Participatory design in open education: a workshop model for developing a pattern language

Yishay Mor and Niall Winters

London Knowledge Lab

23-29 Emerald Street
London WC1N 3QS
United Kingdom

http://www.lkl.ac.uk/

Abstract: Technologically enhanced learning environments raise complex challenges for their designers, developers and users. Design patterns and pattern languages have recently emerged as a potential framework for addressing some of these challenges. However, the uptake of design patterns has been slow outside of the computer science community. We argue that this is largely a consequence of a weak positioning of pattern languages, as a form of delivering expert knowledge to layperson, and suggest an alternative view: the development of a pattern language as a community endeavour. In terms of open education, the workshop model can be viewed as an open production process for developing educational resources, in our case design patterns. We propose a model of pattern elicitation workshops, in which collaborative development of a pattern language provides a framework for sharing design knowledge within interdisciplinary communities. This model was iteratively developed at five international conferences. It was then postulated as a design pattern itself, encompassing a series of practices and a set of supporting tools. We believe this model could be applied in a broad range of communities concerned with the development of open digital educational resources.

Keywords: Design Patterns, Pattern Languages, Open Learning, Case Studies, Methodology, IDR, Games, Mathematics Learning

Interactive Demonstration: A video course demonstrating the workshop model is available at:

http://lp.noe-kaleidoscope.org/outcomes/videos/

1. Introduction

Technology enhanced learning (TEL) is a demanding business. Teaching practitioners are continuously required to evaluate, adopt and integrate new technologies into their practice. This process is complicated by the rapid evolution of technology. One immediate impact of this is that it difficult for practitioners to formulate considered observations that could inform future techno-pedagogic developments in TEL. There is a two-way gap in crucial knowledge between those creating the technology and those using it: innovative technologies frequently reify conservative pedagogies; ground-breaking ideas are manifested in low-quality or unsustainable technological tools. If TEL is not to reach a crisis point, researchers, in collaboration with practitioners and industry, must urgently address this issue. A potential starting point is to inspect and reflect upon the *practices of producing technology and activities for use in educational contexts*, combined with a rich theoretical perspective. The question we wish to explore is how to facilitate productive design-level discourse among those involved in TEL development. We have argued elsewhere (Mor & Winters, 2007) that pattern languages offer a viable way forward in this respect. In this paper, we expand on that argument, highlighting Pattern Languages' potential as a framework for open participatory design. The open process support production of education resources, in our case design patterns.

TEL development practice is by and large dominated by a *closed model*. Closed models are associated with the practice of engaging with a community of peers who have preferential access to information, content and tools. Crucially, the details of how outputs were developed are not open to public scrutiny. While this has clear advantages for commercial players within education it has financial implications for those in the frontline. Most commercial TEL players still follow this model, claiming that it provides a financial incentive for producing high-quality products.

An alternative approach which is gaining popularity and application is the *open model*. As distinct from the closed model, developments are made available to everyone, usually free of charge. The model is further differentiated by the fact that both the finished outputs (e.g. software applications) and the processes by which they were developed (e.g. software code or discussion on curriculum development) are also publicly available. A good example of an open model in education is the Open Archives Initiative. This initiative can be contrasted with traditional closed models, such as subscription-based journals, by its remit as an organisation that "develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content … to enhance access to e-print archives as a means of increasing the availability of scholarly communication" (Lagoze & Van de Sompel, 2008). Another notable example is the Journal of Interactive Media in Education (JIME), which applies an open model not only to the final papers, but to the review process as well (Buckingham Shum and Sumner, 2001). Open models cover both open source software and open educational resources. The first category ranges from operating systems (e.g Edubuntu) through generic tools (such as OpenOffice) to educational applications and packages (Sage, Kde-Edu, GeoGebra, DSpace)[1]. Prominent open content initiatives include openlearn and opencourseware.

Vogel and Oliver (2006) review the use of three VLEs in ten British colleges. Out of these, four chose closed source products (Blackboard and WebCT) and six chose an open source one (Moodle). The users of the open source VLE reported advantages in the level of control it offered them. Five of the institutions using Moodle had previously used a commercial, closed source product. While showing a vote of confidence in the open alternative, in a way, this fact points at one of the advantages of the closed model: because it can attract far greater funding, it can potentially deliver products and updates to the market much faster.

The potential of open models is amplified by the nature of TEL. The development of TEL resources is an interdisciplinary process (Winters, Mor & Pratt, forthcoming). No single stakeholder has access to all the relevant knowledge, while quantifying the respective knowledge components in a manner that can be monetized and traded in a closed development model is practically impossible. The Learning Patterns project (Pratt et al, 2007) demonstrated this assertion in the domain of games for mathematical learning. The development and deployment of such requires deep knowledge of mathematics, pedagogy, software engineering, game design and didactics. Each dimension of this knowledge is manifested in a different profession or discipline. This configuration cries for an open model. Yet openness needs to go beyond code and content. Free and open access to educational tools and resources goes a long way towards narrowing digital divides and consequently reducing knowledge inequalities. But the availability of a tool or resource does not entail the possession of the knowledge how to use it effectively. Furthermore, for tools, resources and practices to be suitable for diverse cultural and institutional settings, they need to be designed within these settings. Open access to the source code is not enough to guide you in the modification and repurposing or even reconstruction of the tools.

We wish to draw attention to the problem we call *the design divide*; the gap between those who have the expertise to develop high-quality tools and resources and those who don't. We frame this challenge based on Simon's broad view of design (1969), as any deliberate cognitive activity aimed at engendering a more desired state of the world. Thus, software developers are designers of technology, educational practitioners are designers of learning experiences, policy makers are designers of educational systems, and so on. Each has expertise in their domain, and some understanding of the

[1] http://www.sagemath.org/, http://edu.kde.org/, http://www.geogebra.org/, http://www.dspace.org/

others. We see these as *design knowledge* (Mor & Winters, 2007). Obviously, one's design knowledge of other domains can never match her intimate familiarity with her own. Yet these strands of knowledge are inherently inter-related; one cannot design a good tool without having a substantially detailed perception of its use. Likewise, it is hard to conceive of a tool being used effectively without knowledge of how it was designed to be used.
Sharing code and content is an open manner is fairly straightforward: publish the artefacts in a standard format in a freely accessible medium. Sharing design knowledge is much more challenging. Firstly, most design knowledge is tacit, it is implicitly coded in the intuitions of expert designers and the languages they use to converse. Secondly, even where there is a willingness to open up the design process and engage in an inclusive discourse, there is no agreed format for expressing design knowledge. The first issue was, to an extent, addresses by the participatory and user-centered design movements (Nesset and Large, 2004; Kaptelinin, Danielsson and Hedestig, 2004). The second was a driving force underlying Christopher Alexander's proposal of *design patterns*. A concrete example of a pattern from our language is provided in Appendix A.

When Christopher Alexander and his colleagues proposed the notion of design patterns (Alexander, Ishikawa and Silverstein, 1977), it was out of the recognition of a similar gap in architectural design. They argued that the design knowledge of built environment is monopolised by architects and engineers, and thus deprived from the users of its products. The aim of their project was to democratise design, giving people the power to design their living environment. As noted by Dearden and Finlay (2006), Alexander's pattern language was "intended to enable users to actively and directly design their own living and working spaces, in part by providing a common language with which they could make proposals and discuss ideas with an 'architect-builder'." (p. 27) The role of architects or other experts would be to advise, highlight potential issues, and propose methods of solution which have been refined by a history of successful applications. Design, as we defined it above, concerns the resolution of problems in their context. *Design patterns* are constructs which capture a recurring problem, the context in which is occurs, and a successful method of solution.

Since the seminal work of the pedagogical patterns project (Bergin, 2000; Sharp, Manns and Eckstein, 2003), several projects have attempted to introduce a patterns-based approach to educational design, and specifically to the design of educational technology (Derntl & Motschnig-Pitrik, 2005; Goodyear, 2005; McAndrew, Goodyear and Dalziel, 2006; Lukosch and Schümmer, 2006; Retalis, Georgiakakis and Dimitriadis, 2006). Fincher et al (2001) note that patterns are a particularly suitable format for sharing practice-related knowledge because they offer a representation which "does not prescribe or patronise and equally does not "overwhelm with information". Despite the vibrant research activity, pattern based approaches appear to have little impact at the field level. Schümmer, Lukosch and Slagter (2005) argue that this is largely due to the lack of user inclusion in the process of developing pattern languages, inspired by the success of pattern languages in software design. The tradition of pattern languages in computer science has by and large ignored the participatory and communicational aspects highlighted by Alexander, and repurposed them as a format for expert software designers to share their design knowledge with novices. This has proven effective in initiating young engineers in the art of programming, and led to the proliferation of high technical standards. Yet at the same time it has intensified the status of software production as a specialist activity, inaccessible to laypersons, thereby excluding users from the design process. Similar criticism, in the realm of HCI, is raised by Dearden and Finlay (2006).

Our analysis, in the course of the Learning Patterns project, has produced similar conclusions. The desire to reintroduce the community aspect of pattern languages was the driving force behind the development of the IDR methodology (Winters and Mor, 2008). We have recently learnt of the Oregon Software Development Process, proposed by Schümmer et al (2005) in the context of groupware development, which bears interesting parallels to IDR. The process of eliciting design knowledge from a community is far from trivial. Goodyear (2005) argues that the process of pattern elicitation must be inherently iterative. The language must be allowed to evolve as a cohesive ecology of ideas. Whenever a new pattern is introduced, it perturbs the structure of the language and modulates existing patterns. The next section outlines the main components of our framework. The rest of this paper illuminates a

key component in that methodology, the pattern elicitation workshop. It is derived from a series of workshops at international conferences, in which practitioners, designers and researchers where engaged in effective sharing of interdisciplinary design knowledge through their participation in the refinement of a pattern language for games for mathematical learning.

1.1. Eliciting patterns: the IDR methodology

IDR stands for "identify, develop, refine". It is an iterative methodology (Winters and Mor, 2008) for the dynamic growth of a pattern language as a stratum for design-oriented discourse in interdisciplinary communities. The IDR framework utilises three main knowledge structures: Typologies, case studies and patterns. All three are supported by multiple representations, indexing and navigation schemes, and bespoke authoring tools. These were developed in tandem with a set of social practices.

Typologies provide a structured lexicon for classifying the critical aspects of design knowledge relevant to the community. In our project, we were concerned with mathematical games and so our typologies were: *mathematical content*, *learning and instruction*, *educational context*, *games*, *interaction design* and *software design*. We developed a *typology tool* which allows community members to browse, review and edit the various typologies. A typology is a complex monolith which captures the knowledge of a single specialist (or specialist group). Hence it is convenient to edit it off-line and upload versions as they mature. This is achieved by using a mind map editor (*FreeMind*). Once a typology is uploaded, it can be viewed either as a map image or as an html tree. The definitions of the terms are displayed alongside the tree view, and can be edited online. Each typology map is accompanied by a discussion forum, where other members of the community can comment, suggest changes or ask for clarifications. When a new version is uploaded, the previous versions are retained for reference. In order to facilitate cross-referencing between typologies as well as typology-driven context descriptions in the case studies and patterns, we provide a wiki-style method of linking to typology terms.

Typologies served two critical functions in our framework. At the first instance, they provided an agreed lexicon for cross-disciplinary conversation. Through the construction of a typology, each domain expert would set the reference lexicon for others to use when referring to issues in her domain. The iterations of review by peers from other domains would force the typology editor to elucidate and elaborate this lexicon. The second role of typologies was a structured and detailed checklists of issues to consider when describing a case study, as explained in the following section.

Our conception of case studies sees them as *structured narratives* of expert practice (Winters & Mor, forthcoming). Schank and Abelson (1977) argue that stories about one's experiences, and the experiences of others, are the fundamental constituents of human memory, knowledge, and social communication. They call for a shift towards a functional view of knowledge, as Schank (1995) explains: "intelligence is really about understanding what has happened well enough to be able to predict when it may happen again" (p. 1). Such knowledge is constructed by indexing narratives of self and others' experiences, and mapping them to structures already in memory. Bruner (1986; 1990; 1991; 1996) identified narrative as the predominant vernacular form of representing and communicating meaning. Humans use narrative as a means of organizing their experiences and making sense of them. A narrative is always contextualized. It habitually begins with an exposition, which lays out the context: time, location, props and characters. These ideas are supported by recent findings in neuropsychology and cognitive psychology (Mar, 2004; Atance and O'Neill, 2005; Atance and Meltzoff, 2005).

In our framework, case studies were the entry form of representing notable incidents from practitioners' experience. These were then used as the raw material from which patterns were moulded. Community members offer case studies for discussion by using a simple template which provides *soft-*

scaffolding:[2] it suggests a structure, but does not impose it. Case studies are automatically indexed and tracked in the *case study repository*. The template prompted contributors to provide the context, aims, details, outcomes and references. The details section, which was typically the main bulk of the case study, is a free-form narrative. Contributors were encouraged to include graphical materials, such as screen shots and diagrams.

Design patterns, and their weaving into a pattern language, are of course the ultimate product of our methodology. Nevertheless, they do not deprecate the typologies and case studies used in their construction. On the contrary: a fully-qualified pattern makes extensive use of the typologies in defining its context, and maintains links to the case studies in which it is manifested as examples. The context of a pattern is critical in defining when and where it is relevant, and the examples are essential for readers to understand it. Patterns also share a common template, much more detailed than the one used for case studies. This template includes a fixed meta-data section, which notes the patterns place in the language hierarchy, its status, ownership name and summary. Additional "soft" sections of the template prompt the author to specify the problem it addresses, its context, the pattern body and links to other patterns and examples. Patterns evolved through four phases: seed, alpha, beta, and final. Seed patterns emerged from discussing the critical design elements emerging from a case study. They were often little more than a note taken during discussion. These were later promoted to alpha status when elaborated by their author and presented for internal review. Following the review they were refined, mapped into the language structure and linked to related patterns, thus attaining beta status. At this stage, they are presented for public review, after which they are marked having a final status.

The process of extracting patterns from case studies parallels Bruner's notion of scripts. Bruner (1990) talks of scripts as the indigenous form of encoding knowledge. A script is a recipe for solving problems. It includes the context in which it is applicable, the sequence of operations to carry out, and the expected implications. Narrative, as a cognitive and communicational construct, has a central role in constructing and sharing scripts. A narrative is grounded in a context, describes a sequence of events, and implies expected outcomes. It serves as the prototype after which the script is moulded. Recently the CSCL community has witnessed considerable interest in the prospect of designing collaboration and interaction scripts (Kollar, Fischer and Hesse, 2006; Kobbe et al, 2007). This trend draws on the ideas of Bruner, Abelson and Schank with two significant exceptions: CSCL scripts are syntactic, in contrast with the tacit nature of the original idea, and they do not include a pattern of preconditions to trigger them. It would appear that CSCL scripts are akin to design patterns, lacking the latter's clear description of problem and context, but adding a rich ontology of interaction semantics. Hernández-Leo et al (2006) make a promising attempt to synergize the two constructs. A comprehensive discussion of the relationship between case studies, narratives, scripts and patterns is beyond the scope of this paper.

2. Participatory Pattern Elicitation Workshops

The IDR methodology emerged from the internal work of the Learning Patterns project team. Having observed initial indications of its success, we decided to apply it to a wider community. We initiated five workshops at international conferences, where practitioners, designers and researchers exchanged design knowledge by discussing case studies and extracting patterns from them. The purpose of these workshops was to produce new insights regarding common design challenges and methods of resolving them. They did not involve the production of any specific designs, solutions or learning materials. Between May 2006 and February 2007 we conducted workshops in Preston, Chelmsford, Berlin, Hanoi and Ra'anana (Israel). Each workshop included between 15 and 30 participants, and lasted between two and six hours. These participants included educational researchers, higher education teachers and managers, designers and developers of educational content and software, game designers and developers, and computer scientists. Workshop participants were contacted a couple of weeks before the event, and asked to introduce themselves via a blog or mailing list, and contribute a

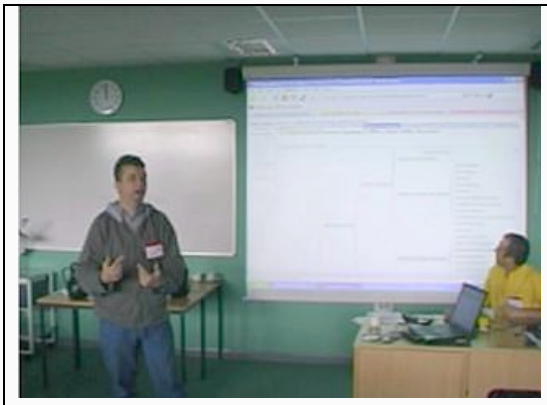
[2] Soft Scaffolding is in itself a prominent pattern in our language (http://lp.noe-kaleidoscope.org/outcomes/patterns/Soft_scaffolding/)

case study using the on-line authoring tool (Figure 1). The incentive for contributing a case study was the opportunity to receive considered feedback on one's work and pressing issues. These case studies were used by participants as a starting point for cross-disciplinary design discussions. The end products of these discussions were encapsulation of design knowledge in the form of a collaboratively constructed pattern language.

> Hi all and thanks for registering for the workshop! If the rest of this mail does not display correctly, you can view it on-line at: http://lp.noe-kaleidoscope.org/workspace/educa/
>
> Our time at EDUCA is limited. To make the most of it, please try to do the following before the event:
>
> Post your personal profile on the group blog. This should only take a few minutes, and the sooner you do it - the better!
>
> Read: Yishay Mor and Niall Winters (in press), Design approaches in technology enhanced learning. Interactive Learning Environments, Taylor & Francis. (PDF). This paper lays out the theoretical background for the project. We want to leave as much time as possible for hands-on experience on the day, so we will assume you have read this paper and only provide a *very brief* review. If you have any questions, please post them on the group blog or email us.
>
> Have a look at the case studies and let us know if you would like to add one for us to analyse on the day.
>
> You can have a look at videos from previous workshops to get a better sense of our approach.
>
> Please do not hesitate to contact us with any questions or comments you may have.
>
> All the best,

Figure 1: example workshop introductory email

Workshop events began with a presentation of the theory and practice of design patterns, followed by a demonstration of the IDR methodology. The demonstration would follow one of our *trails*: easy to follow narratives originating from one of our case studies through its mapping to typologies and leading to the derived patterns. This demonstration set the example for participants to follow. Working in groups, they would discuss one of the pre-collated case studies. The contributor of the case study would present it, and respond to clarification questions. The group would then map the case study to the various typologies. In this process, the details of the case study would be elucidated. Participants would then compare it to similar cases from their experience. These would be surprisingly diverse, where the unifying factor was always a common design challenge. Having identified two or more situations in which this problem emerged, the group would be well-positioned to identify the defining characteristics of the problem and its context which afford a solution that is transferable across practices. This triad of context, problem and skeletal solution would be noted as a seed pattern.



Walking through one of our trails

Discussing case studies and eliciting patterns

presenting back to the workshop assembly

concluding discussion

Figure 2: scenes from a workshop

Towards the end of the workshop, we would reassemble in full forum and each group would present its case study and derived patterns. Participants would then reflect on the process of pattern elicitation, its outcomes, and their relevance to their daily work.

The dynamics and practices of these workshops were captured as a video course, available at: http://lp.noe-kaleidoscope.org/outcomes/videos/.

3. Results

Responses from workshop participants where highly positive, and many indicated that they believe the tools, methods and patterns which we introduced will have a positive impact on their future work. Unfortunately, we were not able to monitor participants after the event, and therefore cannot validate this initial impression.

Out of the 25 case studies in our repository, 11 were contributed by workshop participants. As for patterns, out of over 150 listed in the repository only 13 where directly contributed by workshop participants. These numbers are negatively biased: in the shorter workshops, many case studies and patterns where developed verbally, or with the aid of flipcharts and markers, and were not recorded in the system. Furthermore, some of the workshop discussions were eventually folded into existing patterns, thus enriching them and expanding their remit. Nevertheless, it is obvious that there is a clear room for improvement. Indeed, the one comment that consistently recurred in all workshops was the demand for more time. In retrospect, this is not surprising: our approach highlights the pattern languages as a shared, participatory, community resource. But communities are not formed in 90 minutes. For our workshops to reach their full fruition, they would need to be embedded in a prolonged

trajectory of joint enterprise.

To recap, these results should be taken as a proof of concept. As such, we posit that they are impressively convincing: practitioners from highly diverse backgrounds, with no prior exposure to the notion of pattern languages were able in a short while to identify salient elements of best-practice which cross the boundaries of their respective realms and conduct a productive design-level discussion.

4. Discussion

The Learning Patterns project was prolific in the volume (e.g. over 150 patterns) and quality of outputs (e.g. 4 journal outputs and 1 book chapter) over a very short time, with limited resources. We believe that this success could not have been attained had we chosen a closed model of development. Our open approach was applied to content, code and process.

The project set out to facilitate the sharing of design knowledge in the domain of games for mathematical learning. From this premise, it was obvious that any knowledge or artefact generated by the project should be free, open and easy to access. This agenda posed a challenge in the context of academic research, where contributors are deeply concerned about the validity and accuracy of any public representation of their work. In order to accommodate this tension, we needed to develop tools which facilitated continuous discussion and refinement while clearly distinguishing between drafts and outputs, and between owning authors and discussants. Maintaining the code for these tools as an open resource allowed for its use in future projects.

Extending the open model to the work processes of the project team required a cultural shift on behalf of some members. However, since the advantages were evident, this was not hard to engender. Working in a highly distributed environment, any closed model of collaboration would have induced overheads in terms of management, development and maintained which would have impeded our progress. The choice of an open model not only made our work easier, it also offers anyone who wishes to scrutinize it complete transparency. On the other hand, if other groups wish to replicate our methodology, they have access to its "nuts and bolts", and can thus effectively build on our experience.

The triad of typologies, case studies and design patterns proved to be an effective medium for cross- and inter- disciplinary sharing of design knowledge. For the project team, this allowed us to refine our understanding of the complexities involved in the development and use of games for mathematical learning. The workshops extended this scope to include a wider range of educational domains. We were intrigued by the unexpected links participants made across seemingly unrelated topics such as citizenship and mathematics education.

4.1. Challenges in supporting open content development

One of the aims of our workshops was to encourage and support interdisciplinary practice through the capturing and sharing of design knowledge. Our workshops were framed as models of *open development*. One implication of this was that from the outset participants were engaged in a culture of openness. At no stage were they giving up the "rights" to their inputs but instead were using them as mediating tools for discussion and iteration. Thus, the resources participants brought to the workshops (in our case narratives of case studies of practice) were openly sharable. In turn, this meant that they could be use to seed development of typologies, and by extension design patterns. In particular for education practitioners, this has a number of benefits: (i) The content they bring has a direct bearing on the form the outputs take providing them with an explicit and direct impact on the process; in our project, practitioner case studies were the basis upon which patterns are developed, (ii) Participants can share aspects of the process with their peers for feedback and critique free of the complexities introduced by IPR, (iii) These peer consultations are publicly available, providing an insight into the how any particular perspective/stance emerged.

However, there is one major challenge raised by the above process. What about tools, practices and

other resources developed in a closed manner? This is important to address as some of our workshops were aimed at developing industrial-academic links. Within education, it seems that fruitful links are being built with the industrial community. The rationale for an open approach is not to seek and end to the closed model. There are indeed times when it is required. A more pertinent question to ask is what aspect of the interdisciplinary development approach the open model can support. We have had some experience with this. One solution is to make the development process open, i.e. supporting structures, models and tools are freely available and open. However, the closed model covers the commercial outputs, i.e. TEL software and its associated code. All communities get what they want from the process: academics are free to publish; the community receives open tools, open sharing of practice and new knowledge, and the industrial partners commericalise products. If such a solution is successfully implemented, the resulting outputs benefit from being both pedagogically sound and industrial strength.

4.2. Impact of following the open model

By following an open model of development for *process and tools*, the impact on the TEL community is potentially vast. In our case, our tools and details of how our practices emerged and developed over time were made freely available on the web. This open approach had implications in terms of (i) process management, (ii) tool take-up and use, and (iii) practice sharing and capture.

(i) Process management

A key component of our open model was that our *emergent processes* were publicly available. If we had followed the closed approach, we would have been in the position of only sharing our process amongst the distributed team of project partners. This was not the case: managing the process of pattern development was not confined to a "team voice". Decisions, in particular with respect to the structure of the pattern language and the transition of patterns from *seed* to *beta* to *release* were informed by the critiques received and online public discussion. Indeed by working in an open manner during the five workshops impacted upon our pattern language by better informing us about pattern relationships, determining for example which should be pruned/folded together. Moreover, we were motivated to provide initial pattern examples and mechanisms for workshop participants and the wider community to use them. This was one of the motivations for our trails – the need to provide 'ways-in' for novice users, supporting their participation in the project. One example of the concrete outputs of such support was the contribution of 11 case studies, which in turn influenced our practices as a project team. From our experience, the key message for open model process management is the need to provide mechanisms for wide community influence and to develop timelines and practices within the core team to support this.

(ii) Tool take-up and use

Our workshops attracted over 200 participants, each of which used our tools to contribute their case studies. Our website containing the suite of online tools received over 10,000 unique visits. Pragmatically, if we needed to support paid access to the site, resources spent on tool development would have been diverted. By following on open model, we could make all of our tools readily available in order to attract a critical mass of users within the 1-year timeframe of the project.

(iii) Capturing and sharing practices

Within the TEL community there is a real need for the communities practice to be open to critique and scrutiny. By making our approach open, this was support and encouraged. Indeed, the ways in which our content (i.e. design patterns) developed required the input of as many in the community as possible. One of the rationales for this is that we wanted our pattern to be developed and used in different contexts. Therefore community input supports patterns moving from *seed* to *release* format. As all contributions are open there is equality between contributors, leading to a high level of transparency. Participants' work in a manner by which a shared knowledge base emerges – each person's expertise is clear and the manner in which expertise changed and developed is captured. In terms of the patterns themselves, each version is available providing a historical perspective on how

the pattern emerged. Therefore, analysis at both the process and tool level is supported.

5. Conclusions

This paper expands on the workshop element of the IDR methodology by detailing an open model for participatory pattern elicitation. This model and the set of supporting tools, were developed iteratively through a series of workshops that brought together researchers, developers and practitioners in productive design-oriented discussions. While our workshops focused on the production and use of games for mathematical learning, in this paper we have focused on the pragmatics of the model, which are invariant to subject. We believe therefore that a broad range of communities concerned with the development of open digital educational resources could apply the model.
Our starting point was an examination of the challenges of developing effective tools and content for education. We argued that the complexity and turbulence of the domain implies that an open model is imperative for success. We highlight the challenge of the design divide; the gap between those who posses the knowledge to produce high-quality educational resources and those who lack it, and stressed the implied need to extend the open model beyond the artefacts, and apply it to the process and practices of design.

6. Limitations and future work

This paper highlighted a particular component of our methodology, as it was manifested in the domain of games for mathematical learning. Issues related to the broader remit of the methodology, in a wider range of communities, are beyond the scope of the current discussion. Some of these are explored in (Winters, Mor and Pratt, forthcoming). Others are the focus of the recently launched Pattern Language Network.[3]

Reflecting on the outcomes of the Learning Patterns project, we identify two significant limitations. The first regards completing the development cycle, the second concerns the form of representation. While both are understandable in the context of a short, modest scale project, they are in the centre of our attention for current and future work.

If the aim of design patterns is to facilitate broad dissemination, cumulation, and scrutiny of design knowledge then their ultimate test is when this knowledge feeds back into practice. While we do have informal reports of workshop participants making use of the patterns in their subsequent work, we aim to extend our approach to cover a full iterative cycle of design, development, implementation, deployment, evaluation and redesign. We believe that this will allow us to benefit from a feedback loop, which would potentially amplify the power of our methodology. Part of the process is understanding the social configuration of development teams (Winters, Mor & Pratt, forthcoming). We posit that the most effective strategy for achieving this would be through the facilitation of sustainable design communities: inter-disciplinary groups of practitioners with a common interest, engaged in a design-oriented discourse over a prolonged period of time. The recently launched Pattern Language Network project, as well as several emerging initiatives, will build on our previous work to progress in these directions.

One of the most striking features of Alexander's work is its visual aspect. His patterns are almost always illuminated by carefully articulated illustrations, combining photographic imagery with abstract schematics. Indeed, in “Notes on the synthesis of form” (1970) Alexander proposes a syntax of diagrams to explore the forces underlying design problems, and identifies these with what he later develops as design patterns. Our system has so far emphasised the graphic representation of typologies and of the pattern language as a whole. In terms of the single pattern, we allowed the author to include any visual materials she chooses, but did not provide any structured support. Reflecting on our outcomes, we identify a need to provide a streamlined scaffolding for pattern authors to construct a

[3] http://patternlanguagenetwork.org/

visual representation of their ideas. In software engineering, UML and BPMN have established a de-facto standard as visual modelling and design languages (Larmann, 2002; White, 2004). However, these have two serious shortcoming when we turn our gaze to the domain of TEL. Firstly, they are not readily readable by the uninitiated. For a trained software engineer, a UML diagram is as clear as Harry Beck's tube map is to a Londoner. Yet most people find the notation unintuitive, albeit visually impressive. Secondly, UML diagrams, and to a lesser extent BPMN diagrams, describe the relationships and interactions between software classes and components. They do not provide a visual syntax for representing relationships and interactions between humans, or indeed any element of human condition. Yet any language which strives to address learning design and design for learning needs to take on these issues in its core. Recent works (Hernández-Leo et al, 2006; Harrer and Malzahn, 2006) show some promising beginnings in this direction. In our current work, we are experimenting with a visual framework which could be used by practitioners to describe their case studies, and would be carried over fluidly to their derived patterns.

7. References[4]

Alexander, C.; Ishikawa, S. & Silverstein, M. (1977), *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*, Oxford University Press, New York.

Atance, C. M. & Meltzoff, A. N. (2005), 'My future self: Young children's ability to anticipate and explain future states', *Cognitive Development* **20**(3), 341-361.

Atance, C. M. & O'Neill, D. K. (2005), 'The emergence of episodic future thinking in humans', *Learning and Motivation* **36**, 126â€“144.

Bergin, J. (2000), Fourteen Pedagogical Patterns, *in* 'Fifth European Conference on Pattern Languages of Programs'.

Bruner, J. (1996), *The Culture of Education*, Harvard University Press, Cambridge, Massachusetts.

Bruner, J. (1991), 'The Narrative Construction of Reality', *Critical Inquiry* **18**.

Bruner, J. (1990), *Acts of Meaning : Four Lectures on Mind and Culture (Jerusalem-Harvard Lectures)*, Harvard University Press, Cambridge, Massachusetts.

Bruner, J. (1986), *Actual Minds, Possible Worlds (The Jerusalem-Harvard Lectures)*, Harvard University Press, Cambridge, Massachusetts.

Buckingham Shum, S. & Sumner, T. (2001), 'JIME: An Interactive Journal for Interactive Media', *First Monday* **6**(2).

Dearden, A. & Finlay, J. (2006), 'Pattern Languages in HCI: A Critical Review', *Human-Computer Interaction* **21**, 49--102.

Derntl, M. & Motschnig-Pitrik, R. (2005), 'The Role of Structure, Patterns, and People in Blended Learning', *The Internet and Higher Education* **8**, 111--130.

Fincher, S.; Petre, M. & Clark, M. (2001), *Computer science project work: principles and pragmatics*, Springer-Verlag, London, UK.

Goodyear, P. (2005), 'Educational design and networked learning: Patterns, pattern languages and design practice', *Australasian Journal of Educational Technology* **21**(1), 82-101.

Harrer, A. & Malzahn, N. (2006), Bridging the gap - towards a graphical modelling language for learning designs and collaboration scripts of various granularities, *in* 'Advanced Learning Technologies ICALT 2006', IEEE Computer Society, Los Alamitos, CA, pp. 296-300.

[4] Full references with links to papers where publicly accessible are available at: http://www.bibsonomy.org/user/yish/jime08

Hernández-Leo, D.; Villasclaras-Fernández, E. D.; Asensio-Pérez, J. I.; Dimitriadis, Y.; Jorrín-Abellán, I. M.; Ruiz-Requies, I. & Rubia-Avi, B. (2006), 'COLLAGE: A collaborative Learning Design editor based on patterns', *Educational Technology and Society* **9**(1), 58-71.

Kaptelinin, V.; Danielsson, K. & Hedestig, U. (2004), 'Towards Learning-Centered Participatory Design: Main issues and challenges', *Paper presented at the First Kaleidoscope SIG CSCL Conference (Lausanne, Switzerland, October 2004)*.

Kobbe, L.; Weinberger, A.; Dillenbourg, P.; Harrer, A.; Hämäläinen, R.; Häkkinen, P. & Fischer, F. (2007), Specifying computer-supported collaboration scripts'International Journal of Computer-Supported Collaborative Learning', International Society of the Learning Sciences, Inc., , pp. 211-224.

Kollar, I.; Fischer, F. & Hesse, F. (2006), 'Collaboration Scripts - A Conceptual Analysis', *Educational Psychology Review* **18**(2), 159--185.

Lagoze, C. & Van de Sompel, H. (2008), 'Open Archive Initiative' website, http://www.openarchives.org/OAI/OAI-organization.php, last accessed 09 May 2008.

Lukosch, S. & Schümmer, T. (2006), 'Groupware development support with technology patterns', *International Journal of Human-Computer Studies* **64**(7), 599-610.

Larmann, C. (2002), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Prentice Hall PTR.

Mar, R. A. (2004), 'The neuropsychology of narrative: story comprehension, story production and their interrelation (review)', *Neuropsychologia* **42**(10), 1414-â€“1434.

McAndrew, P.; Goodyear, P. & Dalziel, J. (2006), 'Patterns, designs and activities: unifying descriptions of learning structures', *International Journal of Learning Technology* **2**(2), 216-242.

Mor, Y. & Winters, N. (2007), 'Design approaches in technology enhanced learning', *Interactive Learning Environments* **15**(1), 61-75.

Nesset, V. & Large, A. (2004), 'Children in the information technology design process: A review of theories and their applications', *Library & Information Science Research* **26**(2), 140-161.

Pratt, D.; Winters, N.; Alexopoulou, E.; Bligh, J.; Björk, S.; Cerulli, M.; Childs, M.; Chioccariello, A.; Jonker, V.; Kynigos, C.; Lindström, B.; Mor, Y.; O'Donnell, F.; Tangney, B. & Wijers, M. (2007),'Kaleidoscope JEIRP on Learning Patterns for the Design and Deployment of Mathematical Games: Final Report'. http://telearn.noe-kaleidoscope.org/open-archive/browse?resource=530

Retalis, S.; Georgiakakis, P. & Dimitriadis, Y. (2006), 'Eliciting design patterns for e-learning systems', *Computer Science Education* **16**(2), 105--118.

Schank, R. & Abelson, R. (1977), *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*, Lawrence Erlbaum Associates, Hillsdale, NJ..

Schank, R. C. (1995), *Tell Me a Story: Narrative and Intelligence*, Northwestern University Press.

Schummer, T.; Lukosch, S. & Slagter, R. (2005), Empowering End-Users: A Pattern-Centered Groupware Development Process, *in* 'Groupware: Design, Implementation, and Use, 11th International Workshop, CRIWG 2005', Springer-Verlag , Berlin Heidelberg, pp. 73-88.

Sharp, H.; Manns, M. L. & Eckstein, J. (2003), 'Evolving Pedagogical Patterns: The Work of the Pedagogical Patterns Project', *Computer Science Education* **13**, 315-330.

Simon, H. A. (1996), *The Sciences of the Artificial - 3rd Edition*, The MIT Press, Cambridge, MA.

Vogel, M. & Oliver, M. (2006),'Design for learning in virtual learning environments—insider perspectives', *JISC Learning Design Tools Project Report. Available online at: http://www. jisc. ac. uk/uploaded_documents/D4L_VLE_report_final. pdf (accessed 8 August 2006)* .

White, S. A. (2004), 'Workflow Patterns with BPMN and UML', *IBM, January*.

Winters, N. & Mor, Y. (forthcoming), 'Dealing with abstraction: case study generalisation as a method for eliciting design patterns', *Computers in Human Behavior, special issue on Design Patterns for Augmenting E-Learning Experiences*. Elsevier Ltd.

Winters, N. & Mor, Y. (2008), 'IDR: a participatory methodology for interdisciplinary design in technology enhanced learning', *Computers and Education* **50**(2), 579-600.

Winters, N.; Mor, Y. & Pratt, D. (forthcoming), The distributed developmental network - d2n: a social configuration to support design pattern generation, *in* Peter Goodyear & Simos Retalis, eds.,'Technology-enhanced learning: Design Patterns and Pattern Languages', Sense Publishers, Rotterdam.

Appendix: The participatory pattern elicitation pattern

http://lp.noe-kaleidoscope.org/workspace/patterns/patterns_from_cases/

Summary: How to elicit design patterns by discussing case studies

The problem / intent

Interdisciplinary communities engaged in producing and using technology for learning need methods for sharing and integrating design knowledge. Design patterns are a powerful mechanism to this end, but how do we support community members in formulating patterns?

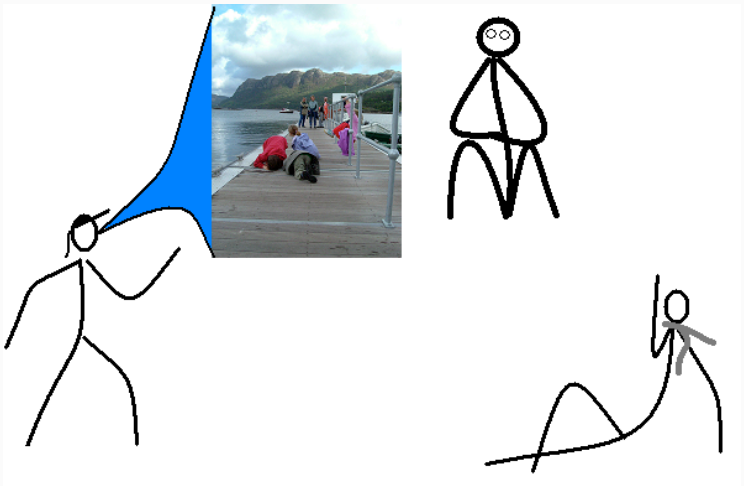
The context

This pattern does not map to the typologies, or rather it addresses all of them. It is a meta-pattern, concerning the problem of generating patterns. The context of a problem is the process of interdisciplinary design and design research in technology-enhanced education. This process involves communities or teams of experts from diverse disciplines: educationalists, technology designers and developers, curriculum designers, researchers in relevant fields, etc. While all these people have valuable design knowledge, derived from their personal experience, this knowledge is tacit: it might have never been formalized, certainly not in the form of patterns.

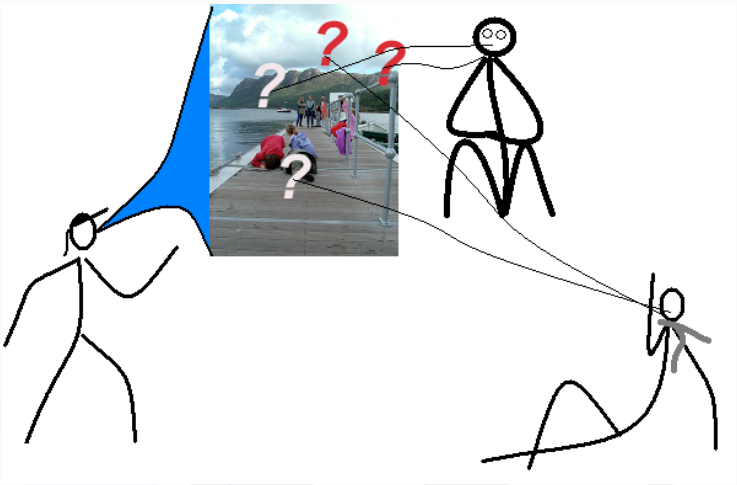
The pattern

The pattern for extracting patterns from cases follows a natural trajectory of abstraction from experience, while structuring and scaffolding it using set tools and procedures.

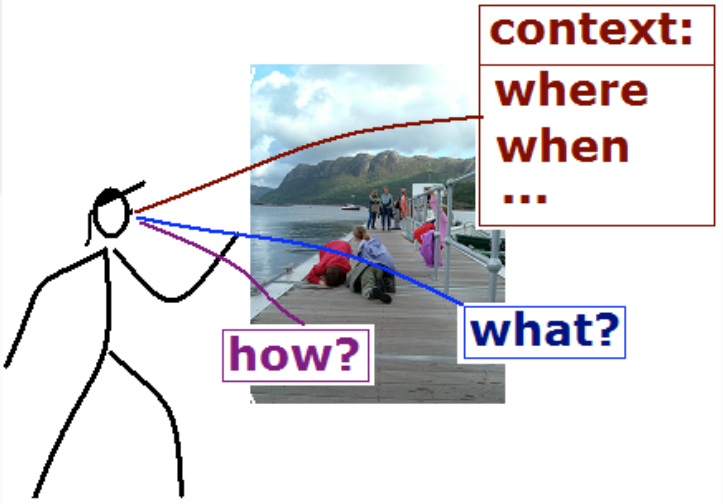
1. Alice describes a case of interest from her experience, noting the design challenge encountered in this situation and how it was dealt with.



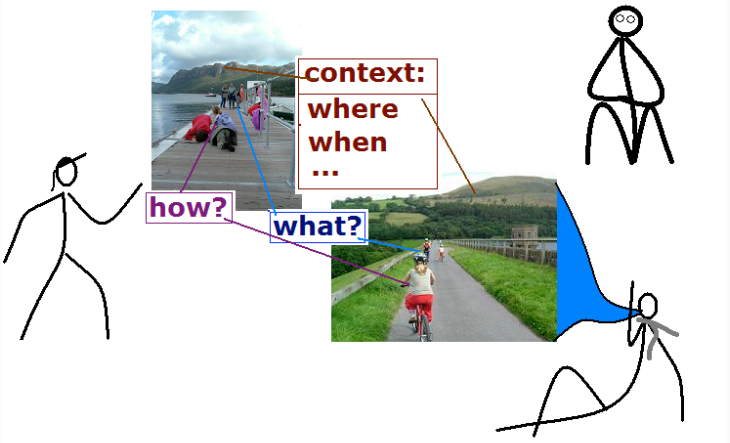
2. Bob and Claire ask Alice to elucidate any ambiguous details.



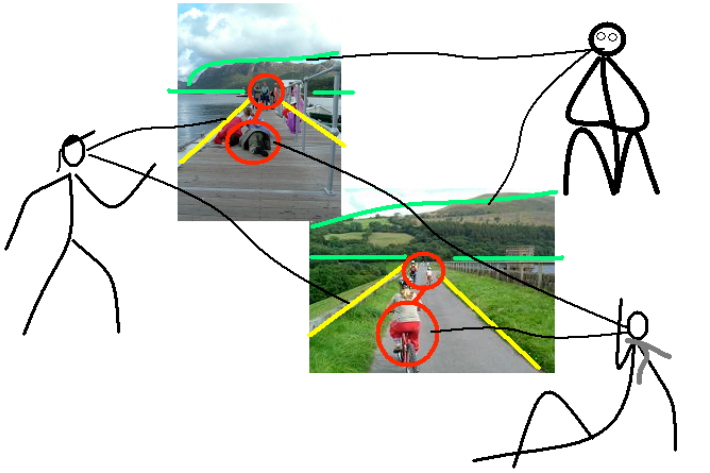
3. Alice elaborates the case context, by mapping it to the typologies.



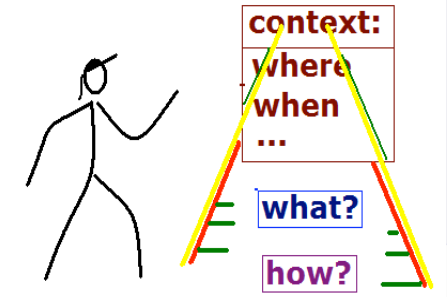
4. Bob notes similarities to the context or problem of a case from his experience, and notes the solution used there. In some cases, Bob's case may be hypothetical - specifying an unsolved problem, and how Alice's solution might be applied by Content morph or Rejigging.

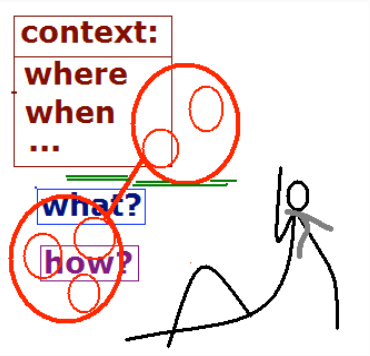


5. Through a comparative discussion, Alice, Bob and Claire identify the salient features which emerge from their cases.



6. Alice and Bob each contribute a seed pattern detailing one of the features identified, and situating it with respect to the context and problem discussed.





Related patterns

Leads to: Design Exploration through Gameplay Design Patterns, Rejigging
Follows: Content morph,
Elaborates: Experimental design

Examples

The GmX Trail .