kaleidoscope

concepts and methods for exploring the future of learning with digital technologies

D40.5.1 (Final)

Final report

Main author : Dave Pratt (UWARWICK)

Nature of the deliverable : Report

Dissemination level : Public

Planned delivery date : December 2006

Prepared for the European Commission, DG INFSO, under contract N°. IST 507838
as a deliverable from WP40
Submitted on 20-12-2006

Summary

This report reviews the methodology and discusses the mia findings of the project.

History

Filename	Status	Release	Changes	Uploaded
D40-05-01-F.pdf	Final	1		20/12/2006
D40-05-01-V1.pdf	Draft	1		11/12/2006

Contributor(s)

Name, First name	Contractor
Mor Yishay	(IOE)
Winters Niall	(IOE)

Deliverable Reference	D40.05.01
Release version	Draft
Submission date	31st December, 2006
Number of pages	39

Deliverable 5:
The Final Report

August 2006

1. Overview.......... 5

2. Re-scoping the Aims and Objectives.......... 7

3. Methodological Processes.......... 9
3.1 Purpose.......... 10
Process.......... 10
3.2 Pattern life-cycle.......... 13
3.3 The tools.......... 14
3.4 Structure of the Language.......... 15
3.5 Trails.......... 15
3.6 Design Patterns as a Framework for Multi-Disciplinary Participatory Design.......... 17

4. Outputs.......... 18
4.1 D40.1.1 Literature Review 18
4.2 D40.2.1 Typological Analysis of Games Development and Game Contexts 19
4.3 D40.3.1 Design Patterns 21
4.4 D40.4.1 International Workshop 32
4.5 D40.4.2 Streaming Video.......... 32

5. Added Value to the KJA.......... 34
5.1 Virtual Doctoral School.......... 34
5.2 Shared Virtual Lab (SVL).......... 34
5.3 The Kaleidoscope Integration Process.......... 35

6. Conclusion.......... 36
6.1 Findings.......... 37
6.2 Recommendations.......... 37
6.3 Final Comment.......... 38

7. Bibliography.......... 39

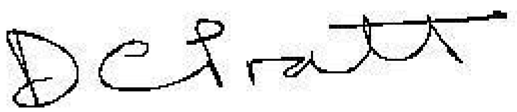
Preface

Over the last few years have witnessed a growing recognition of the educational potential of computer games. However, it is generally agreed that the process of designing and deploying TEL resources generally and games for mathematical learning specifically is a difficult task. The Kaleidoscope project, "Learning patterns for the design and deployment of mathematical games", aims to investigate this problem. We work from the premise that designing and deploying games for mathematical learning requires the assimilation and integration of deep knowledge from diverse domains of expertise including mathematics, games development, software engineering, learning and teaching. We promote the use of a design patterns approach to address this problem.

This deliverable reports on the project by presenting both a connected account of the prior deliverables and also a detailed description of the methodology involved in producing those deliverables. In terms of conducting the future work which this report envisages, the setting out of our methodology is seen by us as very significant. The central deliverable includes reference to a large set of *learning patterns* for use by educators, researchers, practitioners, designers and software developers when designing and deploying TEL-based mathematical games. Our pattern language is suggested as an enabling tool for good practice, by facilitating pattern-specific communication and knowledge sharing between participants. We provide a set of *trails* as a 'way-in' to using the learning pattern language.

We report in this methodology how the project has enabled the synergistic collaboration of what started out as two distinct strands: design and deployment, even to the extent that it is now difficult to identify those strands within the processes and deliverables of the project. The tools and outcomes from the project can be found at:

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



Dave Pratt
Co-director

Niall Winters
Co-director

December, 2006

1. Overview

This JEIRP began with a dipolar structure in which one strand involved the construction of design patterns for mathematical games and the other strand focussed on the deployment of games. Historically, this structure resulted from the merging of two independent project proposals. In practice, each strand has informed the development of the other throughout the duration of the JEIRP to the extent that the two strands have become almost indistinguishable. Evidence of this can be taken from how the construct of a pattern has become the central language whether discussing the design of a game as tested through deployment or the deployment of a game informing its design.

The project website has proved to be a key organising tool in the unification process. The website can be found at:

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

where, as well as the project's deliverables, the reader can locate the tools which facilitated communication, comment and criticism within the project team and from outside contributors and evaluators. The continuous internal conversation, alongside the whole team face-to-face meetings, enabled the unification of the two strands. Indeed, this coming together of the design and deployment strands allowed us to reconsider the aims of the project so that they embraced the relationship between design and deployment of mathematical games.

As a result, in the space of a single year, the project has begun to formally characterise the development practices and deployment of mathematical games by making particular techniques explicit, potentially enabling reuse of practice. By developing a set of reusable patterns, the structure of design and use of games has been captured in forms that may be easily communicated and transferred.

In general, a design pattern is defined as a high-level specification for a method of solving a problem by design. Its particular strength is in highlighting *recurring techniques and solutions* to design problems that are found again and again in real—world application development. Design patterns enable this process of knowledge discovery by specifying the particulars of a problem, and how the designated design instruments can address them. Classically, design patterns have been proposed in a format that consists of the following components (Alexander et al, 1985):

• An introductory paragraph, which sets the *context* for the pattern

• A concise *problem statement*

• The *body* of the problem—it describes the empirical background of the pattern, the evidence for its validity, the range of different ways the pattern can be manifested

• The *solution* that describes the relationships required to solve the stated problem, in the stated context. It is preferable to state the solution in the form of an instruction. A *diagram* may be included here.

• A relationship between this pattern and others.

In setting out to capture the process of designing and using mathematical games, we recognised at the outset the ambitious nature of the enterprise. Clearly within one year such a project can only hope to come to an understanding of the scope of the problem and to put in place an effective methodology and indeed we found it necessary to realign our original aims (Section 2). In Section 3, we discuss in detail the methodology used in this project; indeed, we see a major success of the project to be our identification and creation of tools that are extensible and appear, on the basis of the evidence of this project, to provide an appropriate means to communicate and test the relationship between design and deployment of mathematical games. In this section, we wish to reflect on the scope of the problem.

We will present in Sections 4 and 5 an illustration of the outcomes of the project including some of the design patterns identified. The reader may be surprised to learn that, in the space of one year, we have identified over 100 patterns describing the design and deployment of mathematical games. Numerically, this is an enormous achievement. However, one finding from the project is to recognise that these patterns are structurally related within hierarchical webs of logic. We talk about patterns elaborating and being elaborated by other patterns within that web of relations. In this sense the further one drills down into the specific contexts of application of the pattern the more patterns one can find. Contrarily, the more one abstracts leaving behind the contexts of application the fewer patterns one finds. Indeed, we now see the enterprise as one of mapping that structure of inter-relationships between patterns of design and deployment. There is no question that the project has begun to construct and communicate that map through its website. In a sense, the task continues to be the invention of further constructs, that is to say higher level patterns, which reduce the number of patterns when the designer or the user wishes to access the map at the highest levels of abstraction.

Hence, we would argue that, notwithstanding its extraordinary achievements in the space of one year, the project should be conceptualized as at the first bootstrapping phase of a longer term design-based research enterprise. Over the lifetime of that enterprise, we would anticipate not only the building of tools to enable the design and deployment of games but also the elaboration of a pattern-based language to facilitate communication at the theoretical level. This ongoing work would in our vision be scoped across four broad phases (Table 1):

Table 1: Design Research Phases

Phase 1	Phase 2	Phase 3	Phase 4
Pattern development by interdisciplinary team	Paper description of patterns in a pattern language	Use of these patterns in collaborative, interdisciplinary processes between developers, teachers, students, researchers, and designers in the development of games	Use of resulting games and associated pedagogies in the classroom

In terms of these four phases, we should think of the current project as focussed on Phases 1 and 2, in which the language of patterns is being developed. In developing this basic language of patterns, we have certainly worked with teachers and developers. However, the main effort has been turned towards constructing the tools and putting in place the methods that have enabled us to construct a first draft of that language. If the design-based research were to continue, and we believe the project so far as demonstrated a strong case for extended and in-depth study of design patterns of learning, there would be an increased focus on Phases 3 and 4, resulting iteratively in the continued construction and testing of the patterns and, crucially, the mapping of the structure of those patterns, the grammar of the language as it were.

In Section 6, we will say more about the implications of this work for future research.

2. Re-scoping the Aims and Objectives

The previous section has reflected on how our conceptualisation of the project moved from two distinct strands to our current view of a larger single design-based enterprise, within which this project has successfully established an initial version of the language of patterns that structures the design and deployment of mathematical games and a set of tools that build a methodology for extending that enterprise. This transformation towards a unified project renders the articulation of separate aims and objectives as in the original formulation obsolete. Nevertheless, we have addressed similar aspirations as the original dipolar project envisaged and so below we set out fresh aims and objectives within a single coherent formulation, which now seems more appropriate. In each case, we reflect on the extent to which we have managed to elaborate those aims.

We originally asked (i) on the deployment strand how can games be modified to make them more useful for teaching mathematics and how can the teacher create learning activities around the games (the "wrapper") to make optimal use of the game, and (ii) how we could develop a set of customisable design patterns for mathematical games, promoting best-practice and reuse of proven techniques.

We now see the overall aim of this project as:

To map, through a set of customisable patterns, the relationship between design and deployment of mathematical games, with the longer term aim of promoting the reuse of proven techniques.

We believe that this re-statement of the project aims not only captures the original questions of each strand (design and deployment) in a unified and coherent way but also reflects the re-conceptualisation of the project as discussed above.

In order to achieve this aim, we set out below several objectives, which are essentially cohered versions of the original di-polar objectives (indeed we indicate the root objectives that make up these reformulated objectives), and we briefly comment on the extent to which we have already met these objectives. In this section, we offer the re-scoped aims without explicitly explaining the connection between those aims. Subsequently we will explain the methodology thorough which we began to make those connections. Nevertheless, in the space of this short project, we can only claim to have made a start in understanding the nature of learning patterns that begins to emerge by working in this way. We believe a project of the size of a STREP is now needed to further clarify and elaborate those connections.

Conduct a literature review of the design and use of mathematical games in education.

(Originally separate reviews were envisaged).

In Section 4, we discuss the literature review deliverable. In fact, at this early stage of the project, the design and deployment strands were not fully integrated and this can be identified quite clearly in the structure of the review. Nevertheless, the literature review is extensive and thorough (it is 130 pages long) and we believe that the objective has been met, though no doubt further work could now be done to integrate the two strands to provide a better synthesis that reviews the *relationship* between design and deployment of mathematical games.

Establish a typology of mathematical games for education.
(This objective is unchanged.)
In Section 4, we discuss the deliverable, a typology of mathematical games, which was an important tool in establishing the methodological process of mapping out the relationship between the patterns of design and deployment. Even in the later stages of the project, we have been able to identify omissions and contradictions, which come to light when we try to classify new emerging patterns. We see this inevitable in the nature of design-based research and regard the typology and indeed all of our methodological tools as work-in-progress. Nevertheless, the typology that exists extends across six sub-typologies, namely Mathematical Content, Learning and Instruction, Educational Context, Games, Interface and Interaction, and Software Design, and we believe that the objective has been met, notwithstanding the reservations related to further development as outlined above.

Develop case studies and other descriptions of the relationship between design and deployment by observing, interviewing and reflecting on personal experiences.
(This objective embraces the original design and deployment objective to interrogate our repositories for sets of mathematical games and to identify success factors in the approaches used by teachers through analysing observations and conducting focus groups.)
In Section 3, we discuss case studies and trails, which formed an important part of the methodological process. We observed and interviewed teachers and teacher trainees, both in face-to-face interviews and through the use of focus groups, in order to identify patterns in the use of mathematical games. We also carried out retrospective analyses of the design and deployment of mathematical games in which members of the project team had been previously involved. Inevitably, with one year of a project, the case studies and trails can only hope to reflect a relatively narrow range of experience. Nevertheless, 18 case studies have been posited on the project website. Alongside these case studies, we have constructed 6 trails, which describe the process by which the experiments were carried out to identify the patterns. Given the case studies and trails, we believe that we have met this objective though ongoing research could enrich the descriptions further by identifying further case studies and trails.

Identify the extent to which patterns can capture the general nature, the requirements of the learning environment and modes of use of mathematical games.
(This objective embraces the original design strand objective to identify the extent to which design patterns can capture the general nature and requirements of the learning environment as well as the deployment strand objectives to identify the range of activities used and created by teachers.)
In Section 4, we illustrate the patterns of design and deployment. The identification of patterns became the main focal point of the project, though as the integration of the strands developed, the scope of our notion of "pattern" broadened to embrace fully the relationship between design and deployment. The 54 page delivered document details the patterns identified. However, we recognise that the development of such patterns will not in itself be a useful tool without a comprehensive understanding of the relationship between those patterns, which is the subject matter of the final objective below.

Develop a set of patterns, and the interrelationship between them, that describe the structure of the relationship between design and deployment of games.
(This objective embraces the original design objectives to develop customisable, ready-to-use design patterns and to implement a hierarchy of patterns but is extended to encompass the patterns of deployment.)
The project website sets out in a mindmap document the relationship, as we currently understand it, between the patterns identified. (For an example of a slice taken from the mindmap, see Section 4.3.3.) At its lowest level, the map is likely to contain more information than a designer or practitioner accessing the site could engage with. For this reason the hierarchies built into the map are important so that higher levels of abstraction can be used to pin down the relevant pattern for any specific purpose. Although we believe that this map of patterns is an extraordinary achievement in such a limited project, we also recognise that it represents only the start of a process. It remains the fundamental objective for a longer-term enterprise.

3. Methodological Processes

The first outcome of the project, a review of the literature, highlighted both the educational potential of games and the complex challenges which impede the realization of this potential. We argued that these challenges are dominated by issues pertaining to design knowledge in a broad set of domains. We identified a range of design approaches to educational research in the field of technology enhanced mathematical education, and concluded by suggesting that the design pattern methodology may offer an answer to some of the open issues in this field.

The next effort was aimed at producing a set of typologies, which provide a structured lexicon derived from six different knowledge domains which we see as central to the questions at hand; in effect they act as a resource in the form of a content-based relational map, for classifying the different aspects of design knowledge required in the process of the design, development and application of mathematical games for learning.

These typologies were developed in tandem with several illuminating case studies.

These typologies and case studies inform our pattern language. Most patterns are distilled from the case studies, and the typologies play a crucial role in defining their context and in providing the conceptual vocabulary for their descriptions. These patterns emerge from the case studies and typologies as reflections on how to design and how to deploy games, and crucially on the relationships between those two interconnected activities.

We do not claim to offer a comprehensive set of patterns, but we do strive to construct a coherent language, which has few holes and many open ends. Our aim is it to address issues across a broad range of aspects pertaining to the process of designing, implementing and deploying games for mathematical learning.

The learning patterns we developed attempt to strike a balance between problem solving and being feature specific. Some patterns address the process of game development and in doing so emerged from problems we were trying to overcome. As such, they can be viewed as problem solving patterns. Others are directly concerned with particular game features and interaction issues and are considered feature specific. However, in describing the patterns we use the generic term ‘problem’ to encompass both perspectives. For example, to address a particular problem the designer may add a specific feature.

Our patterns are distributed in two dimensions, breadth and depth:

1. Depth: level of detail, where one pattern elaborates a higher level one, or is used by it as a component.
2. Breadth: modes of abstraction, moving from general methodologies through the dynamics of the game design and deployment process, and down to the specific patterns of game structure.

In this sense, learning patterns have a broader scope than that originally envisaged for design patterns, and in response we have slightly varied the components that make up a pattern (see for example Section 4.3, though without doubt we have remained true to the original spirit.

3.1 Purpose

Our pattern language and its associated interactive tools are presented as resources to be used by researchers and practitioners in several ways. As an analytical asset, design patterns are a means of making visible implicit design decisions. Researchers and designers can reflect on their own work by mapping it to patterns in our language, or by extending the language to account for aspects we do not cover. Identifying the underlying patterns can help understand the strengths and weaknesses of existing games and the ways in which they are used. Once a pattern has been mapped to a case under observation, the context noted in the pattern can be compared to the details of the actual case, and conflicts can be discussed. On the other hand, the related patterns should be explored, to identify possible extensions and enhancements. As a design aid, practitioners from various fields who are involved in game design and deployment can consult the patterns in different stages of their process, and choose those which address the particular questions they are confronted with. Some of our patterns address the flow of the process as a whole, some address specific phases - such as 'bootstrapping' design, and some offer concrete structural elements which can by used as building blocks. It is important to note that patterns are not cookbooks. They do not devolve responsibility form either party, but only help designers and practitioners to communicate and understand the scope of the issues.

However, the most important facet of the pattern language is its potential as a framework for discussing and collaboratively refining design. In fact, this is precisely why it is called a pattern *language*, and not collection or set. This language grew through its use in various assemblies of designers, researchers and educators. Our workshops are structured around the language and the tools, and have used them successfully to sustain effective communication among experts from varied backgrounds. In this function, our pattern language should be seen as a starting point, an example from which each community will derive and develop its own language.

Process

The process of creating, or 'distilling' a pattern began with reflection on expert knowledge represented as a case study of good practice. The pattern authors identified a single element of design which contributed to the success of this case study. This element was phrased in a manner which detached it from the single example, but avoided over-abstraction. In the words of one of the project members, it was a "situated abstraction done by an expert". The pattern was carefully named: names needed to be descriptive, concise and attractive. Its details were then moulded into the pattern template (described below). Using a good pattern in the wrong context was a common pattern of failure. Once the pattern had been described, it was mapped to other case studies and to other patterns in the language. By comparing to similar case studies we were able to refine the pattern and identify its critical features. This led to the need to define new patterns - as a special case or as generalization. At the same time, we classified new patterns using the hierarchical structure of the language and looked for related patterns which were already in our collection. The *Three* C*'s* trail below provides an illustrative example of this process.

The pattern language was developed iteratively and collaboratively by the project team, in dialogue with a wider community of designers, researchers and teachers. Due to the distributed nature of the team, the availability of on-line tools played an important role in our ability to conduct this process effectively. These tools were developed in parallel with the language, as our understanding of the process in which we were engaged evolved.

Our first patterns were drafted in early April, while working on the case studies and typologies. The purpose of these drafts was mainly to direct and inform the development of the typologies so that these would prove useful when the main effort would shift to the patterns. These preliminary drafts also provided us with the basis for a pattern template. The pattern drafts, the template and the emerging structure of the pattern section of the web site were discussed at the project assembly in Genoa in late April. After the discussion we split into cross-site groups and, as an exercise, each group produced one or more patterns. Following the assembly, we reviewed the process of pattern development and the tools needed to support it.

The pattern language and tools were developed intensively over the next two months, leading to the FNG workshop in late June. That workshop brought together over 20 researchers, game designers and educators, and established deep discussions of game and educational design using the pattern language as a framework of communication. Delegates contributed case studies from their experience, and these were processed to a new set of patterns which extended the existing language.

Our pattern language now consists of over 100 patterns. Each pattern we developed followed a pre-defined template structure. This structure is as follows:

- It begins with a short name for the pattern. Choosing a good name is important because it makes the core idea of the pattern explicit in a compact and easy-to-remember manner for users. Additionally, it is what the user sees when browsing the entire pattern language and therefore should not be underestimated. Next, a one sentence summary of the problem is provided followed by a short explanatory paragraph specifying the nature of the problem and the motivation for using this pattern.
- Following on from this, the context of the problem is specified. This begins with a paragraph specifying how the pattern was developed. The context is further described using one or more pre-defined typologies from a list of six, using the typology tool we developed (http://lp.noe-kaleidoscope.org/outcomes/typologies/). In turn, each typology has nodes that more fully detail the context and are topics that the pattern addresses. For example, the software design typology has nodes including programming language and development methodology. The typologies were chosen through an iterative process of construction, testing, negotiation and refinement. We initiated this process through a brainstorming session conducted during a project meeting. This provided the initial outline and candidate typologies.

These were whittled down with initial drafts published online to be scrutinised by team members. Using an online discussion mechanism, we queried each other for clarifications and illuminated possible gaps and overlaps. Next the potential capacity and use of the typologies was explored and tested by drafting small case studies and initial learning patterns from them. The rationale for this approach was our belief that for the typologies to be a robust tool, they needed to be refined through productive use. This resulted in the following form:

1. *Subject content*: used to identify the subject area (in our case mathematics) the game will address. This is often, but not always, based on national curricula.
2. *Learning and instruction*: specifies the theoretical and pedagogical underpinning game development.
3. *Educational context*: defines the culturally situated and socially shared activity of mediated intervention in teaching and learning.
4. *Games*: provides a starting point for design and evaluation processes from an artefact perspective.
5. *Interface and interaction*: characterizes the nature of the interaction with the game.
6. *Software design*: specifies the process of software development.

- Next the practice encapsulated by the pattern itself is detailed. This takes the form of a sequence of numbered steps for users to follow in order to operationalise the pattern.

- After this the relationships between patterns are listed. This is important as it defines the networked structure of the pattern language. We define four types of relationships: *Elaborates, Elaborated by, Follows* and *Leads to*. 'Elaborates' defines a 'is a type of' relationship (see Section 4.3.3). For example, 'poodle' elaborates 'dog' (or from our learning patterns 'event-driven iterative design' elaborates 'iterative design'). This implies that whenever 'poodle' is present it automatically means 'dog' is present. However, each pattern can be elaborated by more than one pattern and usually is. 'Elaborated by' implies a more specific instance of the pattern, for example 'dog' is elaborated by 'poodle'. In this way we build up an easily browsable pattern hierarchy, with the more abstract patterns positioned at a high level and the more specific ones lower down. As such no pattern exists on its own – it is supported by others above and below. 'Follows' and 'Leads to' indicate semantic flow which is not captured by the hierarchy of the language. These could be associative links, in the sense of 'if you found this pattern useful, you may want to consider this one' or 'this pattern will be better understood if you first read that one'. Each pattern also belongs to one category: methodology, design, deployment, evaluation and development. Moreover, a pattern can elaborate another from any category but can only be listed in only one.

We view our language of learning patterns as a dynamically evolving resource, and this vision is reflected in the structure of the language and in the tools which support it. Patterns are classified as having one of four states: 'seed', 'alpha', 'beta' and 'release' (see Section 3.2 for a definition of these states). The pattern language is organized hierarchically. The top layers of this hierarchy are abstract categories of patterns, while the lower ranks are concrete patterns and sub-patterns. The details of the structure are described in the text of the top-level node.

High-level category nodes are a little different: they take form of a free-text description of the category's semantics, and a list of the main patterns and sub-categories in it. While the pattern template provides a soft scaffolding: it suggests a structure, but does not impose it. However, basic meta-data is consistently attached to each pattern, either automatically or by its authors.

3.2 Pattern life-cycle

Our Learning Patterns are rooted in personal experience and refined through interdisciplinary discussion. We apply the open source motto ‘release early, release often’: patterns are placed in the public domain from the first instance of their conception. Nevertheless, they need to go through several iterations of refinement before they are suitable for general consumption. To this end, we defined 4 phases each pattern needs to pass through: ‘seed’, ‘alpha’, ‘beta’ and ‘release’.

Patterns often emerge in the course of a discussion between experts, e.g. at a workshop, or in the process of refining another pattern. When this happens, they are immediately noted in a minimal form, typically a name and a sentence or two of description. Such a pattern is defined as a seed: it is legible only to the person or group who entered it. The purpose of listing it at this state is to ensure it does not get lost. We have seen too many incidents where valuable insights are expressed in a passionate design discussion among experts but are then lost due to the transient nature of the discussion. The seed phase of patterns is intended to address this issue. Good ideas are logged as they arise, but with a minimal impact on the flow of conversation. These ideas can then be refined at a later time by revisiting the seed patterns and adding detail.

Since a seed pattern is little more than a note the author takes for herself, it is obvious that its author needs to elaborate it before it can be shared with others. The alpha phase is aimed at this purpose. In this phase, the pattern’s original author fills in the pattern components according to the template: she describes the problem addressed or the intent it serves, identifies the context in which it is applicable, adds the details of its structure and situates it in the hierarchy of the pattern language. Once the pattern has reached a state where others can review it, it is published within the community.

When a pattern is reviewed by other experts, they relate it to their experience and to other patterns they are aware of. As a consequence, they will ask for clarifications or offer refinements. They will identify lateral links with other patterns and see the pattern in other case studies which the original author was not familiar with. This collaborative process should promote patterns to a mature, coherent and robust state. Such patterns will be classified as being in beta state.

Once a pattern reaches beta state, it should be useful for the broader community, i.e. anyone from outside the development team who is interested in the questions this pattern refers to. At this phase, the pattern is exposed publicly and slowly refined based on public feedback. Thus, a pattern may remain in this phase for several months before it is finally upgraded to release state.

Such an open iterative process raises issues of resource management and prioritization: how does one choose where to invest? How do we divide our attention between upgrading existing patterns and introducing new ones? How do we decide when to elaborate a seed pattern and when to refine an alpha one? And how to we focus on a subset within a category? Our response to these issues was to introduce a ranking scheme, by which any team member can vote for the patterns she finds more useful. New patterns emerge all the time, as an inevitable consequence of elaborating existing ones or analysing case studies. Authors are expected to lead their patterns to alpha state. Once they do so, the cumulative ranking they get defines the priority by which they will be developed to beta state. At this state, the dynamics of public feedback will define the schedule for the patterns' migration to release state.

3.3 The tools

Alongside the development of the pattern language, we have developed a set of interactive tools to support it. The range of tools set out below can be examined at the project website (Figure 1).

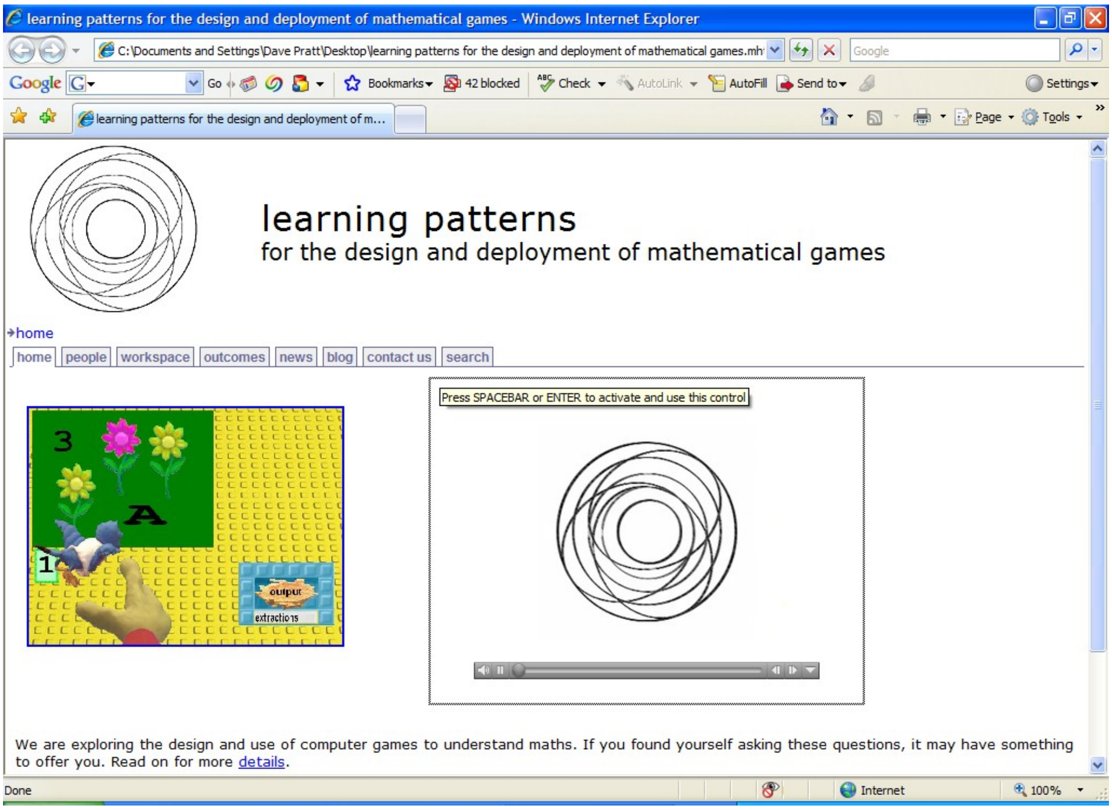


Figure 1: The Project Website

The primary functions of these tools are to allow us to efficiently manage the pattern language, and at the same time make it easy to use by any interested reader. These tools provide various methods of browsing, reading, editing and organizing patterns.

The pattern browser is the central tool in our system. It provides several modes for viewing the patterns, as well as entry points to tools for creating new patterns and structuring the language. All patterns are listed in a database, and can be viewed in table mode and sorted by various keys. The hierarchical structure of the language is represented in a FreeMind map, which can be viewed and used as a navigation scheme for accessing pattern pages.

Patterns are edited using an on-line rich text editor. This editor is based on an open source tool with slight modifications and enhancements. A pattern page begins with a header which displays an expanded view of the meta-data listed in the browser index view. This header is flanked with three links: 'edit info' allows authorized authors to edit the meta-data fields (apart from dates and rank), 'edit page' allows the author to edit the pattern content, and 'publish' makes these changes public. Pattern pages are generated from a template, which scaffolds the author to use a common structure. This structure includes a concise statement of the pattern's intent or the problem it addresses, a detailed delineation of its context based on the typologies, a description of the pattern itself, its relations to other patterns, additional notes and examples. The notes will typically refer to underlying educational research. The examples point to the relevant case studies in our collection.

3.4 Structure of the Language

Our patterns are grouped under three main headings:

- Methodology patterns describe general frameworks of design, development and deployment.
- Process patterns describe specific techniques used.
- Structure patterns describe the details of a design element used identified in one or more case studies.

The Methodology collection provides theoretical grounding and general frameworks for game design, development and deployment in educational contexts. We focus on three common practices for producing games in educational research, and point to several Related Knowledge Collections. The addressed practices are interpreted in terms of the specific context of games in mathematics education.

The Process collection holds patterns related to the processes of design, development, deployment and evaluation of educational games in mathematics. They describe specific techniques used in these processes. Most of these techniques are useful regardless of the methodology, but some are derived from a specific framework. This collection of patterns addresses the main processes of the whole life-cycle of an educational game in mathematics, from the Bootstrap of the Design, to the Evaluation, passing through Development and Deployment.

The Structure collection focuses on patterns for the design (noun) of particular game elements. These are the closest to the common perception of design patterns in software engineering (collection). They are derived directly from the analysis of the team’s past experiences.

3.5 Trails

Paradoxically, often as more expert knowledge is embedded in a pattern languages it becomes less accessible to novices. As a pattern language grows richer and more intricate it becomes the private language of the community which created it. Novices do not know where to start and how to penetrate it, because the structure of the language does not expose the path along which it evolved. In an attempt to address this issue, we have added a tool called ‘Trails’. A trail is an informal illustrative account of how patterns were derived or how they might be used. It is not presented as hard data or detailed analysis, but rather as an aid for the casual reader.

Trails are built using the following procedure:

Using the pattern map, choose the top-level category you are addressing. These are divided into methodology, design, development, deployment and evaluation.

With the category, search through the patterns to find one that you feel best suits your problem. It is also useful to read the online discussions, as they provide a background to the process of how the pattern was developed.

If the pattern is suitable, move onto the 'Leads to' pattern(s) and determine how they can help you with a more focused part of your game development or deployment. If the pattern is not suitable, move to the 'Follows' pattern(s) to investigate your problem at a more abstract level.

Two example trails are provided below *The three C's* and *Beginning the design process*. Underlined terms refer to case studies or typologies; highlighted terms refer to patterns in our language.

3.5.1 The three C's: Construct, communicate, collaborate

The WebReports system is a web-based collaboration platform developed by the WebLabs project. It was conceived as a tool for teacher-led communication between remote groups. The unique feature we designed for was Objects to talk with: allowing users to embed games and game-elements that they had constructed in the conversational medium.

The use of an Iterative design methodology allowed us to react to experimental feedback and adapt the system to a broader use. The first issue we identified was the reluctance of participants to express themselves publicly in an unfamiliar medium. Our remedy for this issue was to Start from self: redesign both the tool and the activity so that users will see clear individual Purpose and utility in writing reports before they are asked to make them public. This required us to allow for Controlled exposure: gradual shift from private to public spaces.

Repositioning our tool for individual and collaborative use throughout the activities, rather than in highly-orchestrated milestones, led us to reconsider the functionality of our tool. We realized that it can be extended to new uses, such as Active worksheet (incorporating Task in a box elements). On the other hand, we identified a need for Visualised social dynamics in order to help us in Sustaining interaction. This was supported, for example by introducing a League chart, and by acknowledging and leveraging Audience awareness.

We identified a tension between the desire to direct learners to formal and structural representation of knowledge and their vernacular forms of expression. This led us to acknowledge the need for Narrative spaces within a knowledge-building system. We addressed this tension by providing report templates based on the Soft scaffolding pattern: a scaffold which is suggestive rather than restrictive. In our case, it meant that the authors or reports could use the template as a starting point, but from there on had full power to overwrite and adjust the structure to their needs.

3.5.2 Beginning the design process

Begin the design process by understanding each methodology available to you. In this pattern collection, these are iterative design, participatory design, experimental design and Related Knowledge Collections.

If you are developing a mathematical game, you will want to bootstrap the design process by understanding where each participant in the process is coming from. An approach, captured by the Knowledge-driven design pattern allows each participant to use their typology to mediate discussion regarding what the overall design needs to address. (The typologies are: mathematical content, learning and instruction, educational context, games, interface and interaction and software design). Alternatively, you could use event-driven iterative design to trace how the different participants interact with each other, and in doing so, construct a common overview of the design process, as well as supporting prioritisation. It is also important to think about concept development so as to embed mathematical concepts into the game.

At this stage, you should have an initial specification. If one has an existing specification, which can be modified, then follow the Metamorphosis pattern.

It is now possible to work on gameplay design. Structurally going from a loose idea of the intended gameplay in a game design to a detailed description or specification can be difficult to do. The design exploration through gameplay design patterns provides one technique to work from an initial set of gameplay design goals, described through gameplay design patterns, to a more detailed description of gameplay.

You are ready to develop an initial prototype. First boundary prototype helps in bounding this process, delineating the scope and depth of the first game prototype developed.

Once this first prototype version is developed, it is ready to be evaluated against the original specification using the remap pattern and to be tested using play-test-use-eval.

3.6 Design Patterns as a Framework for Multi-Disciplinary Participatory Design

A major research challenge is to communicate the potential of tools developed in technology-oriented research to the pedagogy and epistemology research communities, and vice-versa. Design patterns have the potential to bridge between these disparate research and practice communities, and allow each one to enjoy the fruits of the other's efforts. In order to materialize this potential, pattern languages need to avoid jargon, and at the same time make space for higher theoretical discussion. They should be based on a theoretical layer concerning pedagogy and epistemology and consider the learning context.

Dearden et al (2002a; 2002b) point out the strong ideological and methodological parallels between Alexander's original vision of pattern language and the paradigm of participatory design. Pattern languages were conceived as a means of making expert knowledge accessible to naive planners, and enable educated and informed designers to work with naive users in collaboration. By contrast, in practice many pattern languages have taken a highly specialized form, and have become part of a professional jargon. As an alternative, Dearden et al propose the 'facilitation' model developed by Alexander et al (1985) in the Mexicali project. In that project, an 'Architect-builder' worked with a family to enable them to design and build their own house. Very significantly, the pattern language was shared by the designer and the family, and used to present and discuss design problems and solutions. The family could refer to the pattern even when choosing an alternative design.

Participatory design is one of the most exciting and challenging paradigms to emerge in educational research over the last decade. Participatory design is "a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer-based activities" (Muller and Kuhn, 2002). From this perspective, Béguin (2003) points at the close relationship between design and learning. He suggests that effective design should be constructed as a process of mutual learning involving users and designers and argues that the products only reach their final form through use. This should be reflected in an iterative design process which allows the users and designers to collaboratively shape their concept of the product and its actual form simultaneously. Such an approach, if sometimes not explicitly stated in these terms, has led to the emergence of methodologies, which utilize the participatory design of tools and artefacts as a central element in the learning process. For more, we refer the reader to the insightful reviews by Druin (2002) and Nesset and Large (2004), and the recent work by Kaptelinin, Danielsson and Hedestig (Kaptelinin et al, 2004; Danielsson 2004).

The Learning Patterns project has made a modest attempt to foster communication between technology-oriented research and the pedagogy/epistemology research communities, in the form of a series of workshops positioned as interactive, hands-on meetings of researchers, developers and educators. Workshops were initiated by short presentations from participants and organisers. After that, we split into small groups of participants from mixed backgrounds. The goal is to have participants examine critically the process of distilling design patterns as an enabling tool for communication and knowledge sharing. Groupwork was divided into two main sessions: brainstorming and hands-on experimentation. In the brainstorming session, each group provides cases of their design and development processes. Each group then works these into a typology, mapping out major issues of interest and concern to them. This is followed by a hands-on experimentation session which concentrates on the development of a small set of design patterns, which participants feel would help them in their own practices. The day ends with an assembly session where each group will provide feedback on what they achieved and present their patterns. Ample time is allocated for inter-group discussion and sharing of ideas. Overall, the guiding factors are creativity, interaction and discussion. Workshops were supported by a web site, where the outcomes were published and participants have the opportunity to further develop designs, products and connections established on the day.

4. Outputs

This section focussed on the deliverables promised as part of the project's contractual obligations. The remaining obligations relate to the various contributions towards the integrating processes and backbone of the KJA. These will be discussed in Section 5. Apart from this deliverable, the Final Report, the outputs in the form of deliverables have been:

D40.1.1 Literature review

D40.2.1 Typological analysis of Games Development and Game Contexts

D40.3.1 Design Patterns

D40.4.1 International workshop

D40.4.2 Streaming video

We shall address each of these in turn.

4.1 D40.1.1 Literature Review

The Literature Review was delivered in Month 26. All partners contributed towards the final draft.

It was intended as an introduction to the issues that arise when trying to capture the process of designing and developing mathematical games. It offered a perspective on the range of approaches available. Design patterns were suggested as an enabling tool for good practice, by facilitating pattern-specific communication and knowledge sharing between participants. These patterns were termed learning patterns.

At this point in time, the two strands of the project were not fully integrated; the extent of the subsequent integration is in our view testimony to the success of the project. On the design side, we provided a detailed account of the development of mathematical games and the wide range of design approaches taken to address this issue. We discussed the benefits of the patterns approach generally and detailed the pedagogical facets of software design patterns, the extension and adaptation of game design patterns and the relationship between design patterns and didactic functionalities. A significant part of the literature review was devoted to design approaches in education. In particular, we detailed: participatory design (including with children), game design, game design as learning (with a deep focus on mathematical games), and design-based research. On the deployment side, the review concludes that education in general could possibly be improved by adopting some of the principles of gaming. The increasing likelihood of teachers being gamers themselves may impact positively on students' experiences, perhaps giving more emphasis to games as collaborative in nature. The review points also to the relationship between games playing and gender as well as ethnicity. One interesting result of the review is the recognition that there are discrepancies between the use of mathematics games and games in other subjects.

4.2 D40.2.1 Typological Analysis of Games Development and Game Contexts ⇨

The Typology document was delivered in Month 28. All partners contributed towards the final draft.

The typologies are a resource for classifying the different aspects of design knowledge required in the process of the design, development and application of mathematical games for learning. These aspects were Mathematical Content, Learning and Instruction, Educational Context, Games, Interface and Interaction, and Software Design. Each aspect was analysed to generate a content-based relational map.

4.2.1 An illustrative typology

It is outside the scope of this deliverable to detail all the typologies developed. The reader should refer to the main deliverable itself for such detail. However, in terms of describing the overall final outcomes, we regard it as useful for the reader to gain a sense of what a typology looks like, even if they do not wish to examine the detail of each typology. To that end, we offer the following example; the content of mathematics in schools yielded the following typology:

Table 2: Typology of School Mathematics

- -- mathematical content
    - -- content domain
        - -- number and algebra
            - -- numbers and the number system
                - integers
                - powers and roots
                - fractions
                - decimals
                - percentages

                - ratio
                - probability
            - -- calculations
                - number operations and the relationships between them
                - mental methods
                - written methods
                - calculator methods
                - numerical methods
            - solving numerical problems
            - -- equations, formulae and identities
                - use of symbols
                - index notation
                - equations
                - linear equations
                - formulae
                - direct and inverse proportion
                - simultaneous linear equations
                - quadratic equations
                - simultaneous linear and quadratic equations
                - numerical methods
            - -- sequences, functions and graphs
                - sequences
                - sequences and series
                - graphs of linear equations
                - interpreting graphical information
                - quadratic equations
                - algebra and functions
                - exponentials and logarithms
                - other functions
                - transformation of functions
                - loci
        - -- shape, space and measures
            - -- geometrical reasoning
                - properties of triangles and other rectilinear shapes
                - properties of circles
            - -- transformations and co-ordinates
                - specifying transformations
                - properties of transformations
                - co-ordinates
                - co-ordinate geometry in the (x,y) plane
                - vectors
            - -- measures and construction
                - measures
                - construction
                - mensuration
                - loci
        - -- calculus
            - differentiation
            - integration
    - -- target audience
        - pre-school
        - 5-11
        - 11-14
        - 14-16
        - 16-19
        - post-compulsory education
        - adult education
        - specific learning needs
    - -- Skill Domain
        - + Reasoning

- Argumentation
- Factual
- Computational
- Intuition
- Problem solving
- Spatial perception
- Hypothesis - testing
- Substantiation
- Apply basic mathematical concepts
- communicating
- + handling data

The above listing can also be viewed on the website as a FreeMind map.

Inevitably, it is possible to argue that this typology is not exhaustive, though it does seem to reflect the content of mathematics as it is typically presented in schools across Europe, with of course some differences in detail. Neither could it be argued that this typology is unambiguous. As we try to relate newly emergent patterns to a relevant area of mathematics, we often have some difficulties of interpretation. Sometimes this is because the pattern relates more to process than content and the typology tends to reflect content areas less ambiguously than process areas. Sometimes the problems arise because it is in the nature of mathematics that the powerful ideas are inherently related to a range of areas of the typology. Some of these ambiguities and gaps could be addressed through further work. In the end though, it is probably the role of a typology to stimulate such discussion rather than to enable unproblematic classification. With respect to our own work, we have found the content typology helpful in the respect.

These comments apply more generally to the other typologies in so far as they do not represent finished articles, complete in their precision and lack of ambiguity but rather they are helpful stimulants for the development of design patterns. The typologies are themselves open to development beyond the current project.

4.3 D40.3.1 Design Patterns

This deliverable was delivered in Month 32. All partners contributed towards the final draft.

The deliverable consists of over 100 design patterns inter-linked in a complex hierarchy or map, depicting the logical structure of the relationship between those patterns and so between design and deployment of mathematical games. As befits a lively dynamic project, now conceptualised as the early stages of a longer term design-based study, these design patterns are at various stages of development.

The most progressed patterns are referred to as "release" stage. This does not mean the pattern is complete but that it is in a state ready for public critique. When it has gone through this process, it can be moved on to Phases 3 and 4 as described in Table 1 . In this way, we would intend that patterns that have reached such a stage would be the initial candidates for testing with teachers and developers who do not currently form part of the project. We would expect further modifications and developments in these patterns in the future and also in how those patterns connect in the overall structure or map.

Other patterns are categorised as "beta", "alpha" and "seed", according to how close to release they have reached (see also Section 3.2). At the seed level, for example, a pattern represents little more than an idea or an observation that has not yet been formally defined. The table below indicates the criteria that enable a pattern to be categorised in these ways:

Table 3: Definition of states of patterns

Category	**Characterisation**
Seed	An initial idea for a pattern
Alpha	A pattern with sections unfilled
Beta	Completed pattern to be made available for public review. Once a pattern reaches beta state, it could be of value to the broader community i.e. anyone from outside the development team who is interested in the questions this pattern refers to.
Release	Going through public review At this phase, the pattern is exposed publicly and slowly refined based on public feedback. Thus, a pattern may remain in this phase for several months.

It is beyond the scope of this Final Report to set out the full set of design patterns. The reader who wishes to examine the patterns at this level of detail is encouraged to engage with the project website itself. For the benefit of the reader who merely wishes to understand better the nature of these learning patterns, we set out below a single example of a release level pattern and an alpha level pattern.

4.3.1 Illustrative release patterns

We present here three example patterns in ‘release’ state: *Drill & Argue*: a deployment pattern, *Content morph*: a development process pattern, and *Guess my X*: a game-structure pattern. These patterns has achieved release status in that the problem/intent is fully explained, that multiple taxa are added to the fields of the metadata table, each one drawn from the typologies developed earlier in the project.

These patterns have been in beta state for over four months, during which they have been reviewed in several public presentations. The input from these reviews helped us refine them to their current mature state. (For the sake of clarity, background colours are used to delimit each pattern definition.)

Drill & Argue: an example deployment pattern

Drill & Argue		Category:	Deployment
Created	Yishay Mor, 26 May, 2006	Modified:	20 November, 2006
Status	release	Rank	5
Summary	using a drill & practice game to facilitate argumentation		

The problem/Intent

How to engage a class in a mathematical discussion?

Finding good question items to provoke discussion is time consuming. Devising a good graphical representation for them is even worse. Many students will not engage in such a discussion in a traditional setting: a straightforward debate of mathematical questions is considered boring. Typically, when a discussion ensues it will be dominated by the more confident students, while others might be afraid to be embarrassed.

The context

Mathematical content	content domain: Appropriate for factual knowledge, basic skills, and calculations. Less useful for meta-cognitive and abstract skills (e.g., proof). skill domain: Factual, Computational, Argumentation. target audience: any.
Learning and instruction	Can complement instructional or constructivist activities. Stresses argumentation and socio-mathematical norms. Mathematical content: embedded in toy. Metaphour: participatory.
Educational context	Whole class or group discussion during regular maths lesson. Warm up for a new topic or recap of a topic before a new one is introduced. Could also be used as a time-filler at the end of a lesson. Requires Electronic whiteboard / projector, connected to computer with games installed or access to on-line games.

The pattern

- **Teacher** finds a drill and practice game which relates to the topic under study.
- **Teacher** presents the game on the whiteboard and explains that it will be played collaboratively by the class as a team.
- For each problem presented by the game, **Students** raise hand to offer answers.
- **Teacher** selects one to propose response.
- **Students** raise hand if they object.
- **Teacher** asks **objectors** to argue for their objection.
- **Students** vote for solution.
- Repeat.

Notes

The pattern relies on the sensitivity with which the teacher manages the whole class situation, avoiding the control the debate by the most vociferous children. Most experienced teachers are good at this. Less experienced teachers should take note, and take care to distribute their selection of responders and objectors.

It is suitable for some skills and topics, and much less so for others – such as proof.

While using a game takes the strain off the teacher, the task of choosing an appropriate game is critical. The questions being offered by the software need to match the current classroom topic and level, and be appropriate to stimulate the debate and discussion. 'Millionaire' style programmes are good in this respect, setting a good context for discussion and offering a choice from four answers. Time-constrained games will not be suitable for this pattern. The main purpose of this game is to provoke discussion and argumentation. However, if the game times responses, there is an incentive to reach consensus quickly, which works against allowing time for discussion.

The objection and voting mechanism allows the teacher to monitor gaps in class knowledge. Where there are no objections, it is reasonable to assume that the class is confident in its answer to this particular question, and a discussion is redundant.

While we see this as a collaborative learning model, it is a whole class, short term collaboration – wider but shallower than, e.g. project work in groups.

Allowing the 'computer' to set the questions distances the teacher from the control of the questioning, and allows a different, freer, dynamic within the classroom. The teacher is closer to a game facilitator then an opponent. A wrong answer is less embarrassing, because it is judged by the computer – not by the teacher.

Related patterns

Follows: Problem of the month

Elaborates: Scenario

Examples

Case study: Mathionaire in Christopher Hatton

The following games were successfully used in the context of this pattern:

http://www.subtangent.com/maths/mathionaire.php
http://www.subtangent.com/maths/higher-lower.php

Games developed by Duncan Keith, an English maths teacher.

Content morph: An example process pattern

Content morph		Category:	Bootstrap
Created	Michele Cerulli, 26 May, 2006	Modified:	20 November, 2006
Status	release	Rank	4
Summary	Keeping the structure of the game and the educational principles, but changing the addressed mathematical content.		

The problem / intent

Given an existing game, how can a new game be developed that exploits similar educational principles but address different mathematical content?

The context

The pattern emerged from a context where a group of educational researchers were cooperatively designing and deploying educational software and games. The objective is that, once a designed game resulted to be fruitful from an educational point of view, the group of researchers tries to extract the key ideas of the game in order to re-use them to design other fruitful games addressing different mathematical contents. In order to do so it is needed to identify what are the elements/characteristics of the original game that can (or cannot) be adapted to the new mathematical content, and how this can be done. As different mathematical content may need to be addressed in different ways, this pattern is fundamental for avoiding a loss of the effective characteristics of the original game, and for avoiding naïve adaptation of such characteristics.

Mathematical content	Any, but different from the existing game.
Learning and instruction	Modalities of employment, Approaches and theories of reference, Game.
Educational context	play, type of learning activity.
Games	game as activity, game as social function, games as genre.

The pattern

1. Identify the new mathematical content, and compare it with the mathematical content addressed by the existing game. Focus on what immediately changes (for example, the age group).
2. Identify the main educational principles underlying the existing game (for example, the modalities of employment).
3. Investigate how the change of mathematical content will affect the application of the existing educational principles in the new game.
4. Determine the characteristics of the educational context that are considered to be necessary for the existing game to be successfully deployed. Can they also support deployment of the new game?
5. Identify the key game elements and interactions that are to be inherited by the new game, and which must to be altered to support the new content.

Wherever possible the designers of the existing game and the designers of the new game work towards communicating core game characteristics, stating how they can be inherited and modified in the new game design.

Related patterns

Leads to: Rejigging.

Follows: Experimental design.

Elaborates: Metamorphosis.

Examples

- The Guess my Garden game was derived from Guess my Robot.

Guess my X: An example game-structure pattern

Guess my X		Category:	Game
Created	Yishay Mor, 06 June, 2006	Modified:	20 November, 2006
Status	release	Rank	5
Summary	I created something, you have to figure out how.		

The problem / intent

Initiating and sustaining a mathematical discussion in a learning community is vital to the establishment of socio-mathematical norms and to the collaborative construction of knowledge in the community. This goal is always difficult to achieve, especially in geographically distributed communities. We address this by A Challenge exchange game of Build this puzzles, using a League chart to facilitate the game.

The context

mathematical content	content domain: any; target audience: 11-14 (could fit younger & older children with careful planning); Skill Domain: Reasoning, Argumentation, Problem solving, Computational;
Learning and Instruction	Mathematical content: explicit, embedded in rules / embedded in toy; Didactical functionalities: Characteristics of rules - Defined / Ambigue; Metaphor: participatory; Rationale: Games as meaningful contexts for pupils to develop mathematical contents, Games as attractors to motivate and involve pupils; Grand theory: Socio-Constructivism;
Educational Context	role of educator: facilitator, subject teacher; ratio of learning to playing: most of the learning occurs directly through playing; collaborative learning; orchestration: gaming as competition;
Games	players: single or teams; game facilitator: human facilitated, although theoretically could be assisted by artificial agent; game time: turn-based, a-synchronous, ideally spread over several weeks; Intended use: pure educational; games as genres: Puzzle, programming games; game as media: Computer game or Paper based game; games as social activities: both face-2-face and computer-mediated, multi-player;
Interaction Design	
Software Design	

The pattern

Guess my X is a pattern of game structure, which can be adapted to a wide range of mathematical topics. It extends the Challenge exchange pattern to encourage discussion and collaborative learning, and to break down classroom hierarchies. It uses the Build this pattern to engender reflection and discussion about the relationships between mathematical objects and the processes that produce them.

Game instance

The game is facilitated by the teacher. Students play two different roles: *proposers* and *responders*. A *proposer* sets a challenge, in the form of a mathematical object which she constructed. The explicit rules of the game define the nature of the process by which this object can be created, but not its details. The aim of a *responders* is to reproduce this object, by uncovering the hidden process that generates it. If successful, the *responder* publishes the details of this process, typically in the form of a computer program which implements it. For example, the objects can be graphs, and the processes the functions that generate them.

The rules of the game are intentionally left vague, in the sense that the evaluation function used to determine the responders' success is not fully specified. This requires students to negotiate what constitutes a correct answer, and in doing so collaboratively refine the underlying mathematical concepts.

It is important to keep the Mathematical content explicit from the start. The game is not a sugar-coating to disguise the mathematics: it is a game with mathematical game-pieces.

set-up phase

- Before the game begins, the teacher needs to verify that the players have a minimal competence in analysing and constructing the mathematical objects to be used.
- Teacher introduces the rules of the game and the game environment.
- Teacher simulates one or two game rounds as a whole class discussion.
- Students may need to initialize their game space on the chosen collaborative medium.

set-down phase

- At the end of the game, students should write a reflective report on their experience. This report should review both their gaming experience and their learning experience.
- The teacher leads a whole class reflective discussion, based on these reports. This discussion starts off from the game elements, but eventually shifts focus to the mathematical issues. Specifically, the discussion should highlight the issue of the evaluation function and its resolution
- If the collaborative medium supports this, the discussion leads to the writing of a group report.
- The discussion concludes with questions for further investigation.

Game session

The game sessions for the *proposer* and the *responder* are different, although the same player can play both parts in parallel.

- *Proposer* initiates the game, by constructing and object according the game rules and publishing it. She then waits for responses.

- *Responder* chooses an attractive challenge, and attempts to resolve it. If she believed she has succeeded, she responds to the challenge by posting the object she constructed and the method she used.
- *Proposer* reviews the response, and confirms or rejects it. If the response is rejected, an argument needs to be provided.

Play session

Each play session involves one iteration of the game. Students tend to prolong their interaction in the game, by providing secondary challenges, etc. Since the iterations are a-synchronous, there may be a time gap of several days between turns.

Related patterns

Leads to: Implementing a Behavior

Elaborates: Challenge Exchange; Build this

Notes

Both *proposers* and *responders* tend to converge to challenges which are hard but not too hard. When the environment encourages social cohesion, players seem to respond to reciprocate 'good' challenges. This trend has several advantages:

- It ensures that the difficulty level students encounter is optimal for learning.
- It encourages gradual escalation of mathematical difficulty.
- It provides the teacher with a non-invasive monitoring mechanism to assed students' performance.

Examples

- Guess my Robot
- Guess my garden

4.3.2 An illustrative alpha pattern

The pattern below exemplifies an alpha pattern. Although each field in the metadata table has a datum within it drawn from the typologies, these are not unique and other data could be added. The description of the content and the initial problem to be addressed requires more detail. More work is required in ensuring that the hyperlinks to other patterns are set up correctly and consistently. Some minor formatting inconsistencies need to be corrected. See also:

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

Incidental learning		Category:	Deployment	[Edit Info]
Created	Michele Cerulli, 21 August, 2006	Modified:	15 November, 2006	[Edit Page]
Status	seed	Rank	Top of Form 5 Bottom of Form	[Publish]
Summary	Addressing antipathy towards learning mathematics for many students, avoiding making the learning explicit by setting a game.			

The problem / intent

Addressing antipathy towards learning mathematics for many students, avoiding making the learning explicit by setting a game.

The context

11-14, mixed ability, classroom-based, open-ended task, low degree of teacher intervention

Mathematical content	Basic shape and number exercises.
Learning and instruction	Drill and practice.
Educational context	11-14, mixed ability, classroom-based, open-ended task, low degree of teacher intervention
Games	Games involving counting, matching shapes, e.g. Dominos, MyMaths
Interface + interaction	N/A
Software design	N/A

The pattern

This deployment uses no introduction to the game and is simply open-ended play activity within time set aside within the classroom. There is no debrief and no separate assessment.

Related patterns

Leads to:
Follows: Bootstrap
Elaborates: Deployment
Elaborated by: informal learning in parallel

Notes

Rationale: "Kids love it, and often don't realise they are learning. Especially the lads."

Comments: Value of hiding learning. I.e. of setting tasks that do not explicitly appear to be learning-based. This raises questions of gender predisposition to games and/or antipathy towards learning.

The point here is that many standard games require skills such as counting, matching shapes etc. The teacher just lets the children play the games in order to incidentally practice those skills.

Examples

Link to case studies.

4.3.3 Levels of abstraction

The full list of patterns can be found here:

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

However, as can been seen from the site, the patterns do not exist in independent isolation. It is essential to view them as structurally dependent on 'parent' patterns. In this sense, any pattern is seen as elaborating its parent/s. At the same time, patterns lower in the hierarchy are structurally dependent upon it and can be thought of as children. In this sense the pattern is elaborated by its own children and grandchildren. The diagram below shows a small piece of the complex structural relationship that connects the patterns so far identified.

Table 4: A section of the hierarchical map of patterns

-- Deployment

-- Wrapper

Problem of the month

-- Scenario

Drill & Argue

Crescendo

+ The unpredictable

Missing the math

+ Starter

+ Plenary Starter

disrupted plenary starter

Parallel starter

-- Student learning

+ Informal learning

informal learning in parallel

Games in parallel to non-game activities

+ Independent study

Independent study - reinforcement

Independent study - challenge

Own productions

Performance

The inter-relationship of the deployment patterns identified is shown in the above hierarchical structure. The highest level of abstraction is that of *deployment*. When describing this pattern it was felt that this was such a high degree of abstraction that a pattern language was inappropriate for its description. Instead the general principles of deployment are discussed within this part of the tree.

The next level of concretisation of the concept of deployment is the larger generic concepts of types of deployment. Those so far identified are those of *wrapper, scenario* and *student learning.* These are also at too high a level of abstraction for specifics of context and content to be delineated, and these also describe the generic principles of the group of patterns. For example, the group of patterns under *scenario* describe the situations in which teachers may employ games within the classroom that are not orientated towards conveying particular curriculum content, but may instead be to promote group identity, student engagement with classroom activity, etc. Within the pattern, therefore, specifics of mathematical content, interface design, etc are not appropriate. One of the discoveries of the project with respect to deployment was the high proportion of teachers that use games within the classroom, not specifically for conveying curriculum content, but rather for these ancillary motivational and engagement activities.

The next level of concretisation is that of the set of activities that fall within these broad conceptual principles. For example, the *scenario* principle is subdivided into *Drill & Argue*, *Crescendo*, *The unpredictable* and

Starter. The patterns at this level describe the intent, goals and contexts for the deployment, but not the specific individual deployment activities that achieve these goals. *Crescendo* is the name for a set of activities that aim to encourage students in debate, The set of *The unpredictable* patterns are a collection of those patterns that aim to disrupt the students' preconceptions and *starter* patterns aim foster a collaborative atmosphere within the classroom.

Within each set of deployment patterns are the individual patterns that describe an actual sequence of learning events. Within the original collection of patterns these were described within discussions as "level one" patterns, although the term became obsolete as we developed further levels of abstraction. However, this was the level of concretisation at which practitioners generally reported their activities. An example of this level is *Plenary Starter.* Plenary starter describes the activity of beginning a classroom session with a group of students and playing a game with them in order to promote engagement and motivation. At this level the specifics of student type, specific games played, and educational context are significant.

A further level of elaboration is also possible, that of potential complication of learning events. This is always possible within deployment, in that the predetermined structure planned by teachers does not always match the actual activity, due to additional factors, either environmental (for example, technology not working) or due to the responses of the students. One of the goals of those looking at deployment was to apply the pattern language process to describing these chaotic elements, and to use the pattern structure as a means to suggest mechanisms for dealing with these elements. This degree of concretisation then forms the basis for our guidance to teachers, in that a comprehensive set of patterns will be constructed that describe the possibilities for deployment, the range of contexts in which these are appropriate and ways to respond to the potential complications to these activities. A practitioner can then use the set of deployment patterns and apply them to their own practice

Within the path deployment/scenario/starter/plenary starter/ described above we have the example of *disrupted plenary starter*. This describes the experience of some teachers that when deploying games in the *plenary starter* pattern, students can exhibit disruptive behaviour. This is expressed within the pattern language as an elaboration of *plenary starter.* The teacher's experiences of effectively dealing with this disruption are included as part of the pattern of *disrupted plenary starter*.

The production of so many patterns in such a short period of time is testimony to the efficiency with which all partners have worked together. Nevertheless, there is no doubt in our mind that the real project has only just begun. The discovery of further connections that in a sense simplify the map at its highest level will be a crucial indicator in the future as to whether this approach can do real work in helping developers and practitioners to adopt a pattern approach.

4.4 D40.4.1 International Workshop

This deliverable was completed in month 36. All partners contributed towards the final draft.

In fact, the project has generated much interest amongst a variety of communities, including developers and educationalists, including mathematics educationalists. Five workshops have been held:

– A workshop at the *1st World Conference for Fun 'n Games*, on June 26, 2006 at Preston, England.
– A seminar at the *Education Industry workshop: Designing mobile Game-Based Learning Models*, on Sept 14th at the Anglia Ruskin University, Chelmsford, England.
– A seminar at the *London Knowledge Lab* on October 19, 2006, UK.
– A workshop at *On-Line Educa Berlin*, on November 29th at the 12th International Conference on Technology Enhanced Supported Learning and Training, Berlin, Germany.
– A workshop at the *17th ICMI Study Digital Technologies and Mathematics Teaching and Learning* on December 3rd-8th at Hanoi, Vietnam.

The early workshops were used to generate interest amongst the different communities and to help us generate patterns and a working methodology. The later workshops focussed more on disseminating findings to the different communities.

4.5 D40.4.2 Streaming Video

Each of the workshops has been used also to generate video which is streamed to visitors to the project website. We have found that the most effective means of doing this has been to edit the video into smaller elements, each element telling part of the story of the project.

A pattern language for mathematical games

was presented by Niall Winters & Yishay Mor at the London Knowledge Lab on Thursday 19 October 2006.

This presentation outlines the theoretical background of the project and walks through two trails (case studies): *the 3 C's* and *beginning the design process*.

(90 min)

The mGBL Workshop: Dave Pratt, a Design Research Primer

Dave Pratt presents the paradigm of *Design Research*, as a theoretical foundation for our work.
(13 min)

mGBL: Niall Winters, Introduction to Learning Patterns

Niall Winters presents the Learning Patterns approach.
(11 Min)

mGBL: The GmX trail

Trails are designed to provide accessible paths into our resources, tools and methodology. A trail is not a structured resource in itself, but rather a guided tour through a partial set of our outcomes, from which the newcomer can get a sense of our approach, and venture on to independent explorations.

This trail passes through several case studies and their derived patterns.

(1 hour)

5. Added Value to the KJA

The Learning Patterns project added value to the KJA, through a number of activities that were on-going throughout the project.

5.1 Virtual Doctoral School

For the *Virtual Doctoral School (VDS)*, we provided a large literature review (http://lp.noe-kaleidoscope.org/outcomes/litrev/) and six typologies, which provide a structured lexicon for the community to use when developing learning patterns. They were derived from six core knowledge domains we feel are important when in the design and deployment mathematical games: mathematical context, learning and instruction, educational context, games, interface and interaction and software design. Significantly, it was important that the typologies were easily accessible and useable by VDS users and so we designed and built an *online collaboration tool* (http://lp.noe-kaleidoscope.org/outcomes/typologies) that allows for typology development, communication and discussion between users. Furthermore, these typologies were developed in tandem with several supporting case studies, which are also available for VDS users (http://lp.noe-kaleidoscope.org/workspace/cases) to get a sense of the experiences that informed typology development.

5.2 Shared Virtual Lab (SVL)

Our primary contribution to the KJA, through the SVL, has been the provision of over 100 learning patterns. The patterns are a repository of objects aimed at supporting the emergence of community of practice. The ones in *beta* are available from here: http://lp.noe-kaleidoscope.org/outcomes/patterns/.

As a result of our *five workshops*, we have made available to the SVL and *online video course* (see Section 4.5), which consists of the following for every workshop:

- A streaming video of the talk
- The talk slides

- Links to the associated patterns discussed
- Further reading in the form of papers
- An online discussion area to post and reply to questions

An example talk can be found here: http://lp.noe-kaleidoscope.org/outcomes/lkl. This talk was also the first Kaleidoscope Webinar – when researchers around Europe watched the video live, a chat facility was also made available. This resulted in this talk being the most popular run online at the LKL.

5.3 The Kaleidoscope Integration Process

Our main link within Kaleidoscope has been with TELMA, investigating the relationships between learning patterns and didactical functionalities. The construct of Didactical Functionalities (Cerulli et al 2004) individuates three key concerns for describing how a technological tool can be employed to achieve a given educational goal. For the case of games (see Literature Review Deliverable), such concerns are: the game, the educational goal, and the modalities of employing the game for achieving the educational goal. Such concerns have been included in the Learning and Instruction Typology where for each of them a set of possible descriptive values have been defined. Such a set is certainly not exhaustive, but enables us to classify and/or label games and their usages. For what concerns design patterns, it is possible to describe some aspects the context of the pattern precisely in terms of Didactical Functionalities, and this is what we did when this was relevant to the described pattern. In particular it may be the case that the context of a given design pattern needs a specification of (referring to the Learning and Instruction Typology) the educational goal, and some aspects of the modalities of employing the game to be designed. This is the case of those patterns that aim at the design of games, and which also contain, in the description of the pattern itself, information on the game to be designed.

Conversely, it may be the case that a game is given with a set of educational goals, but the modalities of employing the game are to be defined. This is the case for instance of the patterns that we classified as Deployment patterns. Such patterns describe the use made of games by students and teachers. In this case, the context of the pattern is described also in terms of the educational goal to be addressed, and of the characteristics of the considered game (using the didactical functionalities branch of the Learning and Instruction Typology). In this case the pattern's description contains the modalities of employment of the game. Concerning the deployment patterns, three specific ways for designing the modalities of employing a game have been taken into account, and have been instantiated in three collections of patterns: Wrapper, Scenario and Student Learning. Finally, we would like to observe that all the patterns we designed provide (combining context and pattern) a didactical functionality to be associated to a game. As we described, we did this either attaching a modality of employment to a game (addressing a goal), or designing a game to address a goal and to be employed in a specific way. This provided us with basis to communicate with the TELMA ERT of Kaleidoscope enriching the development of our design patterns and helping them with the refinement of the concept of Didactical Functionalities.

Furthermore, as part of the EDEN conference in June 2006, we worked with the *Design patterns from recording and analysing usage of learning systems* JEIRP to present our joint work on Design Patterns. This was organised as follows:

- Niall Winters - London Knowledge Lab, Institute of Education, University of London, UK described research into Learning Patterns for the Design and Deployment of Mathematical Games
- Donatella Persico - Institute of Educational Technology, National Research Council, Italy described research into Design Patterns for Recording and Analysing Usage of Learning Systems

These acted as the basis for a wider discussion on the difference in needs that exists between *researchers and policy makers* and how this is also somewhat to blame for the current, somewhat disjointed relationship between both parties. From a researcher perspective, the workshop proved to be an ideal forum in which to discuss the need for effective channels of communication with policy makers and to delineate the nature of research outcomes in relation to policy formation.

We also built links with the Mobile Learning Initiative, with a view to using our patterns in mobile learning contexts. As part of the project, patterns from the use of mobile learning technologies were developed. In addition, Niall Winters ran a 1-day session on 'What is mobile learning?' as part of the Kaleidoscope Mobile Learning Initiative Workshop on "Big Issues in Mobile Learning" in June 2006, with a resulting book chapter publication.

Finally, we contributed to the integration process by hosting five workshops, as outlined in Section 4.4, linking with researchers and industry across Europe, in particular with the building connections with the mGBL (Mobile Game-Based Learning, http://www.mg-bl.com) European project, supported under the 6th Framework.

5.3.1 Other contributions

Mor, Y. and Winters, N. (in press). Design Approaches in Technology Enhanced Learning, Journal of Interactive Learning Environments

Winters, N. (2006) What is mobile learning? In Sharples, M. (Ed.) Big Issues in Mobile Learning: Report of a workshop by the Kaleidoscope Network of Excellence Mobile Learning Initiative. Nottingham: University of Nottingham.

To the Kaleidoscope *Open Archive*: Contributed papers and videos to http://telearn.noe-kaleidoscope.org/

We've made a notable contribution to the development of the KAL intranet, by working closely with the AC CMS team to define and test functionalities. Some of our innovations were used by the Vision Task force.

Demo of our work given at EDUCA in Berlin, November 2006.

6. Conclusion

In this concluding section, we first summarise the main findings from the project and secondly list recommendations for the relevant communities.

6.1 Findings

This project has shown the feasibility of mapping the relationship between the design and deployment of mathematical games and has done so by putting in place a methodology which could be use to continue the design research that this project has begun. This project has shown that:

1. The relationship between the design of mathematical games and their deployment is deep, complex and structured.
2. This relationship can be successfully explored, mapped and understood through the collaborative efforts of the design and practitioner communities, each community informing the activities of the other.
3. The nature of mathematical game design, deployment and the relationship between the two can be captured in an extensive set of learning patterns with a highly defined and common structure.
4. The ontology of those learning patterns can be structured as a hierarchical map, in which the relationship between patterns is such that each can be seen of a special case of a higher level pattern, and each is elaborated by lower level patterns.
5. The existence of such a hierarchical map potentially enables access by interested communities to a relatively small set of high level patterns after which the user may wish to drill down to more specialised cases.
6. Creating and understand the nature of such a map demands an appreciation by practitioners of its structural nature and by designers of its connection to learning activity and that the construction of such appreciations is non-trivial.
7. The methodology developed within this project, incorporating the use of trails and case studies to complement and illustrate the hierarchical map, may provide the key towards the successful construction of a more exhaustive map, which better represents the relationship between design and deployment.

6.2 Recommendations

We present some initial recommendations on learning patterns development, based on our own practice during this project.

6.2.1 The workshop approach

The delegates at our workshops were primarily from the design community. The process below is intended to help participants develop specific design patterns that might support their development of mathematical games.

1. Use the online tools at http://lp.noe-kaleidoscope.org/ as your resource.
2. Make sure all participants understand the patterns approach. This is outlined in the design patterns deliverable and in the video course. Refer to the literature if further details are required.
3. Introduce the idea of patterns using the trails we developed. This will allow you to contextualise the patterns approach within a *specific* problem to be addressed, for example: Beginning the design process.
4. Get participants to reflect on their own practice by developing short cases of their own practice. Leave ample time for discussing these.
5. Next, working in small groups, have participants begin to turn each case into a set of patterns. Use the typologies as appropriate and our patterns as a guide.

6. Have participants link their own patterns with those developed by the project, to aid understanding of how the pattern approach can support development of mathematical games.

6.2.2 The practitioner approach

The teachers and practitioners involved in our project were primarily close contacts of the project team (see the final comment in Section 6.3). The process below is intended to help such practitioners develop specific design patterns that might support their deployment of mathematical games.

1. Discuss the practitioner's use of mathematical games.
2. Compare that deployment activity with others such as by fellow teachers, departments within the school, or using the case studies at tools at http://lp.noe-kaleidoscope.org/
3. Introduce the idea of patterns using the trails we developed. This will allow you to contextualise the patterns approach within a *specific* problem to be addressed, for example: Beginning the design process.
4. Reinforce the patterns approach. This is outlined in the design patterns deliverable and in the video course. Refer to the literature if further details are required.
5. Working as a department, or supporting the individual through face-to-face contact, have participants begin to turn each case into a set of patterns. Use the typologies as appropriate and our patterns as a guide.
6. Have participants link their own patterns with those developed by the project, to aid understanding of how the pattern approach can support development of mathematical games.

6.3 Final Comment

Perhaps the most significant development has been our own awareness of the extent of the problem of mapping the relationship between the design and deployment of mathematical games. The findings above (Section 6.1) indicate that this project has demonstrated the feasibility of this task. What has also become apparent is the size of the task. We have engaged in the identification of learning patterns by bringing together contacts of the project team from within the design and practitioner communities. This was, in our view, an appropriate means of bootstrapping the design research project. Within this design research framework, we can have begun to construct the hierarchical map of earning patterns.

In fact, we have mentioned earlier (Table 1) that, during this project, we have only engaged in the first two of what we see as a four phase enterprise. We now need to extend the activity to other members of the design and practitioner communities in order to test out the current version of the map of learning patterns. We have set out our methodology for doing this throughout this final report.

As future work passed through the third and fourth phases, we would expect to re-build in an iterative fashion the map of learning patterns. We would expect that the process of working with the extended communities would throw up many difficulties in using this methodology that we do not at the moment foresee. We would also expect that the richness of the patterns would be greatly enhanced. Although the number of patterns would inevitably rise, we do not see this as a problem as long as we continue to strive to identify the relationship through which patterns elaborate and are elaborated by each other. In this way, we would expect the emergent map to remain accessible and manageable.

7. Bibliography

Alexander, C., with Davis, H., Martinez, J., and Corner, D., 1985. *The Production of Houses*. New York, NY: Oxford University Press.

Béguin, P. (2003) Design as a mutual learning process between users and designers. *Interacting with Computers,* 15 (5), 709—730.

Dearden, A., Mcmanus, B., Allgar, E. and Finlay, J. (2002a) *Using Pattern Languages in Participatory Design*. Paper presented at the Computer Human Interaction Conference: Workshop on Patterns in Practice, Minneapolis, USA.

Dearden, A., Mcmanus, B., Allgar, E. and Finlay, J. (2002b) Patterns Languages in Participatory Design. In F. Detienne, J. Finlay, and X. Faulkner, X (Eds.) *People and Computers XVII: Memorable yet Invisible, Proceedings of Human Computer Interaction* (pp. 159–174). Berlin, Germany: Springer Verlag.

Druin, A. The role of children in the design of new technology. *Behaviour and Information Technology*, 21 (1) 1– 25.

Muller, M. and Kuhn, S. (1993). Participatory design. *Communications of the ACM*, 36 (6), 24—28.

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