the structures of the two ontologies.
- **Semantic analysis**: Normalized Google Distance [40] and Wikipedia [41] are used as linguistic resources in order to refine the set of matches using semantic similarity information.

In the third step, starting from the set of matches, the algorithm generates a list of suggestions, containing the operations that can be applied to perform the ontology merging. In the fourth (and final) step, suggested operations are executed and, moreover, the algorithm determines and resolves redundancies (for relations and concepts) that could be generated. The Ontology Merging Tool is fully interactive because we believe that a completely automatic process is very difficult to obtain and it would not be very effective. So, each step of the merging algorithm can be validated by the users that, eventually, can manipulate the step results following the ideas proposed with PROMPT [37].

### 4.4. Semantic wiki engine

In order to reach a common consensus about the validity of an e-Learning ontology we need to have an open collaboratively environment where persons, can effectively discuss, cooperate and validate an ontology using tools that follow the approach proposed in [36]. The AOMS Semantic Wiki Engine (SWE) is implemented by extending an open source Wiki Engine namely ScrewTurn [38]. SWE manages semantic wiki spaces (SWS). A SWS represents an e-Learning ontology, a page in SWS represents a concept of the educational domain of interest and, lastly, a link between two pages represents a relation between two concepts. The link mechanism in SWE is quite different from that of other semantic wiki engines [26]. In fact, our approach foresees the presence of typed links between pages in order to map correctly the different kinds of relations (e.g. HasPart, IsRequiredBy and SuggestedOrder) coming from e-Learning ontologies model. In this way, a SWS is exactly a new representation of an e-Learning ontology that is an alternative...
to the graphical representation provided by the VOE. The main advantage of the SWE with respect to VOE is that SWE is multi-user. The multi-user feature enables more than one domain expert to update concurrently the same version of an ontology. An effective collaborative work is ensured by a tracking feature that stores all information about updates to the SWS in order to enable the coordinator to perform roll-back operations. In the context of AOMS, the SWE is used to implement the Validation Activity handled by the CCT. The e-Learning ontology, coming from the previous task in the collaboration process, is converted into a SWE where the consensus can be reached by all participants driven by the coordinator. When the Validation Activity comes to end, the updated SWE is converted into an e-Learning ontology (AOMS XML-based internal representation), annotated with metadata, indexed and stored and available for IWT users.

5. Case study: C-programming ontology

We have tested the aforementioned methodologies and tools in The Faculty of Engineering at University of Salerno. Professor M. Gaeta taught the course Programazione_C (Computer Programnings Foundations) for the class 2006/2007 of Electronics Engineering. The course programme includes a module about C-programming, so professor M. Gaeta, asked his three laboratory assistants to participate to the collaborative construction of the C-programming ontology. Professor M. Gaeta and his assistants used AOMS to collaborate to the ontology building in order to define and publish the personalised e-Learning experience on Intelligent Web Teacher, that is still the e-Learning platform adopted by the Faculty of Engineering at University of Salerno. In order to simplify the description and the graphical representation of ontologies fragments within this work we will show only top levels of ontology. Professor M. Gaeta chooses to set up a collaborative environment able to support a parallel process. Then, he draws a seed ontology using the VOE within the Start-up Activity. The seed ontology has only four concepts representing a first coarse-grained conceptualization of the C-Programming domain: Programmazione_C (C_Programming), Principi_basilari (Basic_principles), Costrutti (Constructs) and Tipi_dati_strutturati (Structured_data_types). More in details, the seed ontology presents the following relations:

- HasPart(Programmazione_C, Principi_basilari)
- HasPart(Programmazione_C, Costrutti)
- HasPart(Programmazione_C, Tipi_dati_strutturati)

Once the seed ontology is ready, the Professor signs the Start-up Activity as finished. In this way, the seed ontology is stored into the Artifacts Management Module and the CCT can activate the three required Refinement Activities that will be managed by the laboratory assistants of the Professor. In particular, the first assistant has to refine the Principi_basilari concept, the second assistant has to explore the Costrutti concept and the third assistant has to detail Tipi_dati_strutturati concept. Once received his task specifications, the first assistant starts his Refinement Activity and accesses to the VOE that automatically loads the seed ontology. He designs and stores the ontology as version 1.1 (Fig. 5) that is obtained by refining the seed ontology and in particular by exploding the Principi_basilari concept.

Once the ontology has been stored, the first assistant closes the own Refinement Activity. Concurrently, second assistant engages his Refinement Activity, with the objective of exploding Costrutti concept, and produces the version 1.2 starting from professor Gaeta's seed ontology. Even the third assistant has carried out his task. He refines the Tipi_dati_strutturati concept on the seed ontology and builds the version 1.3. Once all parallel Refinement Activities have been closed, the Harmonization Activity can start. Professor Gaeta starts the Harmonization Activity and supported by the AOMS coordination and cooperation tool, he harmonizes versions 1.1, 1.2 and 1.3 executing two different merging sessions. In the first session, he uses the Ontology Merging Tool to merge the first assistant's ontology with the second assistant's ontology. During a first matching phase, the tool identifies the following matches:

<table>
<thead>
<tr>
<th>Version 1.1</th>
<th>Version 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principi_basilari (Basic_principles)</td>
<td>Principi_basilari (Basic_principles)</td>
</tr>
<tr>
<td>Funzioni (Functions)</td>
<td>Funzioni (Functions)</td>
</tr>
<tr>
<td>Operatori_aritmetici (Arithmetics_operators)</td>
<td>Operatori_aritmetici (Arithmetics_operators)</td>
</tr>
<tr>
<td>Costrutti (Constructs)</td>
<td>Costrutti (Constructs)</td>
</tr>
<tr>
<td>Tipi_dati_strutturati (Structured_data_types)</td>
<td>Tipi_dati_strutturati (Structured_data_types)</td>
</tr>
<tr>
<td>Dichiarazione (Declaration)</td>
<td>Dichiarazione (Declaration)</td>
</tr>
<tr>
<td>Invocazione (Invocation)</td>
<td>Invocazione (Invocation)</td>
</tr>
<tr>
<td>Parametri (Parameters)</td>
<td>Parametri (Parameters)</td>
</tr>
<tr>
<td>Programmazione_C (C_Programming)</td>
<td>Programmazione_C (C_Programming)</td>
</tr>
</tbody>
</table>

The second matching phase, that considers Graph Analysis and Normalized Google Distance, discovers a further match: Implementazione (Implementation) in version 1.1 matches with Definizione (Definition) in version 1.2. The term Definizione, in this case, is used as a synonymous of Codifica (in English: Coding). The result of the merging operation, applied on versions 1.1 and 1.2, is temporary harmonized ontology. The main structure of the seed ontology is unchanged while the refinements for Principi_basilari concept (coming from version 1.1) and Costrutti (coming from version 1.2) are added as sub-graphs of the seed ontology. The overlapping zones of versions 1.1 and 1.2 are harmonized with respect to the identified matches. Now, it needs to include version 1.3 into the harmonization process. So, professor Gaeta uses the Ontology Merging Tool on the temporary harmonized ontology and version 1.3. In a first phase, the matching algorithm discovers the following matches:

<table>
<thead>
<tr>
<th>Temporary</th>
<th>Version 1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmazione_C (C_Programming)</td>
<td>Programmazione_C (C_Programming)</td>
</tr>
<tr>
<td>Principi_basilari (Basic_principles)</td>
<td>Principi_basilari (Basic_principles)</td>
</tr>
<tr>
<td>Costrutti (Constructs)</td>
<td>Costrutti (Constructs)</td>
</tr>
<tr>
<td>Tipi_dati_strutturati (Structured_data_types)</td>
<td>Tipi_dati_strutturati (Structured_data_types)</td>
</tr>
<tr>
<td>Tipi (Types)</td>
<td>Tipi_dati (Data_types)</td>
</tr>
<tr>
<td>Puntatori (Pointers)</td>
<td>Puntatore (Pointer)</td>
</tr>
</tbody>
</table>

No other matches are deduced by applying Graph Analysis and Normalized Google Distance algorithms. The merging algorithm, using the identified matches, produces the result stored as version 2.0, the Harmonization Activity ends. A last step is required in order to obtain a fully validated C-Programming ontology: all participants have to give their consensus to the harmonized result. For this sake, professor Gaeta loads version 2.0 into a Semantic Wiki Space (predisposed trough the Semantic Wiki Engine)
and notifies to his three assistants the possibility to access the Wiki in order to approve or reject the elements of the reference ontology (version 2.0). The Professor fixes a deadline of three days. During the validation phase the four participants discover the necessity to define two additional relations between:

- Array and Costrutto_struct (Struct_statement)
- Passaggio_per_valore (Passing_by_value) and Passaggio_per_riferimento (Passing_by_reference)

The final C-Programming ontology (version 2.1) is ready to be imported in IWT in order to support personalised e-Learning experiences.

6. Conclusion

In this work we have proposed an approach for the educational domain modeling through the use of ontologies. We have described in details how to construct e-Learning ontologies and how they are exploited in order to define and execute personalised e-Learning experiences. The personalization allows to execute more efficient and effective e-Learning processes. The approach is fully implemented into a commercial e-Learning platform, namely IWT. Further improvements have been thought, designed and prototyped in the context of ELeGI UE Integrated Project [39] from the delivery infrastructure (based on Grid Technologies) and the pedagogical (several educational methods are implemented in order to enrich the IWT e-Learning experiences) points of view. The necessity to improve knowledge management aspects, with more features, tools and methodologies, comes from the results of numerous experimentation cases carried out using the IWT platform. Therefore, investigating the state of the art of Semantic Web, with respect to ontology management, and experimenting existing software tools we have matured the following ideas:

- We cannot force the e-Learning actors to have knowledge engineering competencies and use complex existing tools. Then, a visual, drag and drop based, and user-centric ontology editing technique has been defined.
- If we want to really exploit the potentiality of personalised e-Learning, also within blended learning processes, we have to help the e-Learning actors to effectively reuse knowledge and manage inconsistencies due to the knowledge natural evolution. Then, ontology harmonization, versioning and changes tracking techniques have been defined.
- Ontology development is often a collaborative process. Furthermore, it can be very difficult to use these tools without competencies in knowledge engineering. Then, a collaborative ontology construction technique has been defined. The technique foresees the exploitation of a workflow engine in order to support the tasks coordination. The technique provides a validation phase based on a semantic wiki engine allowing the collaboration participants to reach a consensus about the final ontology. Without the validation phase there is no assurance that all participants agree with the final artifacts. Conversely, using only the semantic wiki engine in order to construct the ontology does not satisfy the simplified task coordination requirement.

Future works will consist of other validation and testing activities in order to improve the potentiality of Semantic Web in the e-Learning area.

References

