From a Pattern Language to a Pattern Ontology Approach for CSCL Script Design

Jonathan Chacón, Davinia Hernández-Leo, Josep Blat

ICT Department, Universitat Pompeu Fabra, C/Roc Boronat 138, 08018 Barcelona, Spain
{jonathan.chacon, davinia.hernandez, josep.blat}@upf.edu

Abstract. Collaborative activities, in which students actively interact with each other, have proved to provide significant learning benefits. In Computer-Supported Collaborative Learning (CSCL), these collaborative activities are assisted by technologies. However, the use of computers does not guarantee collaboration, as free collaboration does not necessary lead to fruitful learning. Therefore, practitioners need to design CSCL scripts that structure the collaborative settings so that they promote learning. However, not all teachers have the technical and pedagogical background needed to design such scripts. With the aim of assisting teachers in designing effective CSCL scripts, we propose a model to support the selection of reusable good practices (formulated as patterns) so that they can be used as a starting point for their own designs. This model is based on a pattern ontology that computationally represents the knowledge captured on a pattern language for the design of CSCL scripts. A preliminary evaluation of the proposed approach is provided with two examples based on a set of meaningful interrelated patters computationally represented with the pattern ontology, and a paper prototyping experience carried out with two teaches. The results offer interesting insights towards the implementation of the pattern ontology in software tools.

Keywords: CSCL, Pattern Ontology, Pattern Selection, Collaborative Learning Design, Pattern Language.

1 Introduction

The research presented in this paper is framed in the Technology-Enhanced Learning (TEL) research domain, which focuses on how advanced Information and Communication Technologies can improve teaching and learning. In particular, this work is within the context of Computer-Supported Collaborative Learning, a discipline inside TEL that promotes the use of technology to support collaborative learning activities [11, 15].

Supporting the creation of potentially effective learning designs is a relevant topic in TEL, but it is even more relevant in CSCL because of the inherent complexity of collaborative dynamics [1]. Achieving effective collaboration in learning has a number of implications that go beyond the provision of communication tools that allow students to interact with each other. Furthermore, free collaboration does not necessarily triggers effective learning. A solution proposed in the literature to face this problem is the creation of meaningful CSCL scripts to enhance the effectiveness of collaboration. CSCL scripts are designs that specify how students should form groups, how they should interact and collaborate along a sequence of activities and which resources and tools they should use in each activity in order to solve the presented problems or tasks [1, 2, 7].

However, the creation and design of CSCL scripts is not trivial. It is particularly difficult for educators interested but not experts in Information and Communication Technologies (ICT) and collaborative learning. There are a number of evidences in the literature that show the challenges around teachers designing scripts from scratch [3]. A solution to face this problem, which is being proposed in different literature contributions, is to support the design process through the reuse of existing material to create new ones. According to [5] there are different levels where reutilization could be applied: we can reuse a whole exemplar that is ready-to-run [12]; we can reuse generalizations of successful collaboration scripts formulated as patterns; or we can reuse learning objects / resources as components or building blocks. This paper is focused on the use of design patterns, since they seem to be a highly reusable type of learning design solutions [2, 7].

A design pattern provides a means of organizing information regarding a contextualized common problem and the essence of its broadly accepted solution, so that it can be repetitively applied. These patterns can be provided as templates in authoring tools, if formalized using computer interpretable notation. A collection of interconnected (related) patterns which enable the generation of a coherent whole is called a Pattern Language (PL) [2, 4].

However, supporting reusability is not trivial since reuse-oriented design processes pose a number of challenges that need to be tackled in order to be successful. These challenges include the definition of assembling and refining processes when combining and particularizing the reusable material as well as addressing the searching and selection of the materials themselves [5]. When combining a set of selected materials, it is critical that the result is “harmonic”, i.e., a meaningful coherent design. This paper aims at providing a solution that facilitates educators to select of a set of patterns for the design of a meaningful CSCL script. The result is expected to be potentially effective since the script is based in good practices (patterns) and the educators creating the script do not necessarily need to be experts in the use of ICT and in scripted collaborative learning.

The approach followed in this paper to face this aim exploits the semantic of the patterns to computationally relate the knowledge captured in those patterns. In this paper, we propose to move from pattern languages to pattern ontologies and present pattern ontology for the design of CSCL scripts. The pattern ontology is illustrated with two sample CSCL scripts. Moreover, the paper also sketches the integration of the ontology in a tool to assist the selection of interrelated patterns for the design of a script reusing those patterns. To identify relevant issues to be considered in that integration, a paper prototyping with two real users has been carried out.

The remainder of the paper is organized as follows. In section 2, we present the state of the art on reusing learning designs that triggered our work. In this section we explain the research gap in which the contributions of this paper are situated, mention the related work and formulate the specific problem tackled. In section 3 we present the pattern ontology and discuss the approach that will be followed to integrate the ontology in a tool that will enable the selection of a set of interrelated patterns. Section 4 is devoted to preliminarily evaluate the proposals of the paper. In particular, it includes two sample CSCL scripts that illustrate the ontology and a paper prototyping experience providing insights regarding the use of the ontology. Finally, section 5 describes the main conclusions of the paper and the future work.

2 Background and problem formulation

In this section we describe more deeply the “designing by reusing” approach, when particularized to CSCL, that motivate our research. Furthermore, we discuss two significant examples in CSCL that use ontology as their basis. Finally, we formulate the specific research problem faced in this paper.

2.1 Designing by reusing

ICT are being introduced in schools progressively, and practitioners are being encouraged to create material and design new practices using these technologies. In this context, the application of CSCL and, particularly CSCL scripts, is an interesting option that would enhance the teaching and learning processes in the class. However, the process of designing a CSCL script is not an easy task for practitioners. Not all of them are both familiar with technology itself and the CSCL approach. In order to facilitate this task the solution proposed in this paper adopts the “designing by reusing” approach, which has already proved to be useful in the literature [5]. This approach proposes to start reusing already existing and proved-to-be-good material instead of starting from scratch. Doing so, practitioners neither familiar with technologies nor with CSCL approach can have a solid help to design their own CSCL scripts.

The TEL literature has applied and studied different levels of reusability [5]. According to these studied levels of reusability, the options to support the creation by reusing of CSCL scripts would be:

- **CSCL Exemplars** as ready-to-run scripts. These scripts may embrace from one activity session to a whole course. Practitioners could select the exemplar and use it directly, or they could use the exemplar as starting point and change it according to their needs.

- **Templates** as generalizations of successful collaboration scripts formulated as design patterns. Design patterns provide a means of organizing information regarding a contextualized common problem and the essence of its broadly accepted solution. But, still, without the particularizations (such as tasks descriptions and resources) that are needed in order to achieve a ready-to-run CSCL script. Practitioners could use patterns as a starting point and complete or complement them according to the guidelines and information provided by the pattern.

- CSCL script **chunks** as portion of exemplars. These CSCL script chunks could be descriptions of an activity structure containing: activities, environments and resources needed. This kind of material cannot be "playable" by itself. Practitioners could assemble a set of chucks or use them to complete their templates in the creation of a full-fledged CSCL script.

- **Building blocks** or **components** are partially completed chunks. This kind of material could be activities not totally defined, just resources to support activities, etc. Fine-grained design patterns can be provided as refinable building blocks. Practitioners could reuse building blocks or components to build their own chunks and assembled exemplars of CSCL scripts.

From the levels above described we focus on design patterns that can be provided as templates or building blocks, since they have proved to be appropriate reusable components for the creation of potentially effective designs [6]. Besides, the use of patterns establishes the utilization of a common format that it is expected to assist the reutilization along the design process.

2.2 Ontology-based tools for CSCL

The contribution presented in this paper is, to the best of our knowledge, the first approach proposing to move from pattern languages to pattern ontologies in TEL. However, there are related works that use ontologies in TEL. For instance, [10] build a framework that represents the learning processes involved in collaborative educational situations, using an ontology adopting concepts from learning theories. In this approach the learning theories provide the concepts to justify and support the development of effective learning scenarios. In order to verify the viability and usefulness of their proposed ontological framework within the context of systematic design, they developed and used and intelligent authoring tool for CSCL design. This system is able to reason on ontologies in order to give suggestions that help users in the creation theory-compliant collaborative learning scenarios.

Another example is [16], which introduces Ontoolcole as an ontology of collaborative learning tools designed with the aim of supporting educations in the search of CSCL tools. This ontology allows educators to integrate external tools, offered as services by software providers, in order to support the realization of collaborative learning situation. Ontolcole has been implemented in Ontoolsearch [17], an application that supports the semantic querying of CSCL tools. Ontoolsearch has been specifically designed so that its interface is easy-to-use by educators, in terms of the query formulation as well of the analysis of the obtained results.

2.3. Problem Formulation

By adopting the designing by reusing approach, teachers are assisted in the creation of CSCL scripts based on existing designs. If these existing designs are patterns (in the form of templates or building blocks), then the reusable elements represent sound didactic ideas for CSCL scripts. Teachers only need to select the set of patterns that better fit the needs of each educational situation. However, without a guide to select patterns, teachers may end up with a set of isolated patterns that does not fit harmonically with each other. Pattern Languages (PL) were proposed to address this problem, since they are a collection of interconnected (related) patterns which enable the generation of a coherent whole [2]. Though, pattern languages interconnect patterns using natural language. Each pattern, within a pattern language, includes a description indicating which other patterns could be used to complement or complete it. This information cannot be used directly by computers and a computational representation of the interconnections between patterns is needed, if we want to flexibly implement tools supporting the design of CSCL scripts based on patterns.

To achieve the computational representation of pattern languages we propose the use of an ontology-based approach. The semantic characteristics facilitated by ontologies enable the definition of models containing the pattern information along with the relations connecting the patterns. This approach allows to computationally supporting the pattern selection process guiding educators in the design of meaningful CSCL scripts. These scripts are expected to be potentially effective since they will be based on good practices (patterns), even though the educators creating the scripts are not necessarily experts in the use of ICT and in scripted collaborative learning.

3 Pattern Ontology Approach

Moving from pattern languages to pattern ontologies enable the explicit represention of the meaning captured in patterns and their relationships as axioms, obtaining a formal semantic pattern language representation. We use an existing pattern language for the design of CSCL scripts as a starting point [7]. The resulting ontology provides us a solid base, which we plan to extend with more patterns (such as [18]) and linked reusable educational materials. As motivated in the previous section, the final goal is to provide a common framework, which mediated by computers, supports educators in the selection of a set of interrelated patterns to be applied to their own learning situation.

The selection of the ontology instead of other computational approaches such as Databases (DB) is mainly justified because of their flexibility. Although Databases are similar to ontologies regarding both: ontology axioms vs. DB schema, and ontology facts vs. DB data, there are several differences pointed out in **Table 1** [13] that motivate the use of ontologies.

As we can see in **Table 1** ontologies can deal with incomplete information while databases cannot. Furthermore, in ontologies individual elements may have more than one name, what is a critical property when establishing a common vocabulary.

This section introduces the pattern ontology built for the design of CSCL scripts and discusses how this ontology can be implemented in a tool facilitating the selection of patterns.

Table 1 Databases Vs Ontologies

Database	***Ontology***
Closed World Assumption (CWA): missing information treated as false	Open World Assumption (OWA): missing information is treated as unknown
Unique name assumption (UNA): each individual has a single, unique name	No UNA: individuals may have more than one name
Schema behaves as constraints on structure data: define legal database states	Ontology axioms behave like implications (inference rules): entail implicit information

3.1 Pattern ontology for the design of CSCL Script

The pattern ontology resulting from the semantic formalization, using the OWL language [9] and the Protégé ontology editor, of the pattern language proposed in [7] can be graphically seen in **Fig 1**. According to the pattern language, the patterns are classified into four categories: *Flow Patterns* (e.g.: JIGSAW), *Activity Patterns* (e.g: INTRODUCTORY ACTIVITY: LEARNING DESIGN AWARENESS), *Resource Patterns* (e.g.: ENRICHING THE LEARNING PROCESS), and *Roles and Common Collaborative Mechanisms Patterns* (e.g.: FACILITATOR or FREE GROUP FORMATION). Extended descriptions of the levels and the patterns can be read at [7].

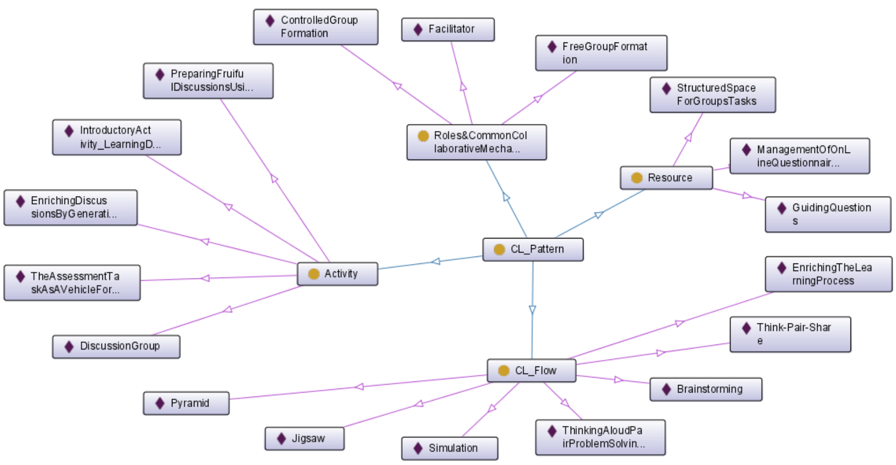

Fig. 1 Tree radial view of our Pattern Ontology (only father to child relationships are showed)

Each pattern represented in our ontology has the following information: category in which the pattern is included, a brief description about the problem it solves, the structure of the solution (especially in flow patterns, which propose sequences of activities and the group types associated to those activities), and two lists: the first with a set of patterns which *complement* it and the second with a set of patterns which *complete* it. Where *complement* means that the pattern neither modifies nor refines the current one, but together form a larger whole; while *complete* means that the initial pattern is refined with a second pattern that adds further design ideas to those already proposed by the initial pattern. Both lists are a set of recommendations the user should follow once a pattern is selected, in order to complement or complete the pattern towards a coherent set of patterns meaningful for a specific learning situation. **Table 2** shows an example of a pattern belonging to the adopted pattern language and which has been represented in the ontology. Part of this computational representation is shown in **Fig. 2.**

Table 2 Example of a pattern belonging to the adopted pattern language and integrated in the proposed pattern ontology.

Category	*Flow pattern*
Name	JIGSAW
Problem description	If groups of students face the resolution of a complex problem/task that can be easily divided into sections or independent sub-problems, an adequate collaborative learning flow may be planned.
Problem Structure	Structure the learning flow so that each student (individual or initial group) in a group (“Jigsaw Group”) studies or work around a particular sub-problem. Then, encourage the students of different groups who study the same problem meet in an “Expert Group” for exchanging ideas. These temporary focus groups become experts in the section of the problem given to them. At last, students of each “Jigsaw group” meet to contribute with its “expertise” in order to solve the whole problem.
Complement:	PYRAMID, BRAINSTORMING, TPS, SIMULATION, TAPPS, ENRICHING THE LEARNING PROCESS, INTRODUCTORY ACTIVITY: LEARNING DESIGN AWARENESS.
Complete:	PYRAMID, TPS, BRAINSTORMING, TAPPS, INTRODUCTORY ACTIVITY: LEARNING DESIGN AWARENESS, DISCUSSION GROUP, THE ASSESSMENT TASK AS A VEHICLE FOR LEARNING, FREE GROUP FORMATION, CONTROLLED GROUP FORMATION.

The process suggested to educators when using the ontology in the selection of a set of patterns to be used in the design of a CSCL script is as follows. Initially, educators are proposed to start selecting any flow pattern. Once they pick up their first pattern, the ontology recommends a set of patterns (of different categories) to complement or complete their selection. See the example in **Fig. 3,** which shows how six patterns are suggested to be used in combination with THE ASSESSMENT TASK AS A VEHICLE FOR LEARNING pattern. This process can iteratively followed (users can navigate through the visible patterns and each time they select a pattern a new set of recommended patterns will be shown) until the educators consider they have all the (interrelated) design ideas they need to create their own CSCL script.

```
<Declaration>
    <NamedIndividual IRI="#Jigsaw"/>
</Declaration>
...
<SubClassOf>
    <Class IRI="#CL_Flow"/>
    <Class IRI="#CL_Pattern"/>
</SubClassOf>
...
<ClassAssertion>
    <Class IRI="#CL_Flow"/>
    <NamedIndividual IRI="#Jigsaw"/>
</ClassAssertion>
...
<SubObjectPropertyOf>
    <ObjectProperty IRI="#complete"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
...
 <owl:Class>
   <owl:unionOf rdf:parseType="complete">
     <owl:Class rdf:about="#Jygsaw"/>
     <owl:Class rdf:about="#Pyramid"/>
   </owl:unionOf>
 </owl:Class>
```

Fig. 2 Fragment of the pattern ontology showing the definition of the JIGSAW pattern and its relationship with the PYRAMID pattern.

Thanks to the flexibility provided by the ontologies, educators can use the name under which they know a specific pattern, though this pattern may be also known under another title (i.e. PYRAMID is also known as SNOWBALL). Also, since ontologies are ruled by inferences, we can add new patterns and they will be properly integrated in the ontology by the reasoner.

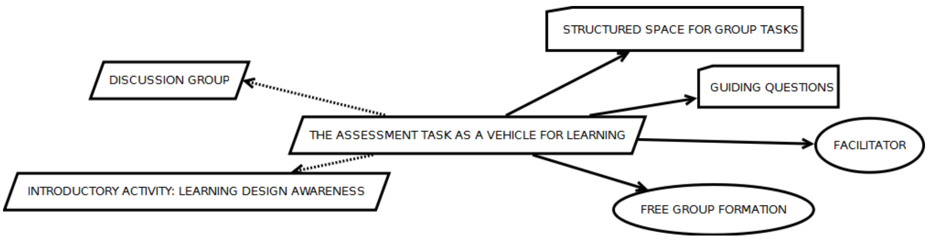

Fig. 3 Pattern selected in the center and patterns recommended in the ontology to meaningfully complement or complete it, shown, at the periphery.

3.2 Towards the implementation of the pattern ontology in the LdShake tool

The pattern ontology, created with the Protégé tool using the OWL language, is, at the end, a valid XML file, see **Fig. 2**. So, to integrate this ontology within any authoring tool or pattern repository two main modules need to be implemented: 1) A parser that reads the ontology and 2) a piece of code that transform this information into objects, so that it can be used by the logic of the tool. Besides, the logic of the tool would need to be extended so that it supports the two main types of users that would interact with the ontology.

The first type of users are the educators that select a set of interrelated patterns according to the recommendations provided by the ontology and the requisites of their particular learning situations. On the other hand, the second type of users can be practitioners, familiar or experts in CSCL scripts, who may want to use other patterns or consider relationships not represented in the ontology. The tool should be flexible in these cases and may even valorize their design decisions and extend the knowledge captured in the ontology with more patterns or new meaningful relationships between the patterns. In this way, the novice educators would be benefited with the contributed knowledge.

We are currently implementing the pattern ontology in the LdShake tool [8]. LdShake is a Web 2.0 tool for the social sharing and co-edition of learning design solutions, including design patterns. This tool is coded in PHP mainly, so for this particular case we are developing a PHP parser that reads the XML file and plan to develop a second module that visually shows the patterns recommendations to the users.

4 Preliminary Evaluation

This section provides a preliminary evaluation of the pattern ontology. First, we present two real scenarios designed by practitioners that describe two CSCL scripts based on a set of patterns. In the description of the scenarios we show how the ontology is able to represent the interconnections between the set of selected patterns so that it is meaningful for those scenarios. Second, we explain a paper prototyping experience with two other teachers, familiar with CSCL, who used the ontology for the creation of new scripts. Before creating the new scripts, we proposed the teachers to read the description of the previously mentioned two scenarios without indicating the patterns but only the main ideas that the teachers considered when planning those scenarios (number of students, main goals, how they pretended to structure the classes, how many sessions they had, the expected outcomes, etc.). Together with the scenario description, we also gave the teachers a separated sheet with the sets of patterns that "solve" or "state the solution" to both scenarios. We gave the teachers some minutes to read the scenarios and look at the solutions. Then, we asked them to make a description of a similar (imaginary or not) course and to use a paper prototype of the ontology. They must select as many patterns as they think they could use to cover their course needs. Finally, we asked the teachers to complete a questionnaire in order to understand the benefits and limitations of the proposed approach as well as to learn lessons relevant for the implementation of the ontology in pattern selection tools.

4.1 Two examples computationally represented with the pattern ontology

The first example belongs to a "Computer Architecture" course, part of the core body of knowledge in the Telecommunications Engineering curriculum in Spanish universities. The whole course is defined as a project that develops along the semester. Its objective is the design and evaluation of a computer system. The teacher defines five fictional clients and assigns one client to the students grouped in dyads. This way, in each laboratory group, different clients are being studied through the course, following the principles of the JIGSAW pattern. The Jigsaw-based structure is completed with the suggestions of the SIMULATION pattern where teacher plays the role of client. Furthermore, during the course the teacher becomes a FACILITATOR marking the milestones and presenting different assessment task to the students, as indicated by THE ASSESSMENT TASK AS A VEHICLE FOR LEARNING pattern. In each milestone, every laboratory group ("Jigsaw group" phase of the JIGSAW) holds a debate. This debate is arranged as suggested by PREPARING FRUITFUL DISCUSSIONS USSING SURVEYS and complemented with ENRICHING DISCUSSIONS BY GENERATING COGNITIVE CONFLICTS. At the end, a technical report is collaboratively produced among all dyads that have worked with the same client in each laboratory session (forming accordingly a PYRAMID). **Fig. 4** shows the graphical representation of the patterns used in the scenario and their relationships according to the proposed ontological model.

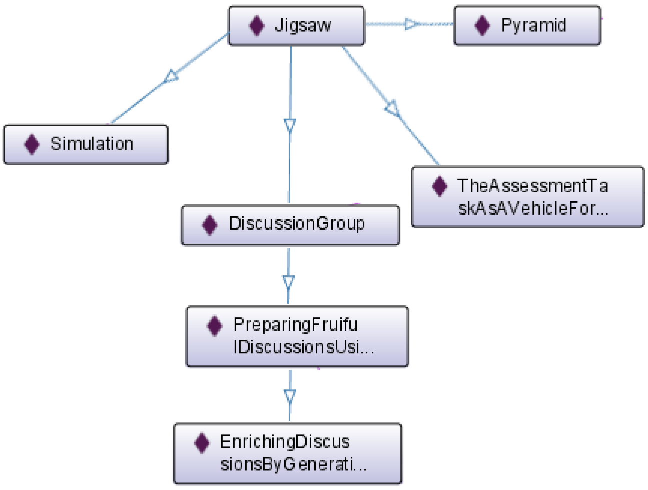

Fig. 4 Patterns and their relationships according to the pattern ontology used in the Computer Architecture CSCL script

The second example is framed in a course on "ICT resources in Education". The global objective of this course is to allow students to create didactic units in collaboration. The course was structured as follow: during the first week they were introduced to the course and the general plan following the indications of the INTRODUCTORY ACTIVITY: EXPLANING THE LEARNING DESIGN pattern. The following weeks were planned according to a two-level PYRAMID: the first level of the Pyramid is, in turn, structured in accordance with a JIGSAW. Taking into account the FREE GROUP FORMATION pattern, the students are assembled in dyads.

In the "Individual phase" of the JIGSAW every dyad studies one of the 3 main topics of the course. Then, students have to summary the main ideas of their topic and elaborate a report for assessment purposes but also as a learning task (ASSESSMENT TASK AS A VEHICLE FOR LEARNING) which pushes them to reflect on a series of questions that they should answer in the report. These questions are explicitly provided by the teacher, as suggested by GUIDING QUESTIONS.

In the "Expert Group" phase of the JIGSAW, groups of six (or seven) dyads that have worked over the same topic join in a single large group to read and discuss the reports written by their partners. In the "Jigsaw Group" phase, new groups are formed. Every group comprises a pair "expert" on each topic. In this phase, the students read and present the second report (outcome of the "expert group") and elaborate a new common report integrating the three different topics.

Finally, the second (and last) level of the PYRAMID is devoted to ENRICHING DISCUSSIONS BY GENERATING COGNITIVE CONFLICTS in a global debate where all students participate. **Fig. 5** shows the graphical representation of the patterns used in this scenario and their relationships according to the proposed ontological model.

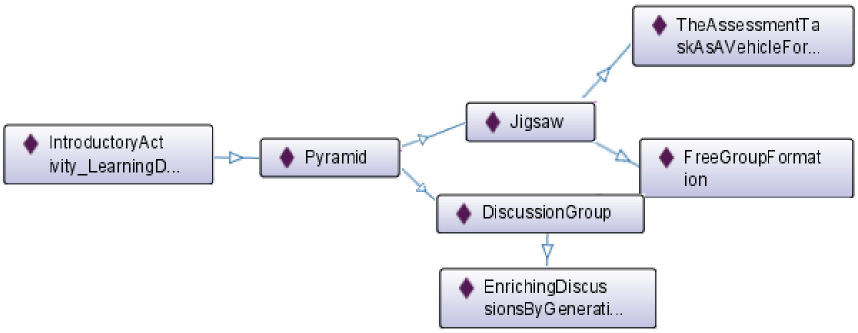

Fig. 5 Patterns and their relationships according to the pattern ontology used in the ICT resources in Education CSCL script

4.2 Paper prototyping with teachers: lessons learnt for the pattern ontology implementation

It is clear that a preliminary evaluation of the proposed pattern ontology approach is needed in order to provide insight about the proposal and its implementation in tools [19]. So, in order to be able to test the approach and obtain a first feedback we selected to use the paper prototyping method. This approach allows us to present the pattern ontology to users who could perform realistic tasks by interacting with a pattern ontology paper version that is manipulated by ourselves as "computers" [14]. The paper prototyping was carried out with two teachers following the process described above.

For the paper prototyping we draw a total of eighteen charts, the number of considered in the ontology. Examples of these charts are **Fig. 3** and **Fig. 6**. Each chart has a pattern in the center, representing the selected pattern, an all patterns related to it according to the ontology, in the periphery. We distinguish the different pattern category with a different shape: square shape for *Flow Patterns* (i.e.: JIGSAW in **Fig. 6**), square shape with a plane corner for *Resource (i.e.:* THE ASSESSMENT TASK AS A VEHICLE FOR LEARNING in **Fig. 3**), oval shape for *Roles and Common*

Collaborative Mechanisms (i.e.: FREE GROUP FORMATION in **Fig. 6**) and diamond shape for *Activity Pattern* (i.e.: DISCUSSION GROUP in **Fig. 6**). Furthermore, we draw of the diverse types of relationships between patterns with different contour lines. We distinguish three different ones: solid lines for *complete* relations (i.e.: JIGSAW to DISCUSSION GROUP, bottom-right in **Fig. 6**); dotted lines for *complement* relations (i.e.: JIGSAW to SIMULATION, top-left in **Fig. 6**); and dashed-dotted lines to patterns that could be used for both, complement and/or complete (i.e.: JIGSAW to PYRAMID, center-top in **Fig. 6**). We included a legend describing all this information.

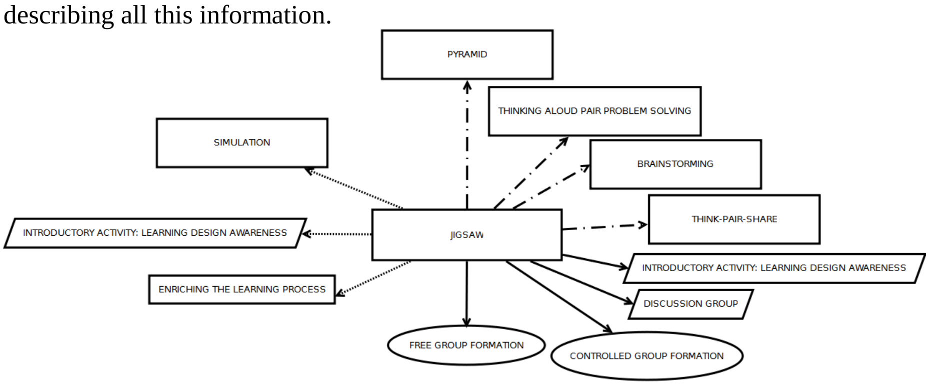

Fig. 6 Paper prototyping example: JIGSAW flow pattern chart

During the evaluation we acted as the computer and any time the teachers selected a pattern we marked it as selected and showed teachers another chart expanding the relations of this pattern with other patterns. Together with these charts we also gave teachers tables with the descriptions of the patterns, so that they could consult them if needed. See **Table 2** to see a compacted version of the table for the JIGSAW pattern.

The analysis of the data collected in the questionnaires completed by the two teachers lead to the conclusions shown in **Table 3.** As a positive feedback we obtained that with our approach we foment the creativity of the teachers. According to the teachers' opinion, the ontology suggests relationships between patterns that made them to consider patterns in their designs that otherwise they would not have included. Moreover, they were satisfied with the resulting design and said that the designed script was well structured and the considered strategies seem to have potential to enhance the collaborative learning of their chosen educational situation. Aspects for improvement suggested by the teachers were around the amount of information presented to the users (especially at the beginning). Besides, the teachers needed to read a lot while starting to be aware about the patterns and relate their already known patterns to the patterns suggested by the system. In some cases, it seemed that they were familiar to the design ideas captured in the patterns but the system was using different titles for the patterns. Finally, the teachers also pointed out that a more clear legend would be required to better explain shapes and line counters (described in the above section 4.2).

Table 3 Main positives and negatives aspects reported by the teachers

Positive Aspects	Aspects for improvement
• Variety of patterns and relations foments the creativity. • Relations between patterns lead to the selection of extra design ideas that enrich the collaborative learning activities. • The differences between pattern categories (at flow level, activity and resource) to guide the structure of the designs.	• Too much information at the beginning. • Visualization comprehension: arrows and figures meanings (clearer legend required). • Though the ontology guides the selection of the patterns, it is necessary to read the descriptions of the recommended patterns.

The lessons learnt from the paper prototyping experience with teachers are being currently taken into account in the implementation of the ontology in LdShake. In particular, the amount of information showed in screen when users start selecting patterns will only focus on flow patterns, then the user will be able to continue selecting the patterns that can be meaningfully combined with their selected flow pattern, complementing or completing it. Besides, we are designing an accurate legend that describes the different figures representing the different kind of patterns (learning flow, activities, resources and roles). Finally, summaries or graphical representations of the patterns will be visualized as small tooltips that support users in the understanding of the patterns' solutions when they pass the mouse over the patterns.

5 Conclusions and future work

This paper contributes to the design-by-reuse approaches aiming at assisting teachers in the creation of potentially effective CSCL scripts. In particular, it focuses on patterns as the design elements to be reused. Pattern languages collect a number of patterns and document the relationships between them. However, these relationships are expressed in natural language and cannot be interpreted by software tools. In this paper, we propose to move from pattern languages to pattern ontologies in order to computationally represent the knowledge captured in an existing pattern language for the design of CSCL scripts. This pattern ontology can be implemented in tools so as to enable teachers, typically not experts in CSCL or ICT, to creatively select a coherent set of interrelated patterns that suit the requisites of their particular learning situation. Moreover, if implemented in tools practitioners, expert in CSCL, could also flexibly use the ontology and extend its knowledge with more patterns or new meaningful relationships. The preliminary evaluation presented in this paper shows that the support provided by the proposed pattern ontology is encouraging, but also that its implementation in tools, guiding the selection of patterns, is not trivial. The main challenges deals with the visualization of the interconnected patterns and their descriptions. We are currently working on facing these challenges and implementing the pattern ontology in the LdShake tool, supporting the sharing and co-authoring of design solutions. Moreover, we are also extending the ontology with the integration of assessment patterns [18] and plan to incorporate more patterns and link other educational materials in the short future.

References

1. Dillenbourg, P.: Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In P. Kirschner (ed.) Three worlds of CSCL. Can we support CSCL?, pp. 61--91. Open Universiteit Nederland, Heerlen (2002)
2. Goodyear P.; Retalis, S. (eds.). E-learning, design patterns and pattern languages, Sense Publishers. The Netherlands (2010)
3. Griffiths, D.; Blat, J., The role of teachers in editing and authoring units of learning using IMS Learning Design, Advanced Technology for Learning, Advanced Technology for Learning, vol. 2 (4), pp. 208—218 (2005)
4. Griffiths, D.: From Primitives to Patterns: a Discussion Paper., UNFOLD project. Available at http://digitalcommons.bolton.ac.uk/cgi/viewcontent.cgi?article=1001&context=iec_jo urnals (2005)
5. Hernández-Leo, D.; Harrer, A.; Dodero, J.M.; Asensio-Pérez, J.I.; Burgos, D. A Framework for the Conceptualization of Approaches to "Create-by-Reuse" of Learning Design Solution. Journal of Universal Computer Science; vol. 13(7): pp. 991--1001. (2007)
6. Hernández-Leo, D.; Jorrín-Abellán, I.M.; Villasclaras, E.D.; Asensio-Pérez, J.I.; Dimitriadis, Y. A multicase study for the evaluation of a collaborative learning pattern-based visual design approach, Journal of Visual Languages and Computing, vol. 21(6), 2010, p. 313–331.
7. Hernández, D.; Asensio, J.I.; Dimitriadis, Y.; & Villasclaras, E.D. Pattern languages for generating CSCL scripts: from a conceptual model to the design of a real situation. In: Goodyear P.; Retalis, S. (eds.). E-learning, design patterns and pattern languages, Sense Publishers. p. 49-64 (2010)
8. Hernández-Leo, D.; Carralero, M.A.; Chacón, J.; Pérez-Sanagustín, M.; Carrió, M.; Blat, J.; Sharing and co-editing learning design solutions. In: Proceedings of the European LAMS Conference. Oxford: pp. 20--23. (2010)
9. Horridge, M., Knublauch, H., Rector A., Stevens, R., Wroe, C. A practical guide to building OWL ontologies using the protégé-OWL plugin and CO-ODE tools edition 1.0. Technical Report, University Of Manchester, url: http://www.co-ode.org/resources/ (last visited January 2011)
10. Isotani, S.; Mizoguchi, R.; Inaba, A.; Ikeda, M., The foundations of a theory-aware authoring tool for CSCL design, Computers & Education, vol 54 (4), pp. 809—834 (2010)
11. Koschmann, T.D., CSCL, theory and practice of an emerging paradigm. In T. Koschmann (Ed.). New Jersey: Lawrence Erlbaum. (1996)
12. LN4LD: Example IMS-LD packages, available at http://dspace.learningnetworks.org/handle/1820/257, Open University of The Netherlands. (2005)
13. Motik, B.; Horrocks, I.; Sattler, U. Bridging the gap between OWL and relational databases. Web Semantics: Science, Services and Agents on the World Wide Web. Vol 7 (2), pp. 74—89 (2009)
14. Snyder, C. Paper prototyping: The fast and easy way to design and refine user interfaces. In Morgan Kaufmann Pub. (2003)

15. Stahl, G.; Koschmann, T.; Suthers, D., Computer-supported collaborative learning: An historical perspective, Cambridge handbook of the learning sciences (2006)
16. Vega-Gorgojo, G.; Bote-Lorenzo, M.L.; Gómez-Sánchez, E.; Asensio-Pérez, J.I.; Dimitriadis, Y.A.; Jorrín-Abellán, I.M. Ontoolcole: Supporting educators in the semantic search of CSCL tools, Journal of Universal Computer Science, vol. 14 (1), pp. 27—58 (2008)
17. Vega-Gorgojo, G.; Bote-Lorenzo, M.L.; Asensio-Pérez, J.I.; Gómez-Sánchez, E.; Dimitriadis, Y.A.; Jorrín-Abellán, I.M. Semantic search of tools for collaborative learning with the Ontoolsearch system. Computers & Education vol 54 (4) pp. 835—848 (2010)
18. Villasclaras-Fernández, E.D.; Hernández-Leo, D.; Asensio-Pérez, J.I.; Dimitriadis, Y. Incorporating assessment in a pattern-based design process for CSCL scripts. Computers in Human Behavior, vol. 25 (l5) pp. 1028—1039 (2009)
19. Zelkowitz, M.V.; Wallace, D. Experimental validation in software engineering, Information and Software Technology, vol 39 (11) pp. 735--743 (1997)