ISSN 2007-9737

A Workflow Ontology to Support Knowledge Management in a Group's Organizational Structure

Mario Anzures-García[1], Luz A. Sánchez-Gálvez[1], Miguel J. Hornos[2], Patricia Paderewski-Rodríguez[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, Mexico

[2] Universidad de Granada, Departamento de Lenguajes y Sistemas Informáticos, E.T.S.I., Informática y de Telecomunicación, Granada, Spain

{mario.anzures, sanchez.galvez}@correo.buap.mx, {mhornos, patricia}@ugr.es

Abstract. In CSCW (Computer Supported Cooperative Work), managing the group's organizational structure allows to control how the group members communicate, collaborate, and coordinate, to achieve a common goal, in order to benefit an organization or a community. Consequently, establishing an appropriate model of this structure's management is very important, as it can be used as a guide for implementing these kinds of systems. This modeling must be flexible enough, so that it can conform itself to changes within the group and to adjust to the different working styles of several groups, as well as to formally support a base of knowledge; helping to eradicate any ambiguity or redundancy. Therefore, this modeling must formally provide a knowledge representation in order to specify the elements and to control the set of orderly steps on an organizational structure. Thus, a workflow ontology to control such a structure is proposed in this paper. Since, the workflow manages and controls the process, via a set of steps ordered and executed by different organization entities, whereas the ontology specifies the domain of knowledge through concepts, relations, axioms, and instances in a formal, explicit, way. A case of study, to demonstrate the knowledge management of the group's organizational structure, through workflow ontology is shown.

Keywords. Group's organizational structure, workflow, ontology, workflow ontology, base of knowledge.

1 Introduction

In CSCW domain, the group's organizational structure defines the division of labor, so that the people appropriately perform the process within an organization or community. Since such a structure determines how the interaction among users is carried out; who does what; how are the turns for users' participation defined; what roles a particular user performs, or how will the users access the system.

So, in CSCW systems, the group's organizational structure determines how the communication, collaboration, and coordination among the group members are performed. Then, the management of this structure is crucial for these kinds of systems.

However, specifying and modeling the management of the organizational structure within the group is quite complex, especially when a dynamic enough structure is required. This dynamism refers to change the structure in accordance to tasks and group needs, at a given stage, as well as to adapt it dynamically to cope with the changing conditions of the organization and/or the own application.

Hence, a knowledge representation scheme to support an adaptable, formal, modeling is required. This scheme must provide a set of procedures, which allows knowledge, to be stored, organized, and managed naturally, in accordance to the changing needs within the community or organization. This leads to need a base of knowledge to specify the group's organizational structure.

This base can help to understand, manage, and control every process performed by the organizations or communities; in addition, it can reason and draw conclusions through an inference mechanism for the content of the base of knowledge [1]. The representation scheme must be denoted by a model of some domain of interest, in which symbols assist as substitutes for real world artifacts. These symbols must be stored as interest domain statements. The ideal knowledge representation schemes are ontologies [2].

An ontology is presented as an organization's resource and knowledge representation through an abstract model. This representation model provides a common vocabulary of a domain and defines the meaning of the terms and the relations among them [3]. This supplies a set of concepts or classes, relations, functions, axioms, and instances to describe a domain in a formal, explicit way. Furthermore, it can be adapted by changing, adding or eliminating some of the elements that constitute it. So, the ontology is ideal to establish a base of knowledge that allows specifying and modeling the group's organizational structure.

The management process of this structure can be characterized by a workflow, which specifies how the entities should be used and combined, i.e.; it refers to a coordinated execution of multiple tasks or activities [4, 5]. In addition, ontology is ideal to control a workflow, as it provides the necessary expressivity to represent the entities, its use, combining the correctness of workflow specification by applying a reasoned; and the management of changes by adjusting the ontology elements, as required.

Therefore, a knowledge-based workflow ontology, to manage the group's organizational structure, is proposed in this paper. Towards this end, the knowledge of this structure and special workflow, are formal, and explicitly modeled. Using this knowledge representation scheme and rules, the application can adapt itself to frequent changes within an organizational structure.

The rest of the paper is structured as follows: Section 2, explains the group's organizational structure on the CSCW domain. Section 3, describes briefly the knowledge representation schemes. Section 4, presents the workflows and the workflow ontology. Section 5, specifies the ontologies, its structure, and languages to be expressed. Section 6, details the workflow ontology of the group's organizational structure, and the conceptual proof according to a case study focused on academic virtual space. Section 7, summarizes the conceptual results obtained. Section 8, outlines the conclusions and future work.

2 Group's Organizational Structure

CSCW has focused on developing a range of heterogeneous computing technologies, which are in continual fluctuation, in order to provide appropriate computational notations to model and execute, organizational procedures. As the interaction of a group is regulated by the way in which it is organized, i.e., how group's members communicate, collaborate and coordinate to meet a common goal.

Furthermore, with the upsurge of information and communication technologies (ICT); virtual organizations and communities have emerged. This is a group of individuals whose members and resources may be geographically distributed, even though they function as a coherent unit supported by ICT; supplying shared and often real-time access to centralized or distributed resources. Both types of organizations provide interactive meeting places where people can work or gather around.

Members of virtual organizations carry out a set of planned tasks for achieving a certain goal; in virtual communities, there is a spontaneous gathering of people whose goal is exchanging experiences. However, they share various common characteristics, such as: distribution across space and time, presenting dynamic structures and processes. In addition, these organizations generally comprise a set of rules that determine its structure, which are the means and the outcome of organizational conduct.

This structure can be hierarchical or not. A structure is governed by a policy, which is a set of rules that determine the behavior of a system in terms of the conditions under which; predefined operations or actions can be invoked [6, 7, 8].

In the CSCW systems, the shared work is supported for sessions, which denote the shared workspace. These systems typically provide a

shared workspace by a session manager. On the one hand, this manager allows to establish the session (i.e., it permits to set up the connection, to create and manage meetings, and to enable a user to join and leave a session using a simple user interface). On the other hand, this manager allows defining the group's organizational structure that states how sessions are organized to accomplish the shared work [9].

In general, CSCW systems do not separate the mechanisms to establish the shared workspace from the group's organizational structure. Therefore, this separation has been considered in the proposal [9]; because it allows us to support changes in the group at runtime, and specify this organizational structure through a policy, which can be implemented, providing a broad variety of policies to users. Three of the most important, are mentioned here:

- **Brainstorming Policy.** It is designed for informal collaborations among identical hierarchy users. Consequently, users must have voting tools to decide, who establishes the turns for their participation, or if such policy is determined on a first-come-first-participating basis. This policy operates similarly to instant messaging applications.
- **Moderate Policy.** This is designed for a structured group work controlled by a single person (the designated moderator). The moderator has an upper status than other users; he/she controls and coordinates the session and establishes turns for users' participation.
- **Customized Policy.** It is designed for specific CSCW systems, in which users can play several roles or to exchange them, in order to achieve the common goal of the organization or community.

On the other hand, systems based on policies for management applications, are of particular importance, because they allow the separation of the rules that govern the behavior of a system, from the functionality provided by that system [10]. So, it is possible to support dynamic and complex systems, by changing the policies. A system can be continuously adjusted to accommodating variations in externally imposed constraints and environment conditions.

Policy-based systems management, requires a semantic approach that simplifies policy analysis and reduces inconsistencies and conflicts; to facilitate policy reuse across various systems [8]. Ontology is ideal for specifying policy-based systems management, since it uses concepts to characterize the domain and components being controlled, by simplifying their description and facilitating the analysis and the careful reasoning over them.

CSCW systems are typically large-scale systems, which require solutions in accordance with the behavioral changes of the system, in order to facilitate the work management of the group´s members. The session management policy has been modeled by ontology [11]. On the one hand, this ontology supports and develops, satisfactorily, an organizational structure, so that the structure can adapt itself to changes of group work, and to the different working styles of several organizations or communities. On the other hand, this paper is based on such ontology; which outspreads to cope with all aspects related to creation of CSCW systems.

3 Base of Knowledge

Given that knowledge is a portion of all human activities, it is necessary to store it by seizing its meaning, organize it and make it available. This leads to require a base of knowledge to represent the problem domain, as well as be able to reason and draw conclusions through an inference mechanism for the contents of such a base [11]. Therefore, it is essential to capture knowledge related to the system requirements, i.e. base of knowledge that enables to specify design elements, starting relationships and restrictions on this knowledge.

Knowledge is an important factor in systems development, principally in the CSCW domain, as it allows creating and modifying collaborative systems in accordance with group necessities and dynamic collaborative scenarios. CSCW domain has mainly focused on knowledge sharing, as well as on systems that can support it. In addition, in CSCW considers that work and knowledge are closely intertwined, i.e.; the development of CSCW systems must generate a base of knowledge

supported by an analysis of work practices and complexity of real organizations and communities.

Consequently, the base of knowledge requires a representation scheme to provide a set of procedures, which allows the knowledge, to be stored, organized, and to represent the problem naturally.

The representation scheme must be denoted by a model of some domain of interest in which symbols assist as substitutes for real world artifacts. These symbols must be stored as interest domain statements. The knowledge representation schemes [2] are:

- **Semantic Network.** This is appropriate for capturing the taxonomic structure of categories for domain objects, and for expressing general statements about the domain of interest. Nevertheless, the representation of concrete individuals or even data values does not fit well the idea of semantic networks.
- **Frames.** It represents a concept consisting of slots for which fillers are specified. The reasoning in frame-based systems, involves both intentional and extensional knowledge contained within the base of knowledge of the frame. However, the frames provide more expressive power but less capacity to infer.
- **Rules.** These come in the form of IF-THEN-constructs and allow to express various kinds of complex statements. Rule-based knowledge representation systems are especially suitable for reasoning about concrete instance data. Complex sets of rules can efficiently derive implicit such as facts from explicitly given ones. They are problematic if more complex and general statements about the domain shall be derived, which do not fit a rule's head [2].
- **Logic.** It is the dominant form of knowledge representation, because it is used to provide a precise formalization and axiomatization of problem domain, which is ideal for representing and processing knowledge within computers in a meaningful way. Nowadays, all symbolic knowledge representation and reasoning formalisms can be understood in their relation to First-order (predicate) logic; therefore, this is the prevalent and single most important knowledge representation and reasoning formalism. First-order logic, allows describing the domain of interest consisting of objects, i.e. things that have individual identity, and to construct logical formulas around these objects formed by predicates, functions, variables and logical connectives [12]. Description logic [13], is essentially a set of decidable fragments of first-order logic, and is expressive enough such that it has become a major knowledge representation and reasoning paradigm. A description logic theory consists of statements about concepts, individuals, and their relations. Individuals correspond to constants in first-order logic, and concepts correspond to unary predicates. Concepts can be named concepts or anonymous (composite), concepts. Named concepts consist simply of a name, which will be mapped to a unary predicate in first-order logic. Composite concepts are formed from named concepts by using concept constructors, similar to the formation of complex formulas out of atomic formulas in first-order logic [13].

Ontology is an ideal solution to represent the knowledge domain, using description logic symbols, which allow to specify it in a simple way, which is readable for both humans and machines; as well as to perform much deeper reasoning through the machine [11]. It facilitates a base of knowledge in order to provide semantic, common understanding, communication and knowledge sharing on the domain of interest and a knowledge reasoning, carrying out an inference process to reach to conclusions on this base by a reasoned, inference rules and query languages.

4 Workflow

Workflow is seen as an automation of a business process, in whole or part, during which, documents, information or tasks are passed from one participant to another, for action, according to a set of procedural rules [14, 15].

The workflow management systems have received considerable attention lately, due to the wide spectrum of applications of the workflow. These systems achieve the management of the workflow through three constructs: routes (it represents task sequences); rules (it defines routing and role constructs); and roles (it represents one, who is responsible for a task).

A suitable management of a workflow requires the following aspects [16]:

- **Expressiveness:** It should provide constructs to represent conditional mapping relationships between roles and actors, based on the organizational model, as well as complex business rules, including exceptional rules.
- **Model verification:** It should allow analysis that assures the correctness of a workflow specification, along with checking the occurrence of inconsistent, redundant, and incomplete rules as well as non-terminally of processes.
- **Change management:** It should allow easy development of the propagation mechanisms against changes on the organizational structure and rules, as well as organizational procedures to assure the correctness of a workflow model.

In addition, standardization efforts of workflow management systems have been presented by various consortiums, such as Workflow Management Coalition (WfMC), Advancing Open Standards for the Information Society (OASIS), and World Wide Web Consortium (W3C). Among other contributions, these efforts resulted in new workflow definition languages, and coordination protocols.

However, there is a lack of a flexible, formal model to control the coordinated execution of different business process activities.

5 Ontologies

There are several definitions of ontology that have different connotations depending on the specific domain. This paper will refer to Gruber's well-known definition [3], where an ontology is a formal and explicit specification of a shared conceptualization. *Conceptualization* refers to an abstract model of some phenomenon throughout the world, by identifying the relevant concepts of this. *Explicit specification* means that the type of concepts used, and the constraints on their use are explicitly defined. Thus the ontology is a high level formal specification of a certain knowledge domain, providing a simplified and well defined view of domain.

On the other hand, Jasper and Ushold [17], identify four main categories of ontology applications: 1) neutral authoring, 2) ontology-based specification, 3) common access to information, and 4) ontology-based search. Although, in this work, the principal idea is the ontology-based specification of the group's organizational structure management.

5.1 Ontology Structure

The domain knowledge in ontologies can be formalized using five kind of components [3]:

- **Classes:** Set of classes (or concepts) that belong to the ontology. They may contain individuals (or instances), other classes, or a combination of both with their corresponding attributes.
- **Relations:** These define interrelations between two or several classes (object properties) or a concept to a data type (data type properties).
- **Functions:** This is a special case of relations.
- **Axioms:** These are used to impose constraints on the values of classes or instances. Axioms represent expressions in ontology (logical statement) and are always true when used inside the ontology.
- **Instances:** These represent the objects, elements or individuals of an ontology.

Nowadays, the ontologies (particularly in OWL, Ontology Web Language) have been extended with rules by Semantic Web Rule Language (SWRL), which use other predicates than just class or property names [18]:

- **Class Expressions:** These are arbitrary class expressions, not just named classes.
- **Property Expressions:** The only operator available in OWL 2, for creating property expressions is inverse of object property; however, the same effect can be achieved by exchanging the property arguments, so there is no need to use property expressions in SWRL.
- **Data Range Restrictions:** They specify a type of data value, such as integer, date, union of some XML Schema types, and enumerated type.

- **SameIndividual and DifferentIndividuals:** These are used for specifying same and different individuals.
- **Core SWRL Built-ins:** They are special predicates defined in SWRL proposal, which can manipulate data values, for example, to add numbers custom SWRL built-ins, it can define its own built-ins using Java's code.

5.2 Ontology Languages

As the knowledge representation and reasoning, ontologies require a logical and formal language to be expressed. In the area of Artificial Intelligence, many languages have been developed for this purpose [19]:

- **First-order (predicate) Logic-Based Languages.** Such as KIF and Cycl, providing modeling primitives and the possibility of redoing formulas that enable them to become in terms of other formulas.
- **Frames-Based Languages.** These have more expressive power, but less inference capability as Ontolingua and F-Logic.
- **Descriptive Logic-Based Languages.** Which are more robust in the power of reasoning as a Loom, OIL, DAML + OIL and OWL [20]. Specifically, OWL is an ontology language recommended by the W3C for use in the Semantic Web. The OWL representational facilities are directly based on Description Logics. This basis confers OWL a logical framework, including both syntax and model-theoretic semantics, allowing a knowledge representation language, capable of supporting a base of knowledge and a practical, effective reasoning. Moreover, the Description Logic provides readily available reasoners such as Pellet [21], and HermiT [22], both of which have been extended to handle all of OWL ontologies, which can also be combined with rules using the new W3C Rule Interchange Format (RIF), standard [23].

The development of ontologies is a laborious and error-prone task, especially when done manually, so it is necessary to have tools that can automate some of this work and hide the features and formalisms for ontology specification languages. These tools provide graphical interfaces that facilitate the knowledge representation and reasoning, such as: Ontolingua server [24], WebOnto [25], OilEd [26], OntoSaurus [27], Protégé [28], Swoop [29], TopBraid Composer [30], WebODE [31], OntoEdit [32], Neon Toolkit [33].

This article focuses on Protégé, which is an engineering tool open source ontology and a knowledge-based framework. Protégé is widely used in the development of ontologies, due to the scalability and extensibility with lots of plugins; and by facilitating inference knowledge through reasoners, query languages and rules. Ontologies in Protégé can be developed in a variety of formats, including OWL, RDF (S), and XML Schema.

5.3 Ontologies in the CSCW and Workflow Domain

There are little works related to ontologies in the domain of CSCW systems, such as: AMENITIES [34], which is a methodology to support the development of this kind of systems based on model-driven architecture. It includes the definition of a domain ontology, from which application ontologies can be derived; providing computer independent models for each system. Nevertheless, this methodology emphasizes high-level concepts and does not provide a set of steps for the development of a CSCW system. SAKE [35], uses the approach of knowledge-based systems and purposes to develop an e-government system, which is adaptable to changing needs. The used approach is founded on ontologies for several different purposes: information ontology, change ontology, etc. But Sake does not present a formal model of the structure organizational.

GTA [36], presents an ontology to model task analysis in CSCW systems, which considers concepts such as role, task, event, object, agent and goal. However, this proposal does not consider, the development of CSCW systems. An ontology for context representation in collaborative systems developed is presented in [37]. This ontology is used by a logic-based reasoning mechanism to recommend tools, founded on the present context of group users. Nonetheless, this ontology is focused on context in CSCW systems.

Other applied ontologies can also be used to specify the situation awareness (it refers to awareness in real-life scenarios) [38], which is an important aspect in CSCW systems. In [39], an ontological approach for awareness based on a model-driven development method is presented. The ontology includes a series of ontologies previously developed. Another ontology for situation awareness with a series of rules for automatic inference is shown in [40].

Besides, with respect to workflow management systems, some works have been presented. Recently, the workflow management automation has been possible with well-formed workflow specifications. Consequently, the ontology provides enough expressivity by the supplied structure (concepts, relations, instances, and axioms); the ontology verification is accomplished through reasoners (Pellet, and HermiT). The change management on the ontology can be made by modifying, adding, and/or deleting concepts, relations, and/or instances.

Hence, some workflow ontologies have been presented for this management, as follows: An approach based on a rule ontology for workflow change management is described in [41]; another [42], uses two ontologies: one, for processes description, and another for domain description, which are independent of each other, this approach describes a project that extends the scope of the current workflow management systems, adding to the workflow model information contained in these ontologies; another one for flexibilizing the workflow execution is shown in [43], this presents an architecture for the workflow system, which is driven by ontologies that capture semantic relationships between workflows, resources and users. All these works do not focus on the group's organizational structure in the domain CSCW.

Finally, workflow ontologies, recently have been paid special attention. In [45], a collaborative workflow for terminology extraction, and a collaborative modeling of formal ontologies using two tools (Protege and OntoLancs [44]), is presented. It allows the development of cooperatives, and distributed ontologies, based on dependencies management among ontology modules. In [46], an ontology-based workflow for ontology collaborative development in Protégé is shown, which presents the combination of workflows with ontologies to design, in a formal manner, protocols for laboratories [47]. In [48], a workflow ontology for the preservation of digital material produced by an organization or a file system is proposed.

All these works are focused, exclusively, on building workflow ontologies to represent collaborative work on different areas. It should be pointed out, that this paper presents a workflow ontology to manage collaborative work by using the group structure organizational, which represents a novel work on the CSCW domain.

6 Workflow Ontology for Group's Organizational Structure

This paper proposes a workflow ontology for having an appropriate base of knowledge for the structured representation of the CSCW systems development in a formal way. Furthermore, this ontology not only provides the static and dynamic structure of a group's organization, but it also specifies the steps to develop this kind of systems and allows its adaptation. This workflow ontology was developed through an extended literature review, which included studies and surveys from multiple venues, such as journals, conferences, and workshops; from which, the terms that constitute the base of knowledge were extracted.

6.1 Workflow Ontology Description

This workflow ontology (WO, see Fig. 1), defines that: the group's organizational structure (GOS), is made up of users, and is governed by one policy (Pcy), which establishes a hierarchical organizational structure or not-hierarchical organizational structure by means of the roles (one or more, Rls), that users can play.

Each role designing one status (Stt, which founds the role priority in the group), one right/obligation (R/O, set privileges for the user in the application), and a tasks set (Tsk, which are role functions), and they can be composed of one or more activities (Atv, which are operations that allow users to achieve a given goal) that uses any resources (Rsc).

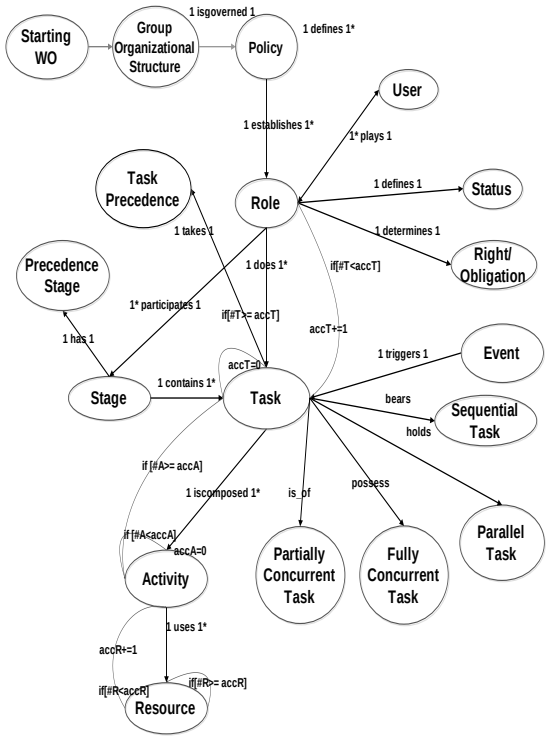

Fig. 1. Workflow ontology of the group's organizational structure

Each task indicates the event (Evt), that triggers it, its precedence (TaP, i.e., tasks order), and its type (*Sequential-task, SqT;* one activity follows the other. *Parallel-Task, PrT*; these happen at the same time, but they use different objects, and no interference between them can occur. *Partially-Concurrent-Task, PCT;* it refers to tasks that can be active at the same time but there is no simultaneous modification of any object). *Fully-Concurrent-Task, FCT;* it occurs when two or more simultaneous tasks modify the rights of the same set of objects).

It establishes the system stages (Stg, it reflects each of the collaboration moments). For each stage determines the order of them (Stage Precedence, SPc), the tasks that correspond to these, and the precedence of the tasks (STk).

The specification of the Workflow Ontology is carried out through the following steps (WOS):

1. Starting Workflow (StW)
2. Defining the GOS name.
3. Determining the Policy name.
4. Establishing the Roles of the CSCW system.
   - 4.1. Determining the user who plays this role.
   - 4.2. Designing a Status to role.
   - 4.3. Signalizing a Right/Obligation to role.
   - 4.4. Specifying the tasks that each role carries out in the CSCW system.
     - 4.4.1. Designating the event that triggers each Task.
     - 4.4.2. Indicating the task type (sequential, parallel, partially concurrent, and fully concurrent).
     - 4.4.3. Mark out the Activities of each Task.
       - 4.4.3.1. Defining the resources to the activity.
       - 4.4.3.2. If there are more resources go to step 4.3.3.1, otherwise go to step 4.4.4.
     - 4.4.4. If there are more activities of one task go to step 4.3.3, otherwise go to step 4.5.
   - 4.5. If there are more tasks for one role, go to step 4.4, otherwise go to step 5.
5. If there are more roles for the application go to step 4, otherwise go to step 6.
6. Establishing the Stage name of the CSCW system.
   - 6.1. Determining the order of the stage.
   - 6.2. Assigning tasks to a stage.
   - 6.3. Indicating the tasks' precedence in each stage.
   - 6.4. If there are more stages, go to step 6, otherwise go to step 7.
7. If you require to create another CSCW system, go to step 1, otherwise go to step 8.
8. Finishing.

6.2 Proof Conceptual of the Workflow Ontology

The study case consists of the development of a groupware for Managing Departmental Test (MDT) in the *Facultad de Ciencias de la Computación de la Universidad Autónoma de Puebla*. The Departmental Test (DET) homogenizes the teaching of a subject, i.e. it guarantees that all teachers encompass the same percentage on the academic program. For this reason, it requires a shared workspace that allows professors to manage and apply a DET.

For reasons of space, the workflow ontology, which displays the knowledge representation in a conceptual, formal manner; and the Table 1, that shows the workflow ontology elements, will be presented in a partial form. Several roles are considered in MDT:

- The Manager (Mgr), who configures the application (CfA), and has status equal to 1; so, he/she authenticates (AMgr), him/herself; registers the (RAC) Area Coordinator (ArC); Delete the ArC (DAC); register (RKA), modifies (MKA), and eliminate knowledge areas (EKA); and finally, he/she register (RSA), modifies (MSA), and eliminate subjects that are a part of them (ESA).
- The Area Coordinator (ArC), with status 2, who manages the test (MaT), and does the following tasks: he/she authenticates (AAC) him/herself, registers (RTC), the Test Coordinator (TeC), deletes to TeC (DTC), schedules the professors' meetings (SPM), proposes number of departmental tests (NDT) by subject that are a part of its area, downloads grades by area (DoG), generates reports (GeR), loading statistics (LoS), posts messages (PoM), posts notices (PoN), and schedules test (ScT).
- The *Test Coordinator (TeC),* with status 3, organizes the test (OgT); and does the following tasks: he/she authenticates (ATC) him/herself, TeC registers (RPf) the Professor (Pfs), deletes to Pfs (DPf), agrees on the number of tests to be applied (ANT), as well as on the dates (ADT), and the number of questions (ANQ), to be included; then he/she will post the test (PoT) and the classroom (PoC), where each Pfs will apply it. He/she, also downloads grades by test (DGT), generates reports (GRT), loads statistics (LST), posts messages (PMT), and posts notices (PNT).
- The Professor with status 4, generates the test (GeT), and does the following tasks: he/she authenticates (AuP) him/herself, registers (RSd) the Students (Sds), proposes (PPD) and votes (PVD) on the date when the test will be performed, and the number of questions included in the test (NQT); he/she, also, consults proposals (CoP), choses date (ChD), loads proposal of questions (LPQ), downloads exercises of the test (Dex), choses questions of the test (ChQ), posts notice (PNP), and posts message (PMP); as well as, he/she loads grades (LGR), posts messages (PMR), and posts notices (PNR).
- The Students (Sds), with status 5, consult information (CoI), and do the following tasks: they authenticate (AuS), themselves, view the date (CDT), and the classroom (CCT), when and where the test, will be carried out, and once they have presented it, they will also look up the grades obtained (CGO).

The collaborative application for managing the MDT has four stages (Stg):

1. *Test Configuration (TCf) with stage precedence (SPc) equal to 1. In this stage only the role Mgr participates; executing the tasks:*

- AMgr takes task precedence (TaP), with value equal to 1, which is triggered by the event accesses to the system (EAM), it is composed by the following activities: it logs user name (LUN, using as resource a text box, TB), it logs password (LPs, using as resource a password box, PB), and it sends data (SeD, using as resource a button, B).
- RAC takes TaP value equal to 2, which is triggered by the event manages to ArC (EMA), it is composed by the following activities: it enters data (EDT, using as resource a TB), and it sends ArC registration (SeR, using as resource a register button, RB).
- DAC takes TaP value equal to 0, which is triggered by the event remove to ArC (ERA), it is composed by the following activities: it selects ArC (SRe, using as resource an ArC list, RL), and it deletes ArC (DRe, using as resource a eliminate button, EB).
- RKA takes TaP value equal to 3, which is triggered by the event adds to area (EAA), it is composed by the following activities: it enters area data (EDA), using as resource a TB), and it sends area registration (SAR, using as resource an area bottom, RA).
- MKA takes TaP value equal to 0, which is triggered by the event updates to area (EUA), it is composed by the following activities: it selects registration (SRA, using as resource a registration list, AL), it selects fields to modify (SFA, using as resource fields list, FL), and it

Table 1. Workflow Ontology Specification

WOS	Cpt	CoI	Cat	CAI	Rel	Cdy	TaC	TCI	TCA	TAI	Rule
1	STW										
2	GOS	GOS-MDT	NoumGOS	1	isgoverned	1	Pcy	PMDT	Num Pcy	1	if [[GOS](?X) and [Pcy](?Y)] (?X,?Y)] then [isgoverned Pcy] (?X,?Y)
3	Pcy	PMDT	NumPcy	1	establishes	1*	Rls	Mgr, ArC, TcC, Pfs, Sds	#R, accR	5	If [[Pcy](?X) and [Rls](?Y)] (?X,?Y)] then if [accR<=#R] then establishes Rls and accR+=1 else if [[Stg](?Z) and [Tsk](?W)](?Z,?W)] then contains Tsk] (?Z,?W)] (?X,?Y)
4	Rls	TcC		1	defines	1	Stt	3			if [[Rls](?X) and [Stt](?Y)] (?X,?Y)] then defines Stt] (?X,?Y) and con+=1;
4.1	Rls	TcC		1	determines	1	R/O	OgT			if [[Rls](?X) and [R/O](?Y)] (?X,?Y)] then determines R/O] (?X,?Y)
4.2	Rls	TcC		1	does	1*	Tsk	Aut, PD, SD, PNQ, SNQ, PQ	#T, accT	5	if [[Rls](?X) and [Tsk](?Y)] (?X,?Y)] then if [[accT<=#T] then does Tsk and accT+=1; else if [[Pcy](?W) and [Rls](?Z)] (?W,?Z)] then establishes Rls] (?X,?Y).
4.2.1	Evt	ScM		1	triggers	1	Tsk	DTD			if [[Evt](?X) and [Tsk](?Y)] (?X,?Y)]then [trigger Tsk] (?X,?Y)
4.2.2	Tsk	SD		1	is_composed	1*	Atv	eD, CD, uD	#Atv, accA	3	if [[Tsk](?X) and [Atv](?Y)] (?X,?Y)] then [is_composed Atv] (?X,?Y)
4.2.2	Atv	uD		1	uses	1*	Rsc	AB	#A, accC	1	if [[Act](?X) and [Rsc](?Y)](?X,?Y)] then [uses Rsc and accA+=1] (?X,?Y); if [[Rsc](?Z) and [accC>=Rsc]](?Z) then [conA*=1 and goes [[Act](?X) and [Rsc](?Y)](?X,?Y)]
4.2.3	Tsk	SD		1	takes	1	PTk	3			if [[Tsk](?X) and [PTk](?Y)](?X,?Y)] then [takes PTk] (?X,?Y)
5	Stg	TeE		1	contains	1*	Tsk	Tcf, TeE, TcC, TEr	#S, accS	4	if [[Stg](?X) and [Tsk](?Y)](?X,?Y)] then contains Tsk] (?X,?Y)
	Stg	TeE		1	supports	1	SPc	3			f [[Stg](?X) and [SPc](?Y)](?X,?Y)] then supports SPc] (?X,?Y)

sends change (SCA, using as resource a change button, CB).

- EKA takes TaP value equal to 0, which is triggered by the event eliminates to area (EEA), it is composed by the following activities: it selects to area (Saa, using as resource area list, AL), and it eliminates area (Eaa, using as resource area button, AB).
- RSA takes TaP value equal to 4, which is triggered by the event adds subject (EAS), it is composed by the following activities: it enters subject data (ESD, using as resource TB), and it sends subject registration (SSR, using as resource a subject button, SB).
- MSA takes TaP value equal to 0, which is triggered by the event updates to subject (EUS), it is composed by the following activities: it

selects subject (SSu, using as resource a subject list, SL), it selects subject fields to modify (SSF, using as resource a subject fields list, SFL), and it sends subject change (SSC, using as resource a subject button, SuB).

- ESA takes TaP value equal to 0, which is triggered by the event eliminates to subject (EES), it is composed by the following activities: (SSu, using as resource a subject list, SL), and it eliminates subject (ESu, using as resource subject button, SBu).

2. *Test Preparation (TeP) with SPc equal to 2, and three roles joining:*

- ArC performing the tasks:
- AAC takes TaP value equal to 1, which is triggered by event MGR accesses to the system (MAS), it is composed by the following activities: it logs user name (LUN, using as resource a text box, TB), it logs password (LPs, using as resource a password box, PB), and it sends data (SeD, using as resource a button, B).
- RTC takes TaP value equal to 2, which is triggered by the event manages to TeC (EMM), it is composed by the following activities: it enters TeC data (EMD, using as resource TB), and it sends TeC registration (SeM, using as resource a TeC register button, MRB).
- DTC takes TaP value equal to 0, which is triggered by the event remove to TeC (ERT), it is composed by the following activities: it selects TeC (STe, using as resource a TeC list, TL), and it deletes TeC (DTe, using as resource a TeC eliminate button, TEB).
- SPM takes TaP value equal to 3, which is triggered by the event schedule meeting (ESM), it is composed by the following activities: it selects date meeting (SDM, using as resource a date list, DaL), and it posts date (PoD, using as resource a button to post date, BPD).
- NDT takes TaP value equal to 4, which is triggered by the event number of test (ENT), it is composed by the following activities: it sends proposals of test number (SPN, using as resource a test number list, TNL), Pfs select a number (PSN, using as resource TNL), and it post select number (POS using as resource a buttoto post number, BPN).
- TeC does the tasks:
- Pfs does the tasks:
  - AuP takes TaP value equal to 7, which is triggered by event Pfs accesses to the system (PAS), it is composed by the following activities: it logs user name (LUN, using as resource a text box, TB), it logs password (LPs, using as resource a password box, PB), and it sends data (SeD, using as resource a button, B). RSd takes TaP value equal to 8, which is triggered by the event manages to Sds (EMS), it is composed by the following activities: it enters Sds data (ESs, using as resource TB), and it sends Sds registration (SSs, using as resource a Sds register button, SRB).
  - PPD takes TaP value equal to 9, which is triggered by the event proposes date (EPa), it is composed by the activity: it sends test date (STD, using as resource calendar, C).
  - PVD takes TaP value equal to 10, which is triggered by the event vote date (EVD), it is composed by the activity: it selects test date (STa, using as resource a test date list, TDL), and it posts date (PDa, using as resource date, D).
  - NQT takes TaP value equal to 11, which is triggered by the event test questions (ETQ), it is composed by the activity: it sends test question (STQ, using as resource a question, Q), it votes by test questions (VTQ, using as resource a questions list, QL), and it posts test questions (PTQ, using as resource a questions selected, QS).

3. *Test Elaborating (TeE), with SPc equal to 3, and two roles joining:*

- TeC doing the tasks:
  - ANT takes TaP value equal to 2, which is triggered by the event test to applied (ETA), it is composed by the following activities: it selects test to be applied (STA, using as resource a test number list, TNL), and it send number of test to apply (NTA, using as resource a TNL).
  - ADT takes TaP value equal to 3, which is triggered by the event test data (VTA), it is

composed by the following activities: it selects test data (LTA, using as resource a test data list, TDL), and it send test data (NTD, using as resource a data, Dat).

- ANQ takes TaP value equal to 4, which is triggered by the event number question (ENQ), it is composed by the following activities: it selects questions number (LTA, using as resource a test data list, TDL), and it send test data (NTD, using as resource a data, Dat).
- PoT takes TaP value equal to 5, which is triggered by the event posts test (EPT), it is composed by the activity: it posts test data (PTA, using as resource a C).
- PoC takes TaP value equal to 6, which is triggered by the event posts classroom (EPC), it is composed by the activity: it posts test classroom (PTC, using as resource a classroom list, CLL).

– Pfs doing the tasks:

- CoP takes TaP value equal to 7, which is triggered by the event consults data (CoD), it is composed by the activity: it reviews data (ReD, using as resource a test data list, TDL).
- ChD takes TaP value equal to 8, which is triggered by the event choses data (CDa), it is composed by the activity: it selects data (SDa, using as resource a data list, TDL).
- LPQ takes TaP value equal to 9, which is triggered by the event proposals of questions (EPQ), it is composed by the activity: it downloads question (DoQ, using as resource a data list, QL), and it sends question (SeQ, using as resource a question button, QB).
- ChQ takes TaP value equal to 10, which is triggered by the event choses question (ECQ), it is composed by the activity: it selects question (SQu, using as resource a questions list, QL).
- Dex takes TaP value equal to 11, which is triggered by the event downloads question (EDQ), it is composed by the activity: it selects question (SQu, using as resource a questions list, QL), and it downloads question (SDQ, using as resource a questions list, Q).
- PNP takes TaP value equal to 0, which is triggered by the event posts advice (EPA), PNP is composed by the activity: it posts advice of Pfs (PAP, using as resource an advice, adv).
- PMP takes TaP value equal to 0, which is triggered by the event posts message (EPM), PMP is composed by the activity: it posts message of (PMe, using as resource a message, Mss).

4. *Test Results (TeR), with SPc equal to 4. Four roles (ArC, TeC, Pfs, and Sds), participate in this stage:*

– The role ArC does the tasks:

- DoG takes TaP value equal to 1, which is triggered by the event downloads grades by area (EGA), DoG is composed by the activity: it selects area file (SAF, using as resource a file, Fil).
- GeR takes TaP value equal to 0, which is triggered by the event generates report by area (GRA), GeR is composed by the activity: it loads report (LRe, using as resource a report, Rep).
- LoS takes TaP value equal to 0, which is triggered by the event loads statistics by area (LSA), LoS is composed by the activity: it loads statistics (LSt, using as resource statistics, RSa).
- PMP above mentioned.
- PNP above mentioned.
- ScT takes TaP value equal to 2, which is triggered by the event schedules test (EvS), ScT is composed by the activity: it is composed by the following activities: it selects date test (SDt, using as resource a date list, DaL), and it posts date (PoD, using as resource a button to post date, BPD).

– The role TeC to perform the tasks:

- DoG above mentioned.
- GeR above mentioned.
- LoS above mentioned.

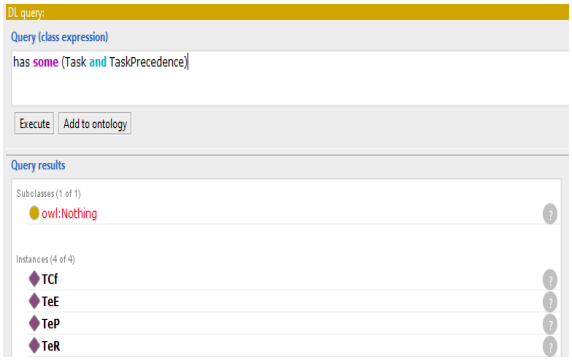

Fig. 2. Query carried out in Protégé of the stages

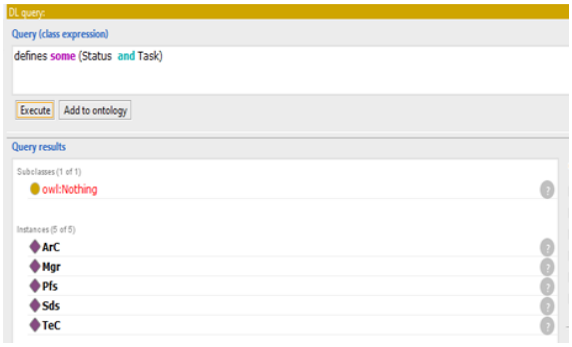

Fig. 3. Query carried out in Protégé of the roles

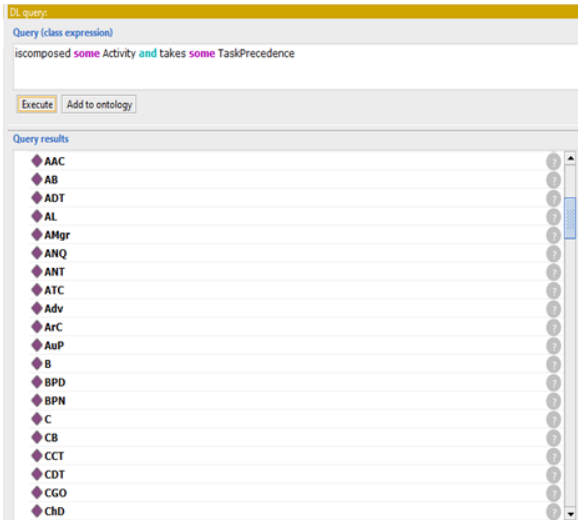

Fig. 4. Query carried out in Protégé of the Task

  - PMP above mentioned.
  - PNP above mentioned.
- The role Pfs effects the tasks:
  - LGR takes TaP value equal to 6, which is triggered by the event loads grade (ELG), LGR is composed by the activity: it loads grade by subject (LGS, using as resource a grade file, GaF).
  - PMP above mentioned.
  - PNP above mentioned.
- The role Sds carries out the task:
  - CDT takes TaP value equal to 7, which is triggered by the event views test data (EVD), CDT is composed by the activity: it shows test data (ShD, using as resource a message, DL).
  - CCT takes TaP value equal to 8, which is triggered by the event views the test classroom (EVC), CCT is composed by the activity: it shows the test classroom (ShC, using as resource a message, CL).
  - CGO takes TaP value equal to 9, which is triggered by the event looks up the grades obtained (EVO), CGO is composed by the activity: it shows the grades obtained (ShO, using as resource a GL).
  - PMP above mentioned.
  - PNP above mentioned.

Table 1, shows the workflow ontology elements that constitute the base of knowledge and that are gotten of the case study with respect to the Stage TeE and the role TcC. This table is expressed in terms of the ontology specification, as well as, of the rules that determine the execution flow of the steps to be performed by this workflow.

Therefore, this table presents the following columns; allowing to proof the ontology: Concepts (Cpt), Concept Instance (CoI), Concept Attribute (CAt), Concept Attribute Instance (CAI), Relation (Rel), Cardinality (Cdy), Target Concept (TaC), Target Concept Instance (TCI), Target Concept Attribute (TCA), Target concept Attribute Instance (TAI), and Rules (Rul).

7 Conceptual Results

The workflow ontology allows to know: the roles which participate in an interaction (fully concurrent, partially concurrent, parallel, and/or sequential),

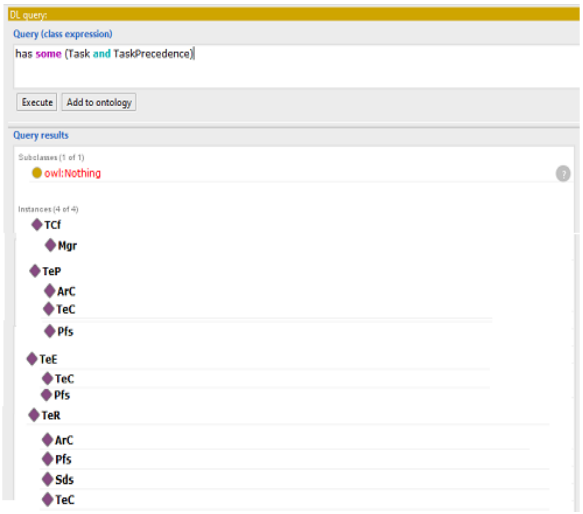

Fig. 5. The roles participating on each stage

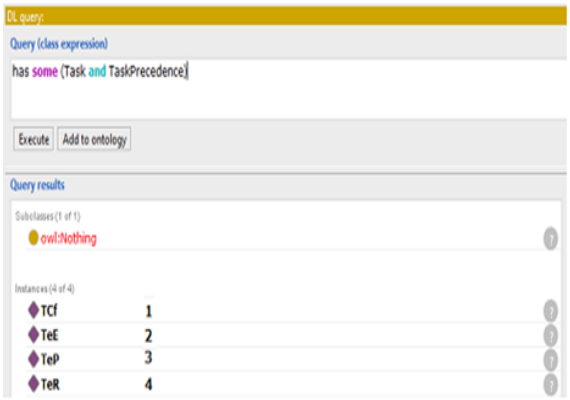

Fig. 6. The priority of each stage

the order in which they do so, and the resources used by each user for accomplishing each activity. This is possible, because this ontology establishes the following:

1. The stages of group's organizational structure (see Figure 2).
2. Roles of group's organizational structure (see Figure 3).
3. The tasks of group's organizational structure (see Figure 4).
4. The roles that access to each stage (see Figure 5).
5. The priority of each stage (see Figure 6).

8 Conclusions and Future Work

In this paper, a workflow ontology to manage the group's organizational structure has been presented. On the one hand, the workflow ontology allows to model and specify in a formal, explicit way, the group's organizational structure in the CSCW domain, on the other hand, it provides a set of orderly steps to control and manage such a structure. This ontology allows to build a base of knowledge that can be adapted to the changes within the group, as well as the different working styles of several groups. The future work is orientated to specify a methodology to develop groupware that extends the workflow ontology described in this article.

References

1. **Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M.J., & Paderewski-Rodríguez, P. (2015).** A Base of knowledge for the Development of Collaborative Applications. *Engineering Letters*, Vol. 23, No. 2, pp. 65–71.
2. **Grimm, S., Hitzler, P., & Abecker, A. (2007).** Knowledge Representation and Ontologies. In: Semantic Web Services: Concepts. *Technologies and Applications*, Springer-Verlag, pp. 51–105. DOI: 10.1007/3-540-70894-4_3.
3. **Gruber, R. (1993).** A translation approach to portable ontology specification. *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199–220. DOI: 10.1006/knac.1993.1008.
4. **Fischer, L. (2004).** *Workflow Handbook.* Future Strategies Inc., Lighthouse Point, FL.
5. **Marinescu, D.C. (2002).** *Internet-Based Workflow Management: Toward a Semantic Web.* Wiley, New York.
6. **Moffett, J.D. & Sloman, M.S. (1991).** The representation of policies as system objects. *Proceedings SIGOIS,* Vol. 12, No. 2-3, pp. 71–84. DOI: 10.1145/127769.122850.
7. **Moffett, J.D. & Sloman, M.S. (1993).** Policy hierarchies for distributed systems management. *IEEE Journal on Selected Areas in Communications,* Vol. 11, No. 9, pp. 1404–1414. DOI: 10.1109/49.257932.
8. **Sloman, M. (1994).** Policy driven management for distributed systems. *J. Network System Management,* Vol. 2, No. 4, pp. 333–360. DOI: 10.1007/BF02283186.

9. **Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M.J., & Paderewski-Rodríguez, P. (2007).** Ontology-Based Modelling of Session Management Policies for Groupware Applications. *Lecture Notes in Computer Science,* No. 4739, pp. 57–64. DOI:10.1007/978-3-540-75867-9_8.
10. **Wright, S., Chadha, R., & Lapiotis, G. (2002).** Special Issue on Policy based Networking. *IEEE Network,* Vol. 16, No. 2, pp. 8–9. DOI: 10.1109/MNET.2002.993217.
11. **Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M.J., & Paderewski-Rodríguez, P. (2016).** Knowledge-based Workflow Ontology for Group's organizational structure. *Research in Computing Science: Advances in Languages and Knowledge Engineering,* Vol. 123, pp. 79–90.
12. **Russel, S. & Norvig, P. (1995).** *Artificial Intelligence – A Modern Approach.* Prentice-Hall.
13. **Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P.F. (2003).** *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.
14. **Allen, R. (2001).** *Workflow: An introduction*. Workflow Handbook, L. Fisher, Ed. Future Strategies, Lighthouse Point, FL, pp. 15–38.
15. **Marshak, R.T. (1994).** An overview of workflow structure*. Proc. of Workflow,* Future Strategies Inc., pp. 13–42.
16. **Lee, H.B., Kim. J.W., & Park, S.J. (1999).** KWM: Knowledge-based Workflow Model for Agile Organization. *Journal of Intelligent Information Systems*, Vol. 13, No. 3, pp. 261–278. DOI: 10.1023/A:1008773617579.
17. **Uschold, M. & Gruninger, M. (2004).** Ontologies and Semantics for Seamless Connectivity. *SIGMOD Record,* Vol. 33, No. 4, pp. 58–74. DOI: 10.1145/1041410.1041420.
18. **SWRL (2004)**. *A Semantic Web Rule Language Combining OWL and RuleML*. https://www.w3.org/ Submission/SWRL/.
19. **Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004).** *Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.
20. **Patel-Schneider, P.F.; Hayes, P., & Horrocks, I. (2004).** OWL Web Ontology Language semantics and abstract syntax. *W3C Recommendation*.
21. **Pellet (2003).** OWL reasoner. Maryland Information and Network Dynamics Lab, http://www. mindswap.org/2003/pellet/index.shtml.
22. **Motik, B., Shearer, R., & Horrocks, I. (2007).** Optimized reasoning in description logics using hypertableaux. *Lecture Notes in Artificial Intelligence, Springer,* Vol. 4603, pp. 67-83. DOI: 10.1007/978-3-540-73595-3_6.
23. **RIF (2010).** *RDF and OWL Compatibility. W3C Recommendation*. http://www.w3.org/TR/rif-rdf-owl/
24. **Farquhar-Fikes, R. & Rice, J. (1997).** The Ontolingua Server: A Tool for Collaborative Ontology Construction. *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Vol. 46, No. 6, pp. 707–727. DOI:10.1006/ijhc.1996.0121.
25. **Dominguez, T. (1998).** Discussing, Browsing and Editing Ontologies on the Web. *Proceedings of the Eleventh Knowledge Acquisition Workshop.*
26. **Bechhofer, S., Horrocks, I., Goble, C., & Stevens, R. (2001).** OILEd: a Reason-able Ontology Editor for the Semantic Web. *Proceedings of KI2001, Joint German/Austrian Conference on Artificial Intelligence, LNAI,* Vol. 2174, pp. 396–408. DOI: 10.1007/3-540-45422-5_28.
27. **Swartout, B., Ramesh, P., Knight, K., & Russ, T. (1997).** Toward Distributed Use of Large-Scale Ontologies. *Symposium on Ontological Engineering of AAAI.*
28. **Musen, M.A., Fergerson, R.W., Grosso, W.E., Noy, N.F., Grubezy, M.Y., & Gennari, J.H. (2000).** Component-based support for building knowledge-acquisition systems. *Proceedings of the intelligent information processing, conference international federation for processing, world computer congress (WCC'2000),* pp. 18–22.
29. **SWOOP**. http://www.mindswap.org/2004/ SWOOP/
30. **Horridge, M. (2011).** *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools.* Edition 1.3.
31. **Gómez-Pérez, A., Fernández-López, M., Corcho, O., & Aspiréz, J. (2001).** WebODE: A scalable ontological engineering workbench. *First International Conference on Knowledge Capture*.
32. **Sure, Y., Angele, J., & Staab, S. (2002).** OntoEdit: Guiding Ontology Development by Methodology and Inferencing. *Proceedings of the International Conference on Ontologies. Databases and Applications of Semantics*, *LNCS* 2519, pp. 1205-1222. DOI: 10.1007/3-540-36124-3_76.
33. **NeOn.** http://neon-toolkit.org/wiki/Main_Page.
34. **Garrido, J.L., Gea, M., Padilla, N., Canas, J.J., & Waern, Y. (2003).** AMENITIES: Modelo de entornos cooperativos. *Actas del III Congreso Internacional Interacción Persona-Ordenador*, pp. 97-104.

35. **Butka, P., Lukac, G., Mach, M., & Sabol, T. (2009).** Semantic-based groupware system for collaborative support. *Acta Electrotechnica et Informatica,* Vol. 9, No. 4, pp. 9–16.
36. **van Welie, M. & van der Veer, G.C. (2003).** Groupware task analysis. **E. Hollnagel (ed.),** *Handbook of cognitive task design*, pp. 447–476.
37. **Vieira, V., Tedesco, P., & Salgado, A.C. (2005).** Towards an ontology for context representation in groupware. **H. Fuks, S. Lukosch, & A. C. Salgado (Eds.)**, *CRIWG LNCS,* Springer, Vol. 3706, pp. 367–375. DOI: 10.1007/11560296_30.
38. **Feng, Y.H., Teng, T.H., & Tan, A.H. (2009).** Modelling situation awareness for context-aware decision support. *Expert Systems with Applications,* Vol. 36, No. 1, pp. 455–463. DOI: 10.1016/j.eswa.2007.09.061.
39. **Gallardo, J., Molina, A.I., Bravo, C., Redondo, M.A., & Collazos, C.A. (2011).** An ontological conceptualization approach for awareness in domain-independent collaborative modeling systems: Application to a model-driven development method. *Expert Systems with Applications,* Vol. 38, No. 2, pp. 1099–1118. DOI: 10.1016/j.eswa.2010.05.005.
40. **Kokar, M.K., Matheus, C.J., & Baclawski, K. (2009).** Ontology-based situation awareness. *Information Fusion,* Vol. 10, No. 1, pp. 83–98. DOI: 10.1016/j.inffus.2007.01.004.
41. **Chun, S.A. & Atluri, V. (2003).** Ontology-based Workflow Change Management for Flexible eGovernment Service Delivery. *Proceedings of the Third National Conference on Digital Government,* pp. 1–4.
42. **Chung, P.W.H., Cheung, L., Stader, J., Jarvis, P., Moore, J., & Macintosh, A. (2003).** Knowledge-based Process Management – an Approach to Handling Adaptive Workflow. *Knowledge Based Systems*, Vol. 16, No. 3, pp. 149–160. DOI: 10.1016/S0950-7051(02)00080-1.
43. **Almeida, T., Vieira, S.C., Casanova, M.A., & Ferrao, L.G. (2004).** An Ontology-driven Architecture for Flexible Workflow Execution. *Proceedings of the WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress,* pp. 70–77. DOI:10.1109/WEBMED.2004.1348150.
44. **Gacitua, R., Arguello-Casteleiro, M., Sawyer, P., Des, J., Perez, R., Fernandez-Prieto, M.J., & Paniagua, H. (2009).** A collaborative workflow for building ontologies: A case study in the biomedical field. *Research Challenges in Information Science,* Vol. 121, pp. 22–24. DOI:10.1109/RCIS.2009.5089275.
45. **Kozaki, K., Sunagawa, E., Kitamura, Y., & Mizoguchi, R. (2007).** A Framework for Cooperative Ontology Construction Based on Dependency Management of Modules. *ESOE, CEUR Workshop Proceedings*, Vol. 292, pp. 33–44.
46. **Sebastian, A., Noy, N.F., Tudorache, T., & Musen, M.A. (2008).** A Generic Ontology for Collaborative Ontology-Development Workflows. *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns,* pp. 318–328. DOI: 10.1007/978-3-540-87696-0_28.
47. **Maccagnan, A., Riva, M., Feltrin, E., Simionati, B., Vardanega, T., Valle, G., & Cannata, N. (2010).** Combining ontologies and workflows to design formal protocols for biological laboratories. *Automated Experimentation*, Vol. 2, No. 3. DOI: 10.1186/1759-4499-2-3.
48. **Mikelakis, M. & Papatheodorou, C. (2012).** An ontology-based model for preservation workflows. *Proceedings of the 9th International Conference on Digital Preservation.*

Article received on 08/08/2016; accepted on 12/10/2016.
Corresponding author is Mario Anzures-García.