PACT: A Pattern-Annotated Course Tool

Andy Carle, John Canny, and Michael Clancy
Berkeley Institute of Design & Computer Science Division
University of California
Berkeley, CA, United States
{acarle, jfc, clancy}@cs.berkeley.edu

ABSTRACT: We present PACT: a visual editor for annotating sequences of learning objects with references to pedagogical patterns. Previous efforts to reform pedagogy via design patterns have focused on two approaches, each of which has severe disadvantages: the exchange of pedagogical pattern descriptions and the creation of learning environments informed by them. The key innovation of PACT is its focus on pattern-annotated courses that connect these methods. We discuss the goals driving the design along with the features that comprise this tool. Finally, we identify opportunities for community involvement and future work.

Introduction

Over the past century, researchers in pedagogy and human learning have continuously updated our knowledge of how to educate effectively. Unfortunately, the scenery in most of today's tertiary-level classrooms looks much the same as it did 100 years ago: a vast sea of students focusing persistently on a stream of information coming from a single source – the mouth of the instructor. Environments like these stand in stark contrast against the modern recommendation of *learner-centered* milieus, where the focus of attention is the student and the emphasis is on learning as an active process. Learner-centered approaches include cooperative learning, problem-based learning, peer instruction, and inquiry-based learning   (Bransford, Brown, & Cocking, R.R, National Research Council (U.S.), ; Brown & Campione, 1994; Collins, Brown, & Holum, 1991; Dillenbourg, 1999).

Instructors must master the creation of a wide variety of *active content* to successfully craft a learner-centered environment. Such content engages the learner in a spectrum of learning activities that complement one another and carefully develop the student's understanding. The content comprises various types of learning objects with specific purposes. For example, objects may require students to answer a question before joining a discussion (gated collaboration), facilitate exploration in a safe environment (sandbox), answer questions with immediate feedback (scripted assessment), steer students toward a learning goal (guided discovery), or help them reflect on their learning process (meta-cognitive scaffolds).

Advanced learning systems (e.g. WISE and UC-WISE (Clancy, Titterton, Ryan, Slotta, & Linn, 2003; Linn, 2005)) and new classroom-friendly platforms (e.g. tablet computers) help manage this complexity and have shown remarkable improvements in learning when used *with appropriate content.* But improvements after putting such a system in place are not automatic, and it is the content rather than the platform which is the key. An advanced learning management system provides the tools – the learning objects and a mechanism for their presentation – for learner-centered courses. However, these tools must be coordinated in a skillful way to achieve an optimal result. If the instructor putting the pieces together does not know the meaning and purpose of all these objects, and how they are used together, then the end result is likely to be no better than an implementation using more conventional methods.

In our experience, the majority of instructors in today's universities still employ a lecture-based format for most of their classes. The problem is, most tertiary-level instructors don't know how (or are unwilling) to organize and deliver a learner-centered course. In fact, most have never even participated as a student in a learner-centered course. The flow of activities and role of the instructor are alien to potential adopters of these systems and techniques. To further exacerbate the problem, the few instructors who are comfortable with learner-centered course design typically find it difficult to have a lasting impact on curriculum. The standard "coordination" between instructors at many institutions ends at the handoff of lecture notes, exercises, and exams – artifacts that are

clearly insufficient to express the rationale behind elements of a learner-centered course. When an instructor is able to make long-term changes to a course, the subtleties of her decisions are often lost in the first curriculum redesign that she is not involved in. Finally, even in a world where curricula never change and courses remain static, the expert course designer has not automatically become a peer instructor. Other instructors designing their own courses will have trouble emulating the "perfect" course without seeing *why* the designer made the choices she did.

Pedagogical design patterns (H. Sharp, Manns, & Eckstein, 2000) provide a flexible scaffold for instructors to use in creating well structured learner-centered courses and a suitable language for expressing why decisions were made without becoming mired in the details of a specific implementation or context. Previous efforts in the pedagogical patterns movement have focused on the exchange of pedagogical patterns or on the creation of learning environments informed by them. In the former case, instructors often have difficulty linking the principles to their own courses. In the latter case, it is difficult for instructors to uncover the principles that have informed the learning environment design. In our assessment, and that of Fincher and Utting (Fincher & Utting, 2002), this divide has been detrimental to the wide-spread adoption of pedagogical patterns.

In this paper we present a tool designed to bridge the gap between these two approaches: PACT: A Pattern-Annotated Course Tool. The key innovation of PACT is its focus on pattern-annotated courses that connect the abstract ideals of pedagogical patterns to learning environments that exemplify these principles in concrete instances. Instructors can see learner-centered courses designed by experts along with the patterns that have guided their design. It provides a tangible representation from which instructors can proceed to deeper understanding of the theory and application of patterns and principles. And it serves as a practical tool to expedite the organization and re-use of course content.

Related Work on Patterns

Design patterns originated in the work of the architect Christopher Alexander, who explains them as follows: "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" (Alexander, Ishikawa, & Silverstein, 1977). Alexander's ideas revolve around his conception that knowledge about beautiful and useful architecture has been lost by ordinary people and the job ceded to professionals. He then went on to describe, in the book *A Pattern Language* (Alexander et al., 1977) a process by which ordinary people could recapture this aesthetic and practical sense by seeing how experts thought about common problems.

The Pedagogical Patterns Project (H. Sharp et al., 2000) was begun as an offshoot of work by Mary Lynn Manns, Helen Sharp, Maximo Prieto, and Phil McLaughlin, inspired by a paper by Susan Lilly (Lilly, 1996). The concept was to search out expert teaching practice and capture it in a pattern format to foster a common vocabulary for pedagogical concerns, aids for documentation and learning, and encouragement for reuse of pedagogical techniques. Early focus was on teaching object-oriented concepts; subsequent work by Joe Bergin extended the focus to other areas of CS instruction (J. Bergin, Fourteen Pedagogical Patterns).

An example of a pedagogical pattern is one named "Spiral" (J. Bergin, Fourteen Pedagogical Patterns) which provides advice about topic coverage in a course where there are a large number of concepts that must be mastered together. The pattern form includes the following: 1. A description of the problem. 2. The "forces" governing the application of the pattern. 3. A description of the solution. 4. Advice on how to implement the pattern.

The intended result of this is that a novice teacher, or someone teaching in a new field or a new way, can judge for herself whether the pattern "fits." The advice given by the pattern is also general enough that the user can tailor it to current needs.

Sharp et al. (H. Sharp, Manns, & Eckstein, 2003) comment on the progress of the Pedagogical Patterns Project: "During the life of the project, we have learned a lot about patterns and their application to pedagogy, and the work is still growing and changing. … For people outside the project who don't know the material as well as members of the project, it can be quite daunting to pick up a [pattern] language and begin to use the patterns it contains." Likewise, Fincher and Utting (Fincher & Utting, 2002) advocate that the collection of additional patterns and pattern instances is the major obstacle between the pedagogical patterns movement and wider adoption. These

sentiments suggest that collaboration between CS content specialists and pedagogical specialists is relatively rare. We seek to bolster this partnership by supporting collaboration and distribution of work among a larger community.

Design Goals and Features

To address past deficiencies in pedagogical reform projects based on design patterns we have developed PACT: A Pattern-Annotated Course Tool. PACT is a learner-centered application (with instructors being the learners) shaped at the highest level by principles from the learning sciences: making thinking visible, scaffolding, and a progression from concrete to abstract. This visual editor allows direct manipulation of learning objects for basic course design, but overlays the course sequence with ***pedagogical pattern annotations***. This connection between patterns and real-world learning objects affords a perspective not present in traditional pattern representations. This perspective is utilized by the features described below to pursue these design goals:

- Capture instructional expertise in a useful way for others.
- Encourage experimentation with, and adoption of, best pedagogical practices
- Guide instructors in framing course goals and refining courses to meet them
- Offer a cohesive framework applicable across a range of instructional settings

We created PACT to encourage exploration and learning about pedagogical patterns and best practices in course design. Instructors live busy lives and want to make the best use of what time they do spend thinking about teaching. As such, we have designed PACT to be the path of least resistance – a fun, visual, direct-manipulation interface that helps expedite design while informing pedagogy. By making tasks easier and more interesting, we provide an enticement that should attract many, opening the door to the more complex field of pedagogical patterns.

Course Authoring

The primary perspective in PACT is course view. This display (Fig. 1), shows the learning objects comprising a course connected to references to pedagogical patterns that indicate why design choices were made. These core artifacts of PACT are reifications of both course content *and* the instructional expertise that was needed to create it. The encapsulation of these two concepts is considerably more powerful than either one alone. These pattern-annotated courses are put into PACT by expert instructors who designed the course with solid pedagogy in mind and have taught the course over several iterations.
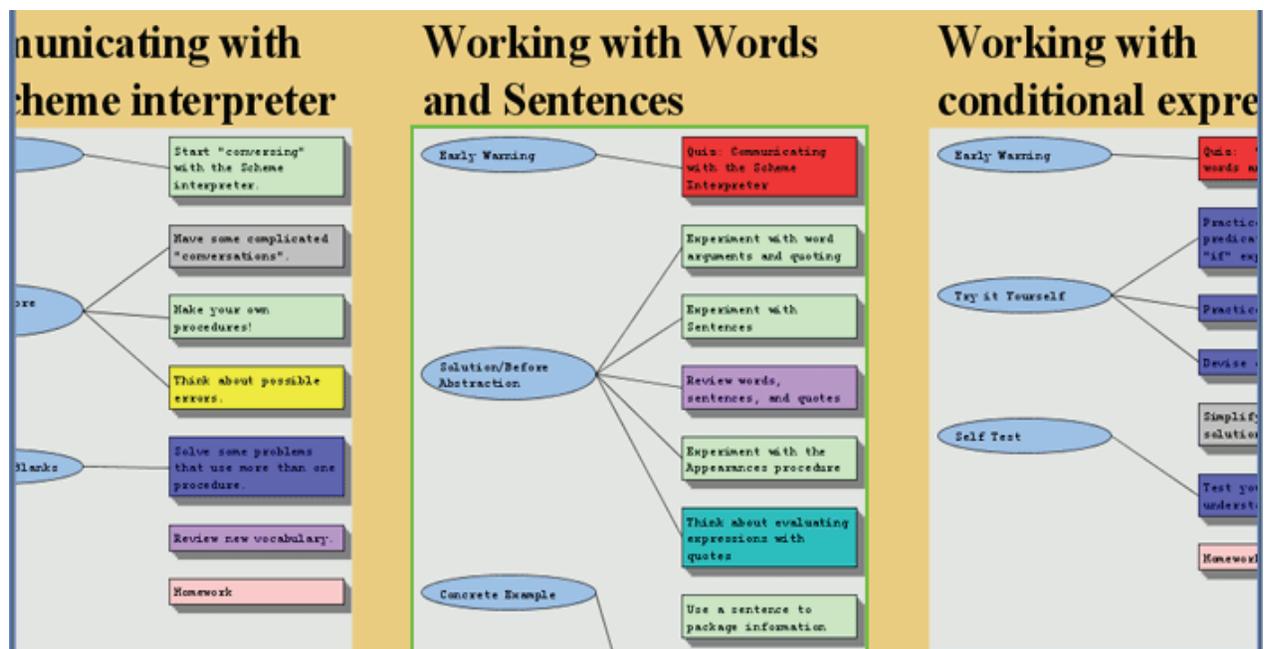

Figure 1: A Day of CS 3, Plus Context as Annotated in PACT

A pattern-annotated course is a very useful reference when creating new curriculum and is exceptionally simple to modify to suit a new situation or instructor. Naïve instructors learn about pedagogical patterns and their value by examining their uses in sensitizing examples – courses that the instructor can relate to in a meaningful way. The process of annotating a course with pattern references in PACT is also useful to experienced instructors as it encourages deep reflection and consideration of course goals and pedagogical methodology. Successive refinements over the course of several semesters can be undertaken in a structured manner using PACT.

Pattern Authoring
PACT contains an additional perspective—pattern view—in which the user can browse simple representations of the patterns in the repository. These patterns are organized by source and purpose to assist in locating patterns of interest. Within this view, pattern authors can establish their own pattern categories and author their own patterns. Making pattern authoring simple encourages exploration and experimentation while guiding individual authors towards adherence to standards that will make their patterns more useable for the community at large.

Intuitive Navigation
PACT is implemented on top of the Piccolo Zoomable User Interface (ZUI) toolkit (Bederson, Grosjean, & Meyer, 2004). The ZUI metaphor makes navigation between different scopes within a course (e.g. an activity, a day, an entire semester) simple. Users can literally "dive in" to the details of a short class segment or see the entire "lay of the land" (J. Bergin, Fourteen Pedagogical Patterns) from far above. This powerful visualization brings a curriculum out of the abstract and gives it a sense of physical structure. The association between this structure and annotating patterns helps bring the patterns into the concrete. Users can easily keep track of their location in the course as a whole thanks to the context provided by surrounding objects. This "focus plus context" (Baudisch, Good, Bellotti, & Schraedley, 2002) helps ease cognitive burden while navigating.

Direct Manipulation
All interface elements in PACT are directly manipulable. Users can easily reorganize learning objects, collections, pattern references, and links between objects and patterns with familiar mouse motions. Connecting the behavior of course objects directly to the user's input further reifies what were once abstract concepts. The sleek simplicity of rearranging course elements helps to encourage experimentation with a variety of course/pattern configurations and provides an incentive to instructors who want to adapt existing courses to their own needs.

Informal Representations
Instructors must be willing to make changes to existing courses to get the full benefit of PACT. Unfortunately, many existing diagramming tools (e.g. Microsoft Visio) use "presentation-style" graphics such as 3D objects, intricate shading, and elaborate fonts. These visual elements suggest a highly evolved or final diagram. PACT uses diagram representations that feature solid boundaries, informal fonts, and playful colors. This style suggests an intermediate and abstract sketch and has been shown to encourage change and exploration (Hong & Landay, 2000).

Pattern Cloning
Users will often see interesting instructional choices in another course that would be useful in their own design. Beyond simply copying the names and descriptions of the patterns that define the intriguing pedagogy into the new course, it is interesting and useful to copy the entire structure associated with that decision. For instance, an "Early Warning" (J. Bergin, Eckstein, Manns, & Sharp, Feedback Patterns) pattern might be connected to a series of assessment objects. The pattern reference is most useful in the new course with copies of those learning objects (or at least some notation indicating that they were assessment objects) attached. We are currently investigating ways of making this feature useable and informative.

Customizable Course Views
Inspecting a fully-annotated course can be overwhelming at first. Often a novice instructor will be interested in only a certain type of pattern or patterns relating to certain objects. We are working on mechanisms for masking out sets of patterns and accentuating others according to the user's present needs. Unfortunately, no widely accepted classification scheme exists for pedagogical patterns. Until such a system comes into common use our efforts are focused on hiding or accenting all references to a common pattern or to patterns from the same source (generally a publication or website). In practice, this method is sufficient as patterns from a common source tend to be commonly themed (e.g. "Feedback" (J. Bergin et al., Feedback Patterns)). However, this topic must be given careful consideration by the pedagogical pattern community as efforts expand to accommodate a larger body of pattern authors.

Opportunities for Community Involvement and Future Work

Our efforts thus far have primarily focused on the user-centered, iterative design of the PACT application. However, during this process we have worked to create an example artifact: a pattern-annotated representation of UC Berkeley's Computer Sciences 3: Introduction to Symbolic Programming. CS 3 is delivered using the UC-WISE system (Clancy et al., 2003; Linn, 2005). UC-WISE provides access to a database of annotated learning objects that include online discussions, programming exercises, reading of Web-delivered text, reflection notes, journal entries, quizzes, and "gated collaborations" as described earlier. The image below shows one day's activities from CS 3 as it appears in PACT (Fig. 1). This successful annotation work has shown that our approach is generally valid and useful, but much work remains to be done. Our work on this annotation has revealed several interesting items.

Course Annotation
We can confirm that annotating a course description does, indeed, provoke careful reflection on course design. Unfortunately, this careful reflection takes time and, as of this writing, we have only completed a full annotation for one course. As one of the central tenets of this project is presenting novice instructors with a rich repository of well-annotated courses, it is difficult to study the effectiveness of our approach with such a small set of annotated learning objects. Thus, there is a clear opportunity for community involvement in this aspect of the project. Expert course designers of all disciplines are encouraged to use PACT to create visual representations and annotations of their own favorite courses. These courses will be added to the PACT repository so that novice instructors may benefit from your unique design choices and the exposure of the thought process behind them.

Pattern Authoring
The members of the Pedagogical Patterns Project have created a robust set of patterns to describe typical lab based active learning environments. Additionally, many of their patterns are general enough to cover a wide variety of situations and environments. However, as educational media and technology evolve there is a constant need for new pedagogical abstractions. The new affordances offered by unique platforms (e.g. tablet computers (Kam et al., 2005), peer instruction devices (Cue, 1998)) present new opportunities for pedagogical design insight. Educators with well-vetted techniques for optimizing environments featuring these new technologies can provide a valuable service to the community by developing new pedagogical patterns. Even instructors that are not comfortable abstracting general patterns away from their learning environments can help the effort by identifying general patterns that can be applied to familiar, yet unique, uses of technology.

Pattern Metadata
The current PACT pattern repository is comprised of overly simplified pattern references. The vast majority of these references are simply the title of a pattern, a few hand-selected keywords, and a URL pointing towards more complete descriptions. The course authoring and annotation processes would benefit greatly from richer metadata. Obtaining this, plus stable versions of pattern descriptions for tighter integration, requires support from the pedagogical pattern community. This is even more significant of an issue with newly developed patterns covering recent design innovations.

Pattern Constraints
The current version of PACT recognizes the pervasive belief that pedagogical patterns are best viewed at a level of abstraction that precludes association with specific types of learning object. However, we have found this viewpoint to be overly restrictive. We believe that a stronger pattern constraint language would greatly facilitate pattern-annotated course development. Such soft constraints would scaffold novice users towards the normative uses of patterns that the pattern author intended.

Moving in this direction requires both a language for expressing constraints and standardized hierarchies of learning object types for any pedagogical pattern language that uses this system. To encourage authors to pursue this direction, PACT includes facilities for specifying object categories relevant to a pattern language or platform and for categorizing learning objects within a course. Work on a constrain-specification language remains for the future and would benefit greatly from community buy-in.

Conclusions

Though the pedagogical patterns movement provides a strong theoretical solution to the current disparity between educational research and practice, efforts grounded in this theory have seen little practical application to reeducating instructors. In response to this deficiency, we offered PACT, a learner-centered tool designed to bridge the gap between theory and practice. We presented the justifications from the learning sciences for design decisions realized in PACT and addressed the goals of the project. Finally, we showed a small portion of a much larger annotation project and identified areas of opportunity for community involvement. We believe that PACT shows great potential as the centerpiece of a reinvigorated, expanded, and distributed pedagogical pattern movement. **The PACT website is: http://www.cs.berkeley.edu/~acarle/PACT/**

References

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language* Oxford University Press.

Baudisch, P., Good, N., Bellotti, V., & Schraedley, P. (2002). Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems,* Minneapolis, Minnesota, USA, 259-266.

Bederson, B. B., Grosjean, J., & Meyer, J. (2004). Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering, 30*(8), 535-546.

Bergin, J. *Fourteen Pedagogical Patterns* .from http://csis.pace.edu/~bergin/PedPat1.3.html

Bergin, J., Eckstein, J., Manns, M. L. & Sharp, H. *Feedback Patterns.*, 2005 from http://www.pedagogicalpatterns.org/current/feedback.pdf

Bransford, J., Brown, A. L., & Cocking, R.R, National Research Council (U.S.). *Committee on Developments in the Science of Learning*

Brown, A. L., & Campione, J. C. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice*. Cambridge: MIT/Bradford.

Clancy, M., Titterton, N., Ryan, C., Slotta, J., & Linn, M. (2003). New roles for students, instructors, and computers in a lab-based introductory programming course. *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education,* Reno, Navada, USA, 132-136.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator, 15,* 6-46.

Cue, N. (1998). A Universal Learning Tool for Classrooms? *Proceedings of the First Quality in Teaching and Learning Conference,*

Dillenbourg, P. (1999). *Collaborative Learning: Cognitive and Computational Approaches* Oxford, UK: Elsevier Science Ltd.

Fincher, S., & Utting, I. (2002). Pedagogical patterns: their place in the genre. *ITiCSE '02: Proceedings of the 7th annual conference on Innovation and technology in computer science education,* Aarhus, Denmark, 199-202.

Hong, J. I., & Landay, J. A. (2000). SATIN: a toolkit for informal ink-based applications. *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology,* San Diego, United States, 63-72.

Kam, M., Wang, J., Iles, A., Tse, E., Chiu, J., & Glaser, D. et al. (2005). Livenotes: a system for cooperative and augmented note-taking in lectures. *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems,* Portland, Oregon, USA, 531-540.

Lilly, S. (1996), Patterns for Pedagogy. *Object Magazine, 5*(8)

Linn, M. C. (2005). WISE design for lifelong learning--Pivotal Cases. In P. Gardenfors, & P. Johansson (Eds.), *Cognition, Education and Communication Technology*. Mahwah, NJ: Lawrence Erlbaum Associates.

Sharp, H., Manns, M. L., & Eckstein, J. (2003). Evolving Pedagogical Patterns: The Work of the Pedagogical Patterns Project. *Computer Science Education, 13*(4), 215-330.

Sharp, H., Manns, M. L., & Eckstein, J. (2000). The pedagogical patterns project (poster session). *OOPSLA '00: Addendum to the 2000 proceedings of the conference on Object-oriented programming, systems, languages, and applications (Addendum),* Minneapolis, Minnesota, United States, 139-140.