Ontology development for computersupported collaborative learning scripts

Aliki Papakonstantinou, Stavros Demetriadis, Nick Bassiliades

ONTOLOGY DEVELOPMENT FOR COMPUTER-SUPPORTED COLLABORATIVE LEARNING SCRIPTS

ALIKI PAPAKONSTANTINOU, STAVROS DEMETRIADIS,
NICK BASSILIADES

Department of Informatics, Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
email: alikipap@csd.auth.gr; sdemetri@csd.auth.gr; nbassil@csd.auth.gr

This paper presents an ongoing effort to develop an ontology for Computer-Supported Collaborative Learning (CSCL) Scripts. Our work merges the field of collaborative learning with the field of semantic web and provides a framework for the formalization of collaboration scripts using the OWL language. Collaboration scripts are didactic scenarios that prescribe learners' interactions in collaborative settings. A script comprises a number of phases and each phase specifies the activity that learners have to perform, the composition of the group, the distribution of the activity, the mode of interaction and the phase duration. Scripts can also be positioned along various design dimensions, such as their granularity, coercion degree and locus of control. The presented ontology is being developed with the purpose of formalizing scripts in order to promote their reusability and portability between various computer-supported learning platforms. We discuss design decisions and illustrate how this ontology could be beneficial when embedded in a knowledge-based system that supports collaborative learning.

Keywords: ontology; computer supported collaborative learning; collaboration script

1 INTRODUCTION

Collaborative learning refers to methodologies and environments, in which learners take part in a common activity. During the activity, each individual depends on and is responsible to each other. Computer-Supported Collaborative Learning (CSCL) is a research field on supporting collaborative learning with the help of computers (Dillenbourg, 1999). CSCL supports and facilitates group processes and group dynamics in ways that are not possible by face-to-face interactions, however, without replacing this kind of communication. This type of learning is typically tailored for use by multiple learners, working either synchronously or asynchronously, at the same workstation or across networked computers. The purpose of Computer-Supported Collaborative Learning is to scaffold or support students in learning together successfully (Stahl, Koschmann, & Suthers, 2006).

The scope of our work is to merge the field of collaborative learning with the field of

semantic web and develop a script ontology as a framework for the formalization of collaboration scripts that can be used by various computer-supported learning platforms. In order to achieve that, we reviewed the literature and developed the ontology using the OWL language. The ontology presented is an original one and, as far as we know, no other ontology on collaboration scripts has been reported to the literature so far.

An ontology is a semantically enriched data model that represents a set of concepts within a domain and the relationships between those concepts (Antoniou & Van Harmelen, 2004). It provides a common vocabulary to refer to the concepts of a domain; it specifies relationships using logical statements that describe how the concepts are related and also provides rules for combining concepts and their relations to define extensions to the vocabulary.

The Web Ontology Language OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It uses the XML/RDF exchange syntax (McGuinness & Van Harmelen, 2004). It is the most recent development in standard ontology languages, certified by the World Wide Web Consortium (W3C) to promote the Semantic Web vision. OWL is used by applications that need to process the content of information instead of just presenting information to humans (McGuinness & Van Harmelen, 2004). OWL goes beyond the languages XML and RDF in their ability to represent machine interpretable content on the Web, because it has more facilities for expressing meaning and semantics than the latter. (McGuinness & Van Harmelen, 2004).

In the following, we present the theoretical background of computer-supported collaborative learning scripts, we describe how our ontology is designed, analyzing every class and property, and, finally, we illustrate how this ontology could be beneficial for users when embedded in a knowledge-based system for supporting collaborative learning.

2 CSCL SCRIPTS

Scripted collaboration differs from other collaborative learning methods mostly in the scripts' ability to specify the cognitive activities that learners are expected to engage in, while the majority of other methods leave these activities unspecified or indistinguishable (O'Donnel, 1999). Dillenbourg (2004a) defines CSCL scripts as "instructional sequences in which peer interactions are expected to constitute the core learning mechanism". Scripts are collaboration didactic scenarios (O'Donnell & Dansereau, 1992) that organize learning activities by specifying tasks and subtasks, their duration, the role of each collaborating learner in the group and other relevant parameters of the collaboration. Each script is analyzed into a number of phases and each phase prescribes the tasks and the activities that learners and tutors have to perform and the goals that have to be achieved. The reason for employing scripts to guide collaboration is that usually unguided collaboration among team members can lead to detrimental learning. Kollar, Fischer & Slotta (2005, p. 332) argue that "the lack of explicit scaffolds for collaboration could result in unequal participation of learning partners and ineffective argumentation". Scripts are expected to facilitate learning by guiding peer collaboration and engaging all participants in activities that trigger the activation of their cognitive and metacognitive processes.

2.1 Schemata

Based on script common features one can classify them into script-schemata, which are patterns that describe the script structure at a more abstract level. Our work refers to eight major script-schemata. The Jigsaw schema prescribes the partition of the knowledge or information necessary to solve the task, either by forming pairs who have complementary knowledge (Hoppe & Ploetzner, 1999) or by providing them with complementary information or by asking them to play complementary roles (Dillenbourg, 2002). The Conflict schema triggers argumentation among group members by forming pairs of learners with conflicting opinions (Dillenbourg & Jermann, 2003). The Negotiation schema involves justifying and entitling viewpoints during collaborative interaction, negotiating about them and trying to convince other team members (Dillenbourg, 1999). The Reciprocal schema defines two roles in teams, one of the peers regulating the other and then switching roles (Palincsar & Brown, 1984). The Regulation schema urges learners to set goals for their learning and then attempt to plan, monitor, regulate and control their cognition, motivation, behaviour and context (Pintrich, 2000). The Inquiry schema organizes a classroom to function as a scientific research community and guides learners to participate in practices of progressive scientific dialogue (Scardamalia & Bereiter, 1996). The Competition schema pushes learners to work together and makes them responsible for their teammates' learning as well as for their own one (Slavin, 1980). Finally, the SWISH (Split Where Interaction Should Happen) schema triggers the interactions that teachers want to foster by splitting tasks to facilitate a shared understanding (Dillenbourg, 2004b).

2.2 Phases

Each phase in a collaboration script is defined by five specific items: the activity that learners have to perform, the group composition, the activity distribution, the interaction mode and the timing of the phase (Dillenbourg, 2002). Moreover, each phase is differentiated in the dimensions of granularity, coercion degree and locus of control (Dillenbourg, 2004a). The "granularity" of a script refers to the grain size of the scripted activity, in terms of time scale of the script (from minutes to whole semester) and to the grain size of sub-tasks definition. The "degree of coercion" refers to what extent learners are forced to follow the collaboration script, i.e. the extent that it constraints the actions of the student. Usually scripts with high granularity tend to be more coercive. For example, a script guiding closely the student through the whole session is considered as highly coercive. Finally, the "locus of control" refers to whether a script possesses some metacognitive value and should be internalized by learners (the locus of control is internal) or it is simply a step by step procedure to be executed by the group (the locus of control is external).

2.3 Roles and activities

The main purpose of roles in collaboration scripts is to prescribe privileges and obligations of students during collaboration (Kobbe, 2005). Baggetun et al. (2004) suggested that a role can be described mostly by two parts. The first part is its responsibilities and privileges within the scope of the scripted collaboration. The responsibilities and privileges of a role are specified by assigning activities and by defining the performable actions within them. The second part is its membership. A role can be assigned to a single person or a

group. Furthermore, a role can divided into sub-roles, which usually inherit the responsibilities and privileges of the role.

An activity refers to learning tasks and interactions between roles that take place during the scripted collaborative learning. Activities are conceptualized as forming a hierarchy; each activity can be decomposed into more fine-grained activities and any lower level activity can be subsumed by one or more higher level (low-grained) activities (Kobbe, 2005).

3 ONTOLOGY DESCRIPTION

Figure 1 presents the current classes and subclasses of the ontology. The upper classes (top level) are *Phase*, *Schema*, *Role* and *Activity*. Class *Schema* has 8 subclasses (middle level) and each of these subclasses has its own subclasses (bottom level). The hierarchy has been designed according to the aforementioned theoretical framework of CSCL scripts. From the ontology it is apparent that some script subclasses may belong to more than one class (script-schemata). For example the class *The_Escape_The_Maze_Script* is an intersection of classes *Competition* and *Regulation*, and essentially this means that this script possesses features that belong to either of the two script-schemata. Respectively, there are scripts that inherit their features from only one script-schema. For example the classes *The_ArgueGraph_Script* and *The_Construction_of_Argumentation_Sequences_Script* belong solely to the *Conflict* class.
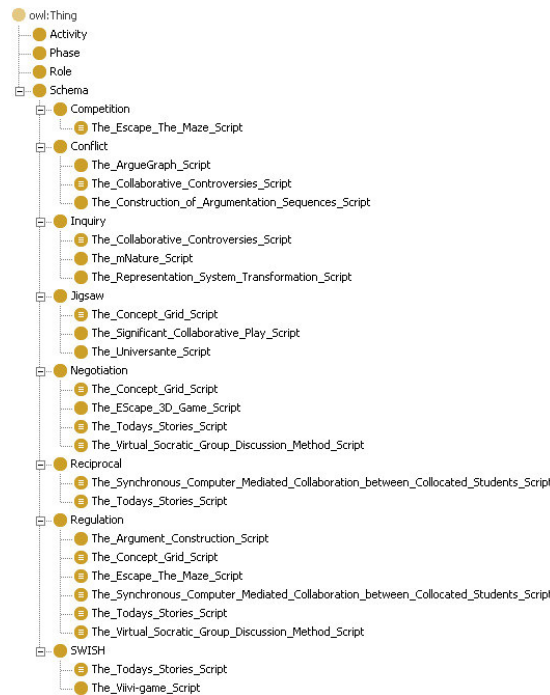

FIGURE 1. Classes & subclasses of the CSCL script ontology

Class *Phase* has a number of properties depicted in Figure 2. Data properties of this class are *Phase_Description*, *Phase_Group*, *Fade_Out_Level*, *Coersion_Degree*, *Granularity*, *Locus_of_Control, Activity_Distribution* and *Timing*.
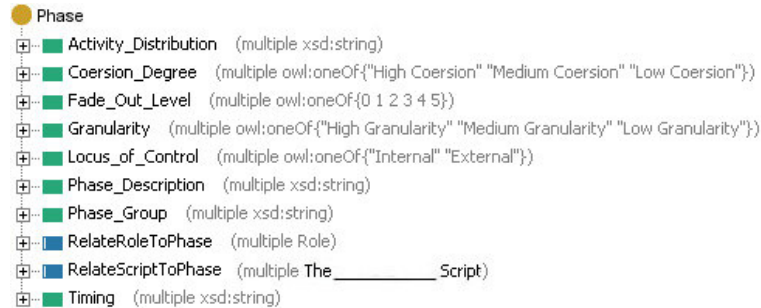

FIGURE 2. Properties of phase class

Data properties *Coersion_Degree* and *Granularity* take only the values High, Medium and Low. Property *Locus_of_Control* acquires only the values Internal and External and property *Fade_Out_Level* have numeral values from 0 to 5 and it describes the level of "script persistence" to guide learners. This practically means that a "non-fading" script consistently appears to students while a "faded-out" script is available only if students ask for it.

Object properties *RelateRoleToPhase* and *RelateScriptToPhase* belong also to the *Phase* class. The purpose of the *RelateRoleToPhase* property is to correlate each role with the phase it participates.The *RelateScriptToPhase* property relates every phase to the script it belongs. This is important because scripts are described as a sequence of their phases.

Figure 3 presents the properties of any script class of the type *The_........_Script* (for instance *The_Concept_Grid_Script*) (the general naming format is denoted by the long underscore).
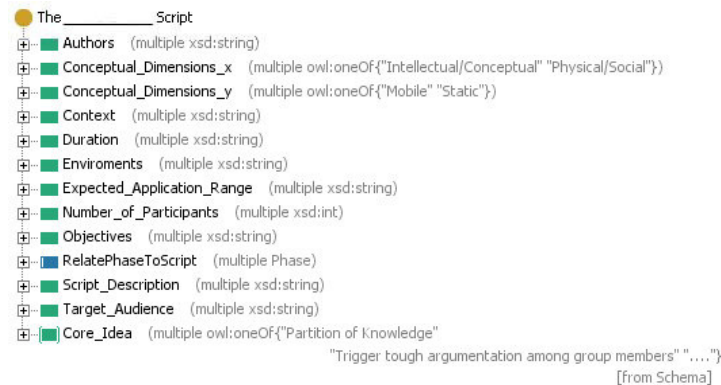

FIGURE 3. Properties of the_......._script class

Data properties of this class are *Script_Description, Authors, Context, Duration, Environments, Expected_Application_Range, Objectives, Number_of_Participants, Target_Audience, Conceptual_Dimensions_x, Conceptual_Dimensions_y* and *Core_Idea. The_......._Script* class has also the object property *RelatePhaseToScript*, which is the inverse property of *RelateScriptToPhase*.

The *Objectives* property describes the learning objectives of the script related also to students' cognitive skills, property *Environments* refers to the technological infrastructure ofthe used CSCL environment and the *Expected_Application_Range* property specifies the possible domains of application and also contains expectations about improvement of specific learning environments or situations. The *Core_Idea* property is inherited by the *Schema* class and it describes the main schema idea.

The data property *Conceptual_Dimensions _x* takes values Intellectual/Conceptual and Physical/Social, while the data property *Conceptual_Dimensions_y* takes values Mobile and Static. Conceptual dimensions set a different organization framework for scripts (Figure 4).
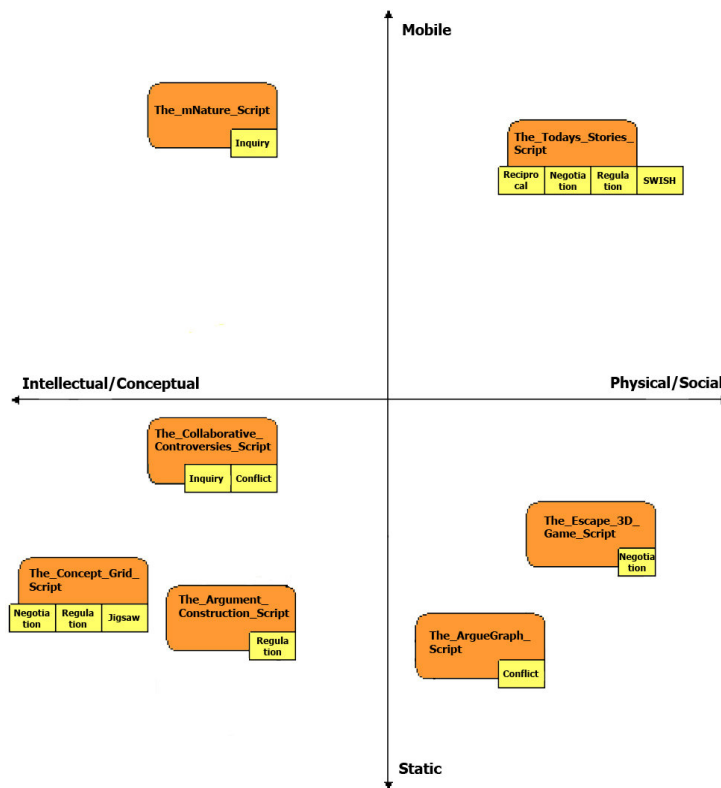

FIGURE 4. An example of how scripts are positioned on conceptual dimension space
(adapted from Stegmann, 2004)

Ontology Development for Computer Supported Collaborative Learning Scripts

Intellectual/Conceptual and Physical/Social are the two poles in dimension x and they position the script according to its major learning objective. If the script is nearer to the Intellectual/Conceptual pole then the objective is to obtain a piece of knowledge or a specific intellectual ability. By contrast, script positioning nearer to the Physical/Social pole means that the objective is to obtain a skill for being in the world. This objective is characterized by physical and mental attitudes in social situations (Stegmann, 2004).

Respectively Static and Mobile are the two poles in dimension y and they position the script according to the importance of being able to change or to choose the context or location of a learning interaction or cooperation. If a script is positioned nearer to the Static end of the range this means that learning occurs in a closed repeating interaction between learning tool and learner, while nearer to the Mobile end of the range learning occurs in a more open repeating interaction between learning tool and learner. This is achieved by changing the context while interaction happens (Stegmann, 2004).

Figure 5 shows the properties of the *Role* class. Data properties of this class are *Role_Description*, *Expectation*, *Obligation*, *Privilege*. The *Role* class also includes the object properties *RelateActivityToRole*, which connects activity to role, and *RelatePhaseToRole*, which is the inverse property of *RelateRoleToPhase*.
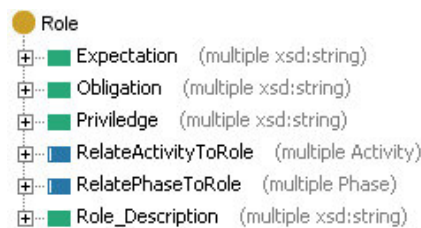

FIGURE 5. Properties of Role class

Figure 6 presents the properties of class *Activity*. Data properties of this class are *Description*, *Deliverable*, *Resource*, *Tool*, *Activity_Duration, t-start* and *t-end*.

FIGURE 6. Properties of Activity class

Aliki Papakonstantinou; Stavros Demetriadis; Nick Bassiliades   497

The *t-start* property describes the time that an activity starts and property *t-end* describes the time that this activity ends. The *Activity* class also includes the object property *RelateRoleToActivity*, which is the inverse of *RelateActivityToRole*.

4   USE CASES

The main benefit from using an ontology is that a common format for developing machine-processable representations of knowledge can be used across various systems, thus enabling the exchange of information between systems and the proliferation of common tools to process the represented knowledge. The collaboration script ontology is proposed and developed with the purpose of formalizing scripts in order to promote their reusability and portability between different computer-based environments for collaborative learning. To better illustrate why such an effort can be beneficial for CSCL we comment in the following on the use cases of web-based systems that would take advantage of the script ontology.

Scripts are expected to be developed and used by educators and instructional designers. Appropriate software tools - like a script editor - can facilitate instructors to create and edit scripts (as instances of the script classes included in the ontology), thus reifying and communicating their ideas on how a collaborative learning session can be organized and scaffolded. Using scripts to further guide students' collaborative sessions can produce data on the effectiveness and efficiency of scripted collaborative learning interactions among the team members. Thus repositories where script-related information is archived can also include information on the effectiveness of the scripts in specific situations and for learners of specific profile. In order this information to be also exploited by any other computer-based system related to CSCL, it has to conform to a common representation format such as an ontology can define. If this is implemented then transferring scripts and using them to other systems could be easily accomplished; for example, a script developed and evaluated in a specific domain could be spotted by an instructor while searching for script-related resources, transferred to another web-based system and evaluated in another context and domain. Furthermore, an ontology-based system will have the ability to query the knowledge database and facilitate system users in their effort to pinpoint scripts with specific features that could be used in their learning design. For example questions like: "which scripts could be appropriate in a specific domain and for a specific user profile? Is ArgueGraph a Conflict or Jigsaw script? What is the locus of control of the ArgueGraph script?" could be easily answered by an ontology-based system.

Overall, we expect that by building ontology-based environments for collaborative learning we promote the formalization of tools for collaboration (such as scripts) in a way that improves our ability to search, retrieve and process data relevant to collaborative learning sessions. This may help system developers to establish transferability between their systems, educators to design and replicate fruitful collaborative activities in various systems and learners to engage in collaborative activities better adapted to their personal learning needs.

5   CONCLUSIONS

This paper presented an ontology-based framework for formalizing computer-based collaborative learning scripts. The ontology organizes semantically several script-schemata, script classes and also the phases, the roles and the activities that constitute a script. It is

argued that the CSCL script ontology can become a key component in a knowledge-based system that aims to efficiently support students and instructors in collaborative learning tasks.

REFERENCES

1   Antoniou, G. & Van Harmelen, F. (2004). *A Semantic Web Primer*, The MIT Press

2   Baggetun, R., Barros, B., Fesakis, G., Girardin, F., Hoeksema, K., Hämäläinen, R., Miao, Y. & Vantroys, T. (2004). A review of schemes for modelling scripts. *Kaleidoscope Network of Excellence deliverable D23.2.1.*

3   Brousseau, G. (1998). *Théorie des situations didactiques*. Grenoble, La Pensée Sauvage

4   Dillenbourg, P. & Jermann, P. (2003). Elaborating new arguments through a CSCL scenario. In G. Andriessen, M. Baker & D. Suthers (Eds.), *Arguing to learn: Confronting cognitions in Computer-Supported Collaborative Learning environments*. CSCL series. Amsterdam: Kluwer

5   Dillenbourg, P. (1999). Introduction: What do you mean by "collaborative learning"? In P. Dillenbourg (Ed.) *Collaborative learning: Cognitive and Computational Approaches*, p. 1-19. Pergamon: Oxford

6   Dillenbourg, P. (2002). Overscripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.) *Three worlds of CSCL. Can we support CSCL*, p. 61-91. Heerlen, Open Universiteit Nederland

7   Dillenbourg, P. (2004a) (Ed). Framework for integrated learning. *Kaleidoscope Network of Excellence deliverable D23.5.1.*

8   Dillenbourg, P. (2004b). "Split Where Interaction Should Happen", a model for designing CSCL scripts. In P. Gerjets, P. A. Kirschner, J. Elen & R. Joiner (Eds.), *Instructional design for effective and enjoyable computer- supported learning*. Tuebingen: Knowledge Media Reseach Center.

9   Hoppe, U.H. & Ploetzner, R. (1999). Can analytic models support learning in groups. In P. Dillenbourg (Ed.) *Collaborative-leaning: Cognitive and Computational Approaches*, p. 147-168. Oxford: Elsevier

10 Kobbe, L. (2005). Framework on multiple goal dimensions for computer-supported scripts. *Kaleidoscope Network of Excellence deliverable D29.2.1.*

11 Kollar, I., Fischer, F. & Hesse, F. W. (2006). Collaboration Scripts – A Conceptual Analysis. *Educational Psychology Review*. 18:159–185.

12 Kollar, I., Fischer, F., and Slotta, J.D. (2005). Internal and External Collaboration Scripts in Web-based Science Learning at Schools. In T. Koschmann, D. Suthers, & T.W. Chan (Eds.) *Computer Supported Collaborative Learning: The Next 10 Years!* Mahwah, NJ: Lawrence Erlbaum.

13 McGuinness, D.L. & Van Harmelen, F. (2004). *OWL Web Ontology Language Overview. W3C Recommendation*. The MIT Press.

14 O' Donnel, A.M. (1999). Structuring dyadic interaction through scripted cooperation. In A.M. O' Donnel & A. King (Eds.), *Cognitive perspectives on peer learning*, p. 179-196. Mahwah, NJ, US: Lawrence Erlbaum Associates

15 O' Donnell, A.M. & Dansereau, D.F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Hertz-Lazarowitz and N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning*, p. 120-141. London: Cambridge University Press

16 Palinscar, A.S. & Brown A.L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, vol 1, no 2, p. 117-175

17 Pintrich, P.R. (2000). Educational psychology at the Millennium: A look back and a look forward. *Educational Psychologist*, 35, p. 221-226

18 Scardamalia, M. & Bereiter, C. (1996). Computer support for knowledge-building communities. In T. Koschmann (Ed.) *CSCL: Theory and practice of an emerging paradigm*, p. 249-268. Mahwah, New Jersey: Lawrence Erlbaum Associates

19 Schmitz, C., Staab, S., Studer, R., Stumme, G. & Tane J. (2003). The Courseware Watchdog: an Ontology-based tool for finding and organizing learning material. In Proc. *Mobile learning and researching,* Kassel, Germany

20 Slavin, R.E. (1980). *Using student team learning*. Baltimore: The center for social organization of schools, The John's Hopkins University

21 Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences*, p. 409-426. Cambridge, UK: Cambridge University Press.

22 Stegmann, K. (2004) (Ed.). Examples of CSCL scripts using mobile tools. *Kaleidoscope Network of Excellence deliverable D23.3.1.*