Learning with Patterns: an Effective Way to Implement Computer Supported Pervasive Learning

Nelson Baloián, José A. Pino, Gabriel Peña
Department of Computer Science, Universidad de Chile
Blanco Encalada 2120, Santiago, Chile
{nbaloian, jpinodcc.uchile.cl}

Gustavo Zurita
Management and Information System Department,
University of Chile, Santiago de Chile
gzurita@fen.uchile.cl

Abstract— Nowadays computers and networks are everywhere. They take different shapes varying from big electronic whiteboards to small mobile devices. Perhaps all of them have been already used to support some kind of learning activity. Although the literature claims they have been very successful supporting certain, often isolated type of learning activity they can seldom be combined with other systems to enable more consistent and comprehensive support. Researchers have already noted the necessity to develop learning methodologies and systems which can support learners along a continuous flow of different learning settings. In this paper we present a learning system and a methodology based on the use of patterns. Students learn about design and architectural patterns by finding instances of them on the field, or by recognizing new patterns unknown to them so far. This work illustrates how computer technology can support the students in a continuous way without technological or methodological disruptions.

Keywords-component: learning with patters, patterns in design and architecture, collaborative learning, mobile computing

I. INTRODUCTION

Computers have been used to support learning in various ways since they were introduced to the market. At the beginning they were used exclusively to support individual learning [1]. The approach was to replace the teacher with an intelligent, tireless and adaptive tutor. The scenario where typically this learning activity took place was a computer lab. With the development of networked computing the approach for supporting collaborative learning processes gained popularity among researchers. Computer Supported Learning activities were now possible at home and in the classroom [2]. More recently, the widespread availability of mobile devices has led to an increased interest in the use of mobile computing to support formal and informal learning, as well as individual and collaborative learning [3]. Nowadays, computers are being used to support a variety of learning modes (individual, collaborative, synchronous, asynchronous) in different scenarios and settings (in-classroom, outdoors, formal and informal learning) and the use of advanced educational technology has increased dramatically in the last ten years [4]. Due to the new mobile technologies, which translate in better and more affordable costs for devices and services, and an access to digital content almost everywhere and every time, learners can experience new learning activities in a variety of situations not only related to school settings, but also work, leisure, etc. This is known as pervasive learning. These technologies give new opportunities for designing innovative educational activities that can be carried on at any place. The design of such activities is especially challenging when it comes to conceptualizing how pervasive technologies can support collaborative knowledge building [5].

However, the expected goal that new technologies will transform learning practices has not yet been fully realized, especially if we take into account what has been going on in computer supported collaborative learning [6]. According to many researchers one of the main problems is that the various systems developed for various learning scenarios mainly address a specific learning situation, thus there are too many different systems supporting the different learning situations [6],[7]. Thus a continuous computer support across different learning activities is difficult to implement with most available computer-supported learning tools, because data from the various application might not be compatible and students and teachers have to learn the usage of various systems with different human-computer interaction paradigms. Although seen in the past few years we have some technical developments aimed to integrate various different computer worlds, enabling a "continuous computing" across mobile and non-mobile devices, an important research question remains: Which type of learning practices are able to make full use and get most benefit from pervasive technologies? In the context of trying to get an answer for this question it is also important to find out which technologies can be used to support these new ways of learning about different educational subject matters, and how can these technologies support learners working individually, collectively, inside and outside the classroom without technological disruptions.

In a previous work, the authors proposed the use of mobile devices to support learning of architectural patterns [7], [8] as a learning activity with the aim of integrating formal (or in-classroom) with informal (or outside-classroom) learning. For this, an application called Pattern Collector was developed (see section II for a deeper description) running on mobile devices. Although it was designed to be used inside and outside the classroom, it did not provide the technological framework allowing a swift
II. LEARNING WITH PATTERNS

A. The role of patterns in learning and design

Patterns play a significant role in learning. Based on findings from the learning psychology we can affirm that (at least part of) the human learning can be explained by the fact that learner discover, register and later apply patterns [9], [10], [11]. This cognitive process “involves actively creating linkages among concepts, skill elements, people, and experiences” [12]. For the individual learner, the learning process involves "making meaning" by establishing and re-working patterns, relationships, and connections" [12]. New biological research reveals that "connection-making" is the core of both mental activity and brain development [12].

Recognizing affective, social and communicative patterns and organizing perceptions into meaningful categories is one of the most fundamental aspects in even the earliest learning experiences [7]. Developmental psychology shows evidence that early patterns of affect signaling and culturally mediated types of mother-child interaction lead to the core capacities of pattern-recognition, joint attention and intention reading. As preconditions for cognitive and language development these capacities then lead to the ability to speak and think logically and creatively [18], representing a fundamental aspect of various learning activities.

Patterns are recurring models, often are they presented as solutions for recurring problems. The concept is based on a distinction between a set of phenomena and the underlying model that can be inferred by analyzing its constitutive features [7]. This is known as “pattern recognition” which has been used by cognitive psychology as well as artificial intelligence researchers in order to implement a similar approach for machine learning [13]. Natural sciences, mathematics and arts also work with patterns. The exact use of the term however varies from discipline to discipline. The first formalization of pattern description and their compilation into networks of “pattern languages” was proposed by Alexander et al. [14]. A pattern consists of a set of components including the name of the pattern, description of the problem it solves, the solution to this problem, an example and the relations it has to other patterns. This approach has been adopted by many disciplines like architecture, software development [15], interaction design [16] and pedagogy [17].

B. Learning with patterns in design and architecture

As we have seen, there is a lot of evidence that patterns play an important role in learning. However they have seldom been used to support development of cognition and skills apart from the mathematics. In [14] the authors present an interesting experience about collaborative learning of early algebra based on patterns. In [7] the authors of this paper present a system that supports the collaborative searching and documenting of instances of a certain pattern on the field for learning purposes.

However, learning with patterns in design and architecture can involve more activities than just collecting evidence on the field. It may start in the classroom with the teacher giving an introduction into the design pattern approach, the structure of patterns and the creation of pattern languages within the respective field, in this case architecture. The teacher then gives a topic like neo-gothic architecture and asks the students to collect examples of this type of architecture in the city. Students then explore their neighborhoods, taking pictures of buildings having elements that characterize this type of architecture like towers, entrances, etc. In the field or at home they reflect upon why they find a certain building a suitable example, which are the elements is has that makes it a good representing of this type of architecture, and document what they found within the given categories. In the field or at home they may exchange and debate with peers for or against the patterns they want to propose and the examples they had found. Each pattern proposition is reviewed by two peers. Back in the classroom, they present their patterns on the whiteboard, and, moderated by the teacher; they evaluate their propositions and discuss the hierarchy, its distinctive levels, and the relations between the patterns they intend to work with in order to create their own pattern language. Design and architecture students would then go on to apply their own patterns by building models that represent ideal representations of these patterns and pattern languages for a specific context. In the end, they will not only have learned how to extract and work with patterns, but will have also addressed critical issues and trade-offs in designing entrances and exits. In an advanced scenario, students collect examples of a certain architecture solution that is repeated in many building and reflect upon whether they constitute a pattern.

III. APPROACH AND SYSTEM REQUIREMENTS

As we see from the previous chapter, the learning scenario we want to support requires computer support in various scenarios and settings. First, the teacher might use computer support to present examples of neo-gothic architecture and draw the patterns that repeat itself in this type of architecture. She distributes then the patterns among the students who store them in their computers in order to study and analyze them. Then the students download the patterns in their mobile devices and go out with them in the city in order to look at them when trying to find some instances. They also use their mobile devices to record the instances founded relating them to the corresponding pattern. For example, they may find some example of typical neo-gothic towers which may correspond to the conic pattern (towers which are shaped like a cone) or to the rectangular pattern (towers shaped like a parallelepiped) which they photograph, record information about their location, history and highlight the characteristics that makes them fit the pattern. They do this working collaboratively on the field. Back at home they may work individually or collaboratively to prepare a presentation with the data of their finding. Normally they will do this work on a desktop computer, so they will have to trespass the data from the mobile device to the PC. Back in the classroom they will present their findings using the material prepared at home and they will have to distribute it to the rest of the students.
A. Requiring swift interaction among different platforms

As we have seen, this learning activity requires intensive data exchange and constant switches from desktop to mobile platform and back. It also requires good support for switching between collaborative to individual work and back, which means switching form synchronous to asynchronous working modes. Although we can find ways to use off-the-shelf software to do this, it has been noted that it might be a very cumbersome process which drive the focus of attention of the students away from the important and meaningful tasks for learning [19].

Some authors have already noted the need to have platforms supporting and effective and easy exchange of data between platforms, since pervasive computing is becoming more and more popular. It is interesting to note that most contributions have been developed to support learning activities. This is certainly due to the disserement of the potentials of mobile computing to support learning and the necessity to couple learning activities inside and outside the classroom. An interesting example is MOOsburg [20] developed to support students to develop their awareness about the environment by assessing the biological state of the water streams around the community of Blacksburg. MOOsburg is a community oriented collaborative environment that models the town providing an interactive map with shared content such as chat spaces, message boards, and so on. The system was extended in order to allow access from mobile devices to the virtual world.

SQLspaces [21] was developed to support the implementation of distributed learning environments with components programmed in different languages. It implements a blackboard architecture based on Tuple Spaces. A tuple space is a server which implements a repository where data vectors can be stored and retrieved by different client agents implemented on different platforms. The uniform access to the server is ensured by providing an interface for each programming language. Currently there are interfaces for Java, Prolog and C#. The Java interface is meant to enable desktop applications to have access to the repository while the C# interface is for application running on mobile devices. The prolog interface enables applications written in this language to access the data and make intelligent processing, like checking the semantic consistence of some contributions. There are two interesting characteristic of this tool that make it very suitable for the learning with patterns scenario: first, since the data repository is implemented with the help of a database management system (SQL) the data stored is persistent and can be easily accessed and complemented by latecomers, even when the working session is over, thus serving as an effective tool to implement asynchronous as well as synchronous work. Second it implements the “loosely coupled” and stateless server approach which means that there is no direct communication between the various application interacting trough it and that application do not have to maintain a complicated conversation with the server but they rather interact by many simple, atomic request-response operations. This makes the development of distributed applications easier since there are less possibilities of introducing errors.

Another framework implemented to support the development multi-platform applications is the SharedObjects platform [22]. This platform was implemented to support the development of peer-to-peer applications running on a mobile ad-hoc network (MANET) provided by the networking capabilities of the mobile devices. This means that all users will have the same program running on their handhelds and there is no central service. The program using this framework recognizes the presence of other participants and establishes a secure communication with them in order to transfer data for synchronizing the applications. This is done via multicasting, peer discovery and synchronization via point-to-point data communication. The synchronization mechanism is based on the shared object paradigm, which means that any object declared as shared in the application will distribute its changes to all other applications participating in the session. The data synchronization protocol is based on XML, using a simplified SOAP representation of objects. This allows applications written in different languages to exchange data without problems. Currently there are two versions of the platform implemented in C# and Java which can interact and synchronize data between them.

Based on the requirements described in the first paragraph of this section we decided to combine and use both SQLspaces and Shared Objects platforms to implement synchronization. Shared Object will support the work of the students on the field, where they need to work synchronized without the presence of a reachable central server or repository. At home or in the classroom SQL Spaces will help them to transfer data from their mobile devices to desktop computers and work with them in an asynchronous way. Figure 1 shows the implemented synchronization schema.

![Figure 1](image_314_to_529)

Figure 1. the figure shows the interactions between students, teacher and the repository in different scenarios using SQLSpaces and/or SharedObjects as needed

B. Functionality requirements

In order to allow support learning with patterns as described in section II the system must provide at least the following functionalities:

- **Creating Patterns**: the teacher creates a pattern during the lecture defining its components. This pattern will then be used by the students to find instances on the field. Depending on the assignment, students may also create patterns in order to document findings which
following a certain pattern. For example, the teacher can ask the students to find patterns in decoration elements that are repeated in gothic buildings.

- **Duplicating patterns:** Students record their findings and create new patterns. For this, they take the original template defined by the teacher and create “clones” of them which will inherit the structure. They can add or modify descriptions of its components. A user can create clones of patterns by clicking on the respective icon when browsing an already created pattern.

- **Instantiating patterns:** the students create instantiations of the pattern when they find a certain element that they think it corresponds to the pattern giving by the teacher. Instantiations consist of photographs or handmade sketches of a certain object found which complies with the pattern definition

- **Linking patterns:** Finding links among patterns is an important cognitive task. By making this, a “visual pattern language” is created enriching the understanding of the pattern. In the architecture example, students may find that certain patterns are found very frequently together and they can try to find an explanation for this fact.

- **Sharing patterns and their instantiations:** A key requirement for the system is that the information generated can be shared with others under the following circumstances: a) Teacher can distribute an initial pattern to the students so that they can search for instantiations and/or generate new ones based on them; b) students work synchronously on the field collaboratively creating pattern instances or new patterns; c) Students exchange their patterns and/or pattern instantiations and clones with the whole class and/or the teacher

- **Comparing patterns:** Comparing patterns in order to highlight their differences and similarities is also an important part of the process of learning with patterns.

**C. Interaction requirements**

The learning activities supported by the system will take place in different locations, using different platforms. However, it is important to have a single interaction paradigm for all interfaces running on different platforms in order to easy the task of the users having to switch between different devices.

For various reasons we adopted an interaction paradigm oriented to pen-based sketching and gesturing. There are several works supporting the thesis that sketching and gesturing with pen-based systems are natural modes for design-task-oriented interaction. In [23] it is noted that a sketch is a quick way of making designs that a) facilitate the creator’s idea generation process, b) stimulate communication of ideas with others, and c) stimulate the use of early ideas, thanks to the accessibility and interpretation they provide. It has been shown that the participation of various persons in the elaboration of a sketch using computer support improves the creativity of the group. We can also mention that gesture-based interaction minimizes the number of widgets needed on a workspace and avoids the need of a virtual keyboard, thus maximizing the space available for entering content, which in case of handheld devices is usually reduced. The content consists of exclusively free handwriting inputs. Although free handwritten text may take more space than typed text, it allows a flexible combination of sketching and writing. Gestures can combine in one action a command and its arguments, thus avoiding errors. As observed in [5], collaborative design based on gestures, sketches and an interface allowing pen-based interaction enhances the design process in a natural and harmonious way, enabling the sharing and exchange of design information in order to improve efficiency.

**IV. THE SYSTEM IMPLEMENTATION**

Instead of building a system from scratches we improved some existing ones assembling them together. In the core of the system we have the Pattern Collector which allows the creation of patterns following the schema proposed by Alexander. Its interface was improved by incorporating the results of a usability test already conducted with this tool. As teacher and students can perform the same tasks on patterns there are no different roles implemented. From the functional point of view, we basically supply the same application for all users and for all platforms. As working with patterns is supposed to take place both inside and outside the classroom, it is expected that the application will be used with a number of different touch-sensitive devices like PDAs, Tablet-PC and electronic whiteboards. The application adapts the size of the user interface to the available screen; however, most of the functions (including gestures) remain the same in order to have a consistent interaction model.

**Figure 2.** At the left the pattern workspace with an empty pattern. The dotted line shows the gesture for the creation. At the left, the selected pattern with the interaction icons (zoomed). At the top various icons are displayed. Thy trigger session management and collaborative work functionalities, for showing alternative views for the pattern set, and for providing group awareness.

**A. Pattern creation**

As we said, the interface design incorporates fee-hand sketching and gestures. In order to create a pattern, a user draws two lines of a rectangle in one stroke. This gesture is recognized by the system and a complete rectangle will appear on that place representing the new (empty) pattern on the patterns’ workspace (see figure 2 left). When the pattern is selected with a single click, 4 small icons appear around the pattern (figure 2 right). Clicking on the unlocked lock will protect the pattern from being changed by other users. Clicking on the “+” icon will clone the pattern. A clone pattern is editable even if the original pattern was locked. Clicking on the icon with a horizontal arrow over a document will create an instance of the pattern.
This will open a new workspace where the elements of the instantiation can be created and placed (see figure 3 left). In order to remain consistent with the requirement of an easy to use interface based on gestures and sketching the way to specify the content of a component are: a) hand-written text b) sketching, c) images (or icons) from a file, d) typed text entered with the virtual keyboard or the built-in text recognizer of PDA or Tablet-PC. By clicking on the icon with the vertical arrow pointing down the system enters the workspace where the content for the various components of the pattern (name, context, solution, forces, evidence, image, notes) can be provided, where a list with the names of the components is displayed, as seen on figure 3, right.

Figure 3. At the left the principal workspace for inputing elements of the instantiation, here with an image and handwritten text. At the right the workspace for defining the content of the pattern components. Here the component Name is being defined. In both workspace a new icon with a vertical arrows appears. This takes the user back to the pattern workspace.

In order to provide content for each component the user has to first enter the content in the workspace for a component in the same way as described for adding content to a pattern instance. For ending the addition of the component a horizontal line should be drawn at the bottom. Before starting the addition of the next component the name of the already created should be selected from the list. For this, the user has to draw a stroke starting from the name of the component in the list to the rectangle highlighted at the bottom right corner of the component area. After this, the stroke will disappear and the name of the component will be displayed in this area. The content of the name and the image will be combined and displayed as the icon for the pattern (see figure 4).

**B. Linking patterns and alternative pattern views**

It is important to provide different views of the pattern set the student is working on, especially when relations have been defined in order to keep the overview of the whole structure and help them compare the various patterns and instantiations they may have created. Linking patterns is done at the principal workspace. In order to create a link between patterns the user has to draw a stroke starting from one pattern and ending inside the area of another. This gesture is recognized by the system and the stroke is replaced by a straight arrow (as shown in figure 4, left).

As more patterns and pattern instances are created and linked together, the overview of the whole structure might be difficult to maintain. Also the navigation can get quite complicated. In these cases a tree-like view of the whole structure might be helpful (see figure 4 right).

This view will adjust the size of the pattern icons in order to make them all fit the screen size. Although the initial view the system offers contains all the pattern icons, it can be zoomed in or out by making a gesture of a diagonal line. It is also possible to have a view of a single pattern icon showing its instances, incoming and outgoing links by double clicking on the corresponding icon of the tree-view. Another important view is the one that allows users to compare the various patterns in an easy and flexible way. Figure 5 shows the view provided by the system. Patterns and their components are shown as a table. The table can be traversed horizontally or vertically by doing the corresponding gesture as shown by the dotted line.
V. CONCLUSIONS

There are still many amazing challenges we face in computer supported learning. As technology evolved researchers have experimented how to use the new computer devices and services to support learning achieving various levels of success. Nowadays computer technology is getting more and more ubiquitous, mobile, and integrated. This is quite an interesting opportunity to experiment with learning integrated approaches to computer supported learning, since the technology is now available. Moreover, we see in this approach an answer to some of the problems Computer Supported Learning (CSL)-researchers have noted that are still unresolved, especially those related to systems supporting a particular learning setting successfully but failing to be used on a day by day basis. We have seen in the pattern-based learning approach an opportunity to implement an integral learning environment because it comprises learning activities in various settings and modes. The described system will be formally tested in the near future with students from the second year of the Architecture and Design Faculty of the University of Chile, and we have some reasons to expect encouraging results: we used the results of the formal testing of Pattern Collector to improve the version we are presenting here and we are using a single human-computer interaction paradigm design for all working situations.

ACKNOWLEDGMENT

This paper was funded by the Latin American Collaborative Research Federation (LACR). Project number R1209LAC002.

REFERENCES