Participatory Learner Modelling Design: a Methodology for Iterative Learner Models Development

Mihaela Cocea[a,*], George D. Magoulas[b]

[a]*School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, United Kingdom*
[b]*London Knowledge Lab, Birkbeck College, University of London, Malet Street, London WC1E 7HX, United Kingdom*

Abstract

Learner models are built to offer personalised solutions related to learning. They are often developed in parallel to the development of adaptive learning systems and thus, linked to the system's development. The adaptive learning systems literature reports numerous accounts of learner model development, but there are no reports on the methodological aspects of developing learner models and the relation between the development of the learner model component and the rest of the system. This paper presents the Participatory Learner Modelling Design methodology, which outlines the steps for learner model development and their relation to the development of the system. The methodology is illustrated with a case study of an adaptive educational system.

Keywords: Learner/User Models, Adaptive Systems, Participatory Design, Methodology, Iterative Development

1. Introduction

The ability to personalise in order to adapt to the needs of a variety of students and accommodate their different background, skills and abilities is becoming an important feature of e-learning systems. To this end, a lot of research effort has been spent in the last 10 years in the area of adaptive learning systems and a variety of methods have been proposed to build learner models, which allow a system to personalise its

*Corresponding author
Email addresses: mihaela.cocea@port.ac.uk (Mihaela Cocea), gmagoulas@dcs.bbk.ac.uk (George D. Magoulas)
URL: http://coceam.myweb.port.ac.uk (Mihaela Cocea)

interaction to individual learners. A recent review paper on the subject of learner modelling [19] outlines the different approaches for learner modelling used in the last decade.

Learner models are a type of user model, where the user is a learner. User models typically store information about a user (e.g. individual traits, goals, plans, preferences) and enable a system to adapt its behaviour to the individual user. User models are used in a variety of systems, such as Adaptive Information Systems and Recommenders, Mobile/Ubiquitous Systems, Adaptive Hypermedia Systems and Adaptive Educational Systems [15].

Several terms are used to indicate a systems' capacity to adapt to users. The most frequently used terms are personalisation and adaptation. The first refers to the effect the system has on the users, while the latter refers to the changes the system produces for different users based on their user models. In other words, from system design point of view, we think it is important to separate the purpose, i.e. personalisation, from the mechanism that achieves that purpose, i.e. adaptation. The adaptation is typically achieved through the utilisation of user models, which are the focus of this paper; consequently, throughout this paper we use mostly the term adaptation. The two terms, however, are deeply interlinked as the purpose of building user models in to provide personalised interaction.

The process of creating a user model, and consequently a learner model, and keeping it up-to-date includes three stages [80]:

1. what is being modelled? (nature)
2. how is this information represented? (structure)
3. how is the model maintained? (user modelling approaches)

User models can be built for individuals or groups of users. Early user modelling research focused on groups and used stereotypes available *a priory* for this purpose; later on, most research focused on individual user models; however, research on group models continued (e.g. [72, 10]) and grew over the last decade (e.g. [91, 52, 8, 62, 86, 42]).

In the last decade, there have also been growing developments in the direction of user models interoperability [17], including: a general user model ontology for uniform interpretations of distributed user models [45], generic user models that can be used to define user models for a variety of applications [56], cross-system user modelling where a user model from one system is re-used in another [1]. These developments are possible when the user modelling process is independent from the domain [56]; cross-system user modelling allows re-use of user-models in applications that deploy similar user information, such as web-based recommender systems.

For many adaptive systems, however, the user models are still tightly linked to the system that uses them in general, and the user interface and adaptation modules in particular. This is notably the case for educational systems where a participatory design [35, 66] is used.

At the same time, there has been a shift from building systems as a whole to separating the different components of the system. The movement towards service-oriented architectures [49] and component-based development [30] emerged from the need to separate the development of the various components of a system from the development of the system as a whole, and led to challenges in assembling different components and services.

Particularly within the educational technology research, there is a move towards grid technologies, which enable sharing of learning resources in heterogeneous and geographically distributed environments [76]. This paradigm promotes the focus on stateful services and on flexibility in the way they are combined [3]. Unlike stateless services, stateful ones keep a record of the previous transactions; the interested reader can find more details on stateful services in [37].

Moreover, there is increasing focus on user involvement in product or service design, not only in the initial development phases, but throughout the development process [71], as well as involving the user in the design of particular components of the system [6].

The separation between the development of the learner model component and the development of the system is known to be a difficult issue [55] because the development of the learner model component needs to be coordinated with the development of the system. Despite the advances in the learner modelling research area, the literature is lacking in methodological frameworks for the development of learner models and the interplay between the learner model and the system development.

In this paper we focus on learning applications with a strong link between system and user model development. Moreover, we are particularly focusing on user-centred participatory design [66], where the users are involved in the development of the system and of the user model component. Consequently, this methodology is appropriate for adaptive learning systems which are built with the involvement of users.

A case study illustrates how the methodology works in practice. The case study refers to the development of an adaptive educational system for teaching mathematical generalisation in classroom settings to children of 11 to 14 years old. More details about the design of the entire system can be found in [70], while details about feedback elicitation are given in [63]. The case study is representative of adaptive learning systems, and in particular of exploratory learning environments, and illustrates how the methodology facilitated the integration of complex requirements in the development of the learner model component in parallel to system

development.

The rest of the paper is organised as follows. Section 2 provides an overview of the learner modelling process, previous literature on learner modelling and adaptive systems development, and on iterative and participatory design. Section 3 describes the specifics of our methodology and the interplay between system development and learner model development. Section 4 illustrates, through a case study, how our methodology provides a structural systematic approach to learner model development in the context of participatory design of a complex adaptive educational system. Section 5 discusses our methodology, including its generality and the lessons that we learned from its use, that we believe other researchers will find useful.

2. Background

This section presents an overview of the literature in relation to: (a) learner modelling, (b) adaptive educational systems development with a focus on the learner model development and relation to system development, and (c) iterative participatory design.

2.1. Learner modelling

A learner model is a representation of a learner and consists of data about the learner or about what the learner does. Typically, a learner model would store data about a learner's knowledge, preferences, goals, tasks and interests [14].

The term Learner Modelling refers to the process of generating a learner model in the context of an intelligent learning environment [13]. A learner model enables the system to adapt to the learner who uses it and ideally includes all information about the learner's behaviour and knowledge that influences their learning and performance [88]. The content of a learner model depends on the learning environment and includes inferred information about aspects such as a learner's goals, plans, knowledge, attitudes and abilities, but the most important information about a learner is his or her knowledge of the subject that is being studied [13].

Table 1 gives an overview of learner modelling approaches, looking at: what is modelled; when adaptation occurs; the form of adaptation, modelling technique and modelling approach. This overview focuses on capturing the variety of aspects that are modelled, as well as the diversity of approaches used. It is not meant to be an exhaustive overview of the filed.

In terms of what is being modelled, a variety of aspects are used for different purposes: knowledge, goals and tasks, used background, individual styles such as cognitive and learning styles, and contextual information such as affective states of the user, user device and user location [15].

When the adaptation occurs depends on what is being modelled and the purpose of the adaptation. For example, if knowledge is modelled, the adaptation occurs when learning can be facilitated by adapting the levels of details presented or by providing feedback when solving a problem/learning task.

The form of adaptation refers to the change that is introduced according to the information that is being modelled. For example, different learning goals may lead to different materials or different sequences of materials being presented to the learners.

The modelling technique describes the specific user modelling technique used (e.g. Bayesian Networks, Case-based reasoning), while the modelling approach describes categories of user modelling approaches (e.g. stereotype, feature-based, overlay). Two broad categories of modelling approaches are feature-based and stereotype. The feature-based approach models specific features of individual users, such as knowledge, goals and interests [14]. The stereotype models work by grouping users in several categories called stereotypes; all users belonging to the same stereotype receive the same adaptation.

Other types of modelling approaches are overlay models and uncertainty-based models. Overlay models represent the users' knowledge as a subset of a domain model; the domain model represents the expert knowledge of a subject. Uncertainty-based models refer to the uncertainty introduced in the diagnosis through inference. For example, when assessing users' knowledge of a concept by their performance to a test, the observation that they did not do well in the test leads to the conclusion that the user probably does not master that concept to a satisfactory level. On the contrary, no uncertainty is involved in establishing the platform of a user to inform adaptation to screen size, for example.

2.2. Adaptive learning systems methodologies

Several approaches have been proposed for the development of adaptive systems that employ user models for adaptation and personalisation. A framework for the development of adaptive systems taking into consideration context and user models was proposed by Zimmermann et al. [92]; they focused on the relationship between user and context modelling. Michaud and McCoy [64] proposed a methodology for acquiring stereotypes to be used in the modelling process.

Benyon and Murray [5] outlined an architecture for adaptive systems that includes a domain model, a user model and an interaction model. Methodological aspects were also pointed out such as the development of the adaptive parts of the system in parallel to the development of the application and the explicit separation of the user model from the other components of the adaptive system, which would facilitate easy modifications in the adaptive mechanism as the details of interaction are better understood. Our methodology endorses these methodological principles and outlines a systematic way of making modifications

Table 1: Learner modelling approaches

What is modelled	When adaptation occurs	Form of adaptation	Modelling technique	Modelling approach	References
Knowledge/ competencies/ skills	When selecting to read a particular section	Presentation only of text that is relevant to the user (i.e. is not yet known); level of details according to user knowledge	Scalar models (scalar value indicating where the user is on a scale for novice to expert)	Stereotype	[11]
	During problem solving	Feedback on the violated constraint	Constraint-based modelling	Feature-based, Overlay	[65]
	After solving a task	Feedback on correctness; verification and explanation	Bayesian Networks	Uncertainty-based, Overlay	[79]
	During problem solving	Feedback on progress towards solution	Case-based reasoning	Uncertainty-based, Overlay	[24]
Goals & tasks	When choosing a new task, e.g. read pages/take tests	Suggest sequence of tasks to reach learning goal, e.g. recommend pages	Goal/task catalogue and adaptation rules	Stereotype	[12]
User background	When presenting a section	Restrictive vs non-restrictive adaptation level of detail presented; annotations of recommended content	Self-assessment/ teacher input about previous knowledge before start of using the system	Stereotype	[36, 85]
Individual traits, e.g. cognitive indicators, learning styles	When presenting a new section	Presentation according to the cognitive indicators of user, e.g. text, graphics, sound or a combination of two of the three media	Self-assessment; stable indicators, so no updating needed.	Feature-based	[87]
	When navigating	Adapt recommendation of learning sequences according to learning style (and other learner characteristics)	Self-assessment (questionnaire)	Stereotype	[54]
Context, e.g. affective state, user platform	When disengagement is detected	Intervention by on-line teacher from suggested motivational strategies	Data mining / machine learning and self-assessment	Feature-based	[23, 50]
	When loading web pages	Adaptation to device characteristics for optimal display	Adaptation rules	Feature-based	[69]

to the user model as new information about details of interactions emerge/change from other components of the adaptive system.

Mangina and Kilbride [61] outlined some methodological issues when designing user models for online learning environments such as Moodle; these issues are important in term of the responsiveness of a system when there are many users: granularity, storage and retrieval.

One area of research that provides description and analysis of the processes involved in designing adaptive systems is authoring of adaptive systems. This field of research focuses on building tools that allow users without programming knowledge/experience to design adaptive solutions [29]. Several architectures have been proposed, such as the AHAM [90] and LAOS [28, 67]. Most proposed models include a domain model, a user model and an adaptation/interaction model, as suggested by Benyon and Murray [5]. However, "the few tools that have been designed for non-technical experts to author adaptive courses are not commercial but are prototypes which have only been used within third level or formal learning" ([38], p. 2781). Moreover, these approaches focus on the system as a whole, while our approach focuses on the user model component and its relation to system development.

2.3. Iterative and participatory designs

Iterative development involves building and delivering software in iterations, with each iteration being a working software system that generally has more functionality, i.e. range of operations of the system, than the version of the previous iteration. Iterative development dates back to mid-1950s [59] and it is increasingly used in research and commercial projects due to the possibility to deliver functionality in parts, which, in turn, allows effective management of risks [51]. For other models of software development, the interested reader can consult [83].

Similar to iterative development, user-centred and participatory designs are increasingly popular due to the rise of highly interactive systems, which can be defined as systems that require a significant degree of user interaction [58]. They also involve an iterative approach, but unlike the software iterative approach focused on functionality, the focus is on usability. Adaptive systems are user-centred systems in which both functionality and usability are important. In fact, these are highly interlinked, as adaptivity could be considered a functional requirement that enhances usability.

A lot of research has emerged within the last 10-15 years in the area of learner-centred design, arguing for the learners' involvement in the design of intelligent educational systems, especially when learners are children, as adults have a limited knowledge about how children make sense of software. Following is an overview of several approaches proposed in the area of iterative design with children for educational systems.

Some proposed approaches for learner-centred design focus on the design *product* (the educational system) (e.g. [81, 82]), while others focus on the design *process* (e.g. [35, 77]).

The TILT model (tasks, interfaces, learner's needs, tools) [81] was inspired from user-centred design that uses three of the aforementioned concepts, i.e. tasks, tools and interfaces, and adds a new concept that the authors argue as necessary for learner-centred design, i.e. learner's needs.

The Persistent Collaboration Methodology (PCM) [27] focuses on the process of designing intelligent educational systems. Teachers, researchers and technologists are involved in a cycle of observation, reflection, design and action. This approach is considered by Good and Robertson [40] to be more *school-centred* than *learner-centred* because learners were not part of the design team.

The term participatory design in which end users are involved and in which the users are children has been used by Druin [34, 35] who defined a methodology called cooperative inquiry. It involves a four-step process:

1. the contextual inquiry phase which involves collection of data in users' own environment;
2. the 'sticky note critiquing' phase during which children and adults critique an exiting piece of technology and, using sticky note pads record their likes, dislikes and a third category, e.g. surprises;
3. the participatory design phase in which the design team, including children, takes part in low-tech prototyping sessions;
4. the technology immersion phase which involves creating a space where children are able to access and use the existing technologies over a sustained period of time with researchers observing children's activity patterns in an unconstrained setting.

Several forms of involvement [35] are proposed to include children in the design of learning environments, which are given on a gradual scale. At the bottom of the scale the children's involvement is small as they act as *users* of technology. On the next steps, the children are more involved, acting as *testers* of prototype software and as *informants*, i.e. giving input in the design process. At the top of the scale, the children have the status of design *partners* acting as equal stakeholders throughout the design process.

Good and Robertson [40] pointed out that the focus of cooperative inquiry is on children as *technology users*, while learner-centred design has a more constrained focus on children as *technology learners*, i.e. children who use the technology as a vehicle for learning.

The Informant Design Framework [77] considers several stakeholders including children, teachers, software designers and psychologists to contribute to the design of the interactive learning environment. It

starts with specifying the learning goals and teaching practices for the domain and translate the specification initially into low-tech and later into high-tech designs. The expertise of the different stakeholders is used on specific aspects of the learning environment throughout the design process rather than having all the stakeholders working as an integrated team at all stages of the project.

The CARSS framework (Context, Activities, Roles, Stakeholders, Skills) [40] was specifically developed for participatory, learner-centred design with children. The context refers to the awareness of the broader context in which the design activity takes place. The activities describe the sequence of events that occur in the typical educational software design cycle. The roles describe the various functions that a member of the design team can fulfil, with each member possibly fulfilling more than one role. The stakeholders cover all the individuals who have a vested interest in the design process, and the skills refer to personal attributes and dispositions necessary to conduct successful design sessions. This framework can be applied to both intelligent and non-intelligent learning environments. It attempts to be fully inclusive and to be used for the design of interactive learning environments for children.

Another methodology entitled Identification-Development-Refinement (IDR) methodology [89] was proposed to address the issues related to interdisciplinary design. It aims to look at the full cycle of the design process and not just the software output and thus to include other outputs such as design patterns and pedagogical plans. Also, it focuses on engaging participants to reflect on their previous successful practices and to scaffold this reflection to generalizable solutions useful to the wider community. This methodology includes three stages: (a) the aim of the first stage is to identify potential patterns through the use of typologies and case studies; (b) the second stage looks at developing a set of patterns based on design evidence from the case studies; (c) the third stage aims to improve the patterns through collaborative discussion and reworking. The patterns are meant to *mediate* the interdisciplinary design process through their identification, development and refinement by the project participants.

Our methodology, described in the next section, complements these methodologies by outlining how the information from participants is integrated in the development of the learner model component.

3. Participatory learner modelling design

Our proposed methodology is for adaptive systems that use an iterative design in which users participate by at least providing interaction data. The next subsection, i.e. Section 3.1, presents details about the development of user models and describes our proposed methodology in terms of iterative processes. The following subsection, i.e. Section 3.2, outlines how these processes relate to the parallel development of the

system.

3.1. Learner model development

In the previous section, three stages were mentioned in relation to user models; in the following we outline the methodological aspects involved in these stages.

What is being modelled? The first stage relates to the nature of information in the user model, which depends on the adaptive system of which the user model is part. As mentioned in Section 2.1, different types of adaptive systems store different user information depending on the aim of the adaptation. For example, if the aim of the adaptive system is to deliver information through a variety of devices, the user model needs information about the user's device. Consequently, the answer to the question "what is being modelled?" is partly included in the system's requirements. This, however, needs further elaboration to address the following stages. For example, in an adaptive educational system that aims to recommend study materials based on the user's knowledge, further details are needed such as the domain of study, the concepts of this domain and the relations between these concepts (for example, concept A is a prerequisite for concept B). Designing a conceptual model to include these details is our proposed way of formalising the answer to the question "what to model" in a way that would facilitate finding answers to the questions corresponding to the subsequent stages.

A linked question to "what to model?" is "when should adaptation be provided?". This is typically a requirement for the adaptation module that should trigger adaptation at certain times and in certain forms. However, this cannot be separated from the user model module because the answer to "when to provide adaptation?" is linked in certain situations to the knowledge about the user and it is that information (from the user model) that triggers the adaptation. An increasingly popular way of addressing these complex situations which involve requirements for several components of a system is through scenarios.

Scenarios are used in the development of interactive systems to understand the situation in which the user needs to be "supported" by the system; they are now accepted in software engineering research and practice [46]. According to Nardi [68] the purpose of a scenario is to provide "an explicit concrete vision of how some human activity could be supported by technology" ([68], p.13). Scenarios are considered the basis for the overall design and for technical implementation, and facilitate cooperation between users and designers [9]. A survey of typical scenarios usage in different fields is presented in [39]. In relation to previous methodologies, one could argue that for the purpose of user modelling the patterns discussed in the IDR methodology [89] fulfil the same role as scenarios.

For the purpose of user model design and development, scenarios are used to establish what information is needed about the user and to test the modelling technique with respect to each scenario. For example, if the user model should contain information about the user's knowledge of a particular concept in order to recommend study materials at the appropriate level, the modelling mechanism should provide information about the user's knowledge of that particular concept.

How is the information represented? The second stage concerning the representation of information is informed by the first stage. Consequently, at this stage appropriate representations should be identified for the conceptual model developed previously, while also considering the scenarios. This may require only a literature search and adoption of a known representation or more innovative approaches.

How is the model maintained? The third stage involves the construction and maintenance of the user model. This is tightly linked to the previous stage, as representation of information is linked to the way it is used. In practice, the representation of information and the user modelling techniques are often decided at the same time because some representations can be employed only with some techniques and some techniques require particular ways to represent information.

Our methodology, illustrated in Figure 1, proposes three iterative processes: analysis, mapping and evaluation. The analysis process aims to answer the question "what to model?" and the result of this analysis could be formulated in a conceptual model and a list of relevant scenarios. The conceptual model should include information about the data that is needed – this is typically informed by the user interface and by knowledge of the domain. The link to the user interface is important because the needed knowledge about the user is 'extracted' either directly from the user (e.g. they declare their interests) or indirectly from their usage of the system.

The scenarios specify different situations that are relevant for the adaptation. For example, if the learner gave a wrong answer to a test question, scenarios can be defined for different levels of feedback depending on previous interaction. For example, if it is the first time the learner provided a wrong answer to that question, the feedback could inform the learner that the answer is wrong and prompt them to try again. If the learner had several failed attempts to provide the correct answer, the feedback could provide the correct answer with an explanation.

The mapping process is informed by the results of the analysis and aims to map the conceptual model

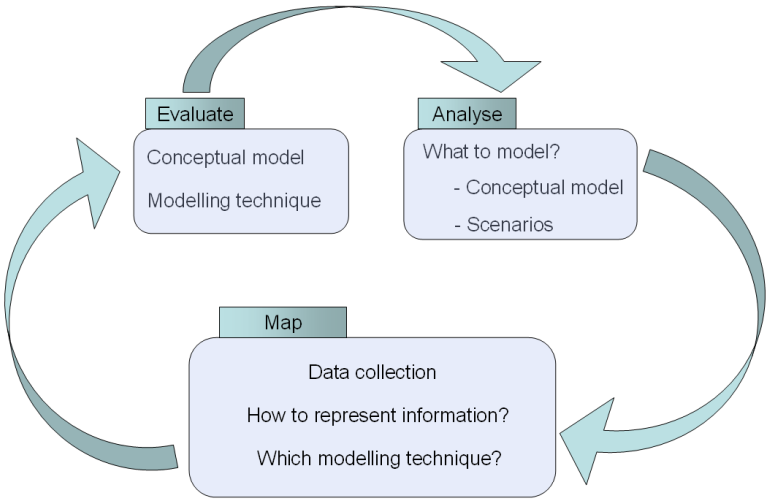


Figure 1: Processes of the proposed methodology

with user data. To enable this mapping, three aspects need to be addressed. The first one is the availability of user data, which is related to data collection. This is important especially when user characteristics are inferred, as the collected data is essential for the inference and it is informed by the current interface of the system. The other two aspects are knowledge representation and modelling technique. These are decided based on the aim of the modelling technique and they are informed by the conceptual model and user data. The design and development of the user model component takes place at this stage.

The evaluation process involves the evaluation of the practical use of the conceptual model, the scenarios and the modelling technique. The conceptual model is generally evaluated in reference to the requirements of the user model content, while the scenarios refer to adaptation requirements, i.e. in what situation is adaptation needed, and for that to happen, what is the information that the user model needs to have? In the case of educational systems, the conceptual models and scenarios are often evaluated by experts of the learning domain. The evaluation of the modelling technique involves testing its performance in terms of successful diagnosis for each of the scenarios.

The three processes are iterated, with each iteration being related to iterations of system development. This interplay is described in the following subsection.

3.2. Learner model development in relation to system development

The processes mentioned above are influenced by the development of the system in general, and two components in particular: the user interface and the adaptation module. The design of the user interface "dictates" the way users can interact with the system and what data can be collected. This has an influence

on user modelling because it is through capturing interactive behaviour that the system collects data about the user for either direct storage or for inference. Therefore, the user interface plays a key role in user model development underpinning the design of conceptual models and the choice of modelling techniques, especially when data-driven approaches are needed. In areas such as adaptive learning systems, expert knowledge may also be needed to inform the conceptual model about the domain of learning. There are of course several challenges involved relating to the differences among experts, as well as their level of expertise and perceptions of the domain [78]. For example, an academic-expert/educational researcher may be concerned to demonstrate theoretical aspects of the domain and characterise the scope and limitations of the domain theory or of the pedagogical design of the adaptive learning system. In contrast, a practitioner-expert/teacher may be driven by experiences with learning situations they are dealing with on a daily basis, and could have compiled teaching techniques, scaffoldings, or problem solving techniques that help learner's progression in the domain and the accomplishment of the teaching objectives.

The user interface is also linked to another system component, i.e. the adaptation module, which performs the adaptation based on information from the user model. As the adaptation is provided through the user interface, the design of the two is interlinked. The adaptation module also plays a role in the definition of scenarios, which in turn, inform the modelling technique development. Similar to the conceptual model, for adaptive systems where expert knowledge is needed, the decision about scenarios is informed by experts.

These interactions between the user interface, the adaptation module and the processes involved in user model design and development are displayed in Figure 2. The numbers illustrate the order involved in the development of the user model. The blue and green arrows show how the reciprocal influence between different components. The dash line block with the expert knowledge and pedagogical design indicates that this is applicable only for some adaptive systems.

Similarly to the user model, the system is developed in an iterative manner. The interplay between the iterative development of the system and of the user model module is displayed in Figure 3. The number of iterations for the user model development depends on the number of iterations for system development. In practice, there is not always a one to one correspondence between the user model and system version. After the initial development (UM v0), the next user model version may be developed after several iterations of the system development. This approach gives more stability and provides more time for user model development which cannot be as easily changed as the user interface for example. Consequently, the number of system versions will be greater than the number of user model versions. The next section present a case study for our proposed methodology.

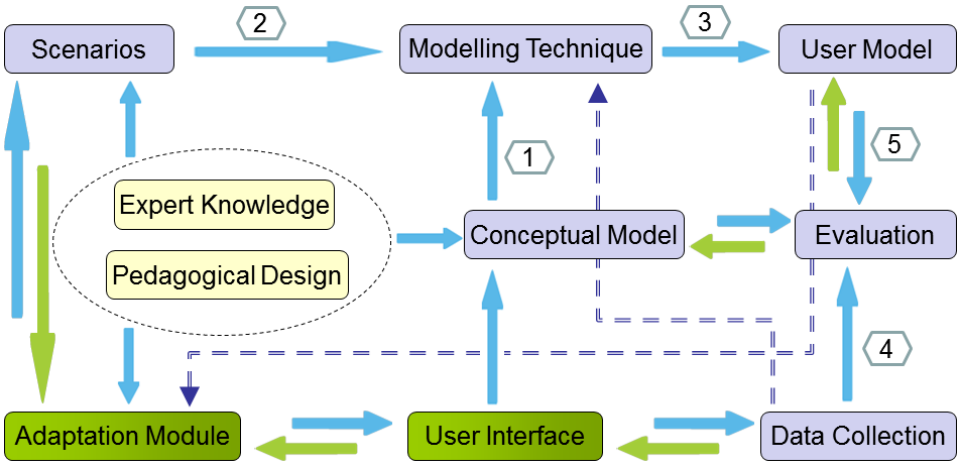


Figure 2: Interaction between system components and user model development processes

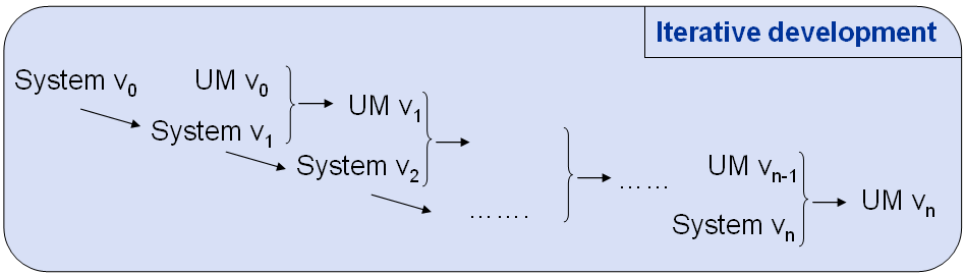


Figure 3: Iterative development of system and user model module

4. Case study

In this section we illustrate how the above methodology was applied to the user model development of an intelligent educational system for the domain of mathematical generalisation. The aim of the system is to provide tasks for 11 to 14 years old pupils in which they need to build a construction and derive an algebraic-like rule, and to provide intelligent support for the pupils while they solve the tasks. The intelligent support is provided via two components: the user modelling and the adaptation modules.

Figure 4 illustrates an example of a task that pupils are asked to solve using the system. Pond-tiling is a mathematical generalisation problem for which the students are presented with a pond typically of rectangular form of a certain width (w) and height (h) (see Figure 4) and are asked many tiles are needed to surround *any* pond.

The algebraic solution for this problem is that the number of tiles needed to surround any rectangular pond is $2 * w + 2 * h + 4$. The challenge from pedagogical point of view is to support learners in developing an understanding of algebraic rules and of how these can be used to develop general expressions. Thus

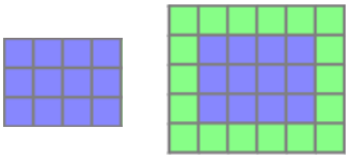


Figure 4: Pond tiling problem - pond and surrounded pond.

the activities/tasks undertaken by the student aim to facilitate their transition from building a pattern construction to appreciating the algebraic formula behind it and making it general.

Thus, the system would present such tasks to the learners and would provide affordances that allow the learners to build constructions and express algebraic rules. Consequently, a user modelling mechanism needs to enable diagnosis of both aspects.

In the following we present the development of the user model component and its relation to system development. The user model development was done in three iterations. We illustrate here the methodology for user model development for the first and the last iteration.

4.1. Initial design: learner model v0

We start by presenting the first version of the system, called *ShapeBuilder* [73, 20], based on which the learner model v0 was developed. The user interface of the system is presented in Figure 5 and includes an Expression Toolbar (b), a Shape List (c), and the Expression Palette (d).

The affordances of the system with the information available from the interaction with the interface are listed in Table 2. *ShapeBuilder* allows construction of different shapes, e.g. rectangles, L-shapes, T-shapes, and supports *numeric*, *iconic* and *symbolic representations*. *Numeric representations* include numbers (constants or variables) and expressions with numbers; *iconic representations* correspond to icon variables; *symbolic representations* are names or symbols given by users to variables or expressions. An icon variable has the value of a dimension of a shape (e.g. width, height) and can be obtained by double-clicking on the corresponding edge of the shape. It is represented as an icon of the shape with the corresponding edge highlighted - see Figure 6.

Shapes can be linked through icon-variables by defining a dimension of one shape as an expression including an icon variable of another shape. This would lead to both shapes being modified at the same time when a change occurs in the icon variable.

The Expression Toolbar allows the creation of constants, variables and composite expressions using addition, subtraction, multiplication and division. These are placed in the Expression Palette and can be used for defining an expression for the task at hand or to define the properties of the shapes in the

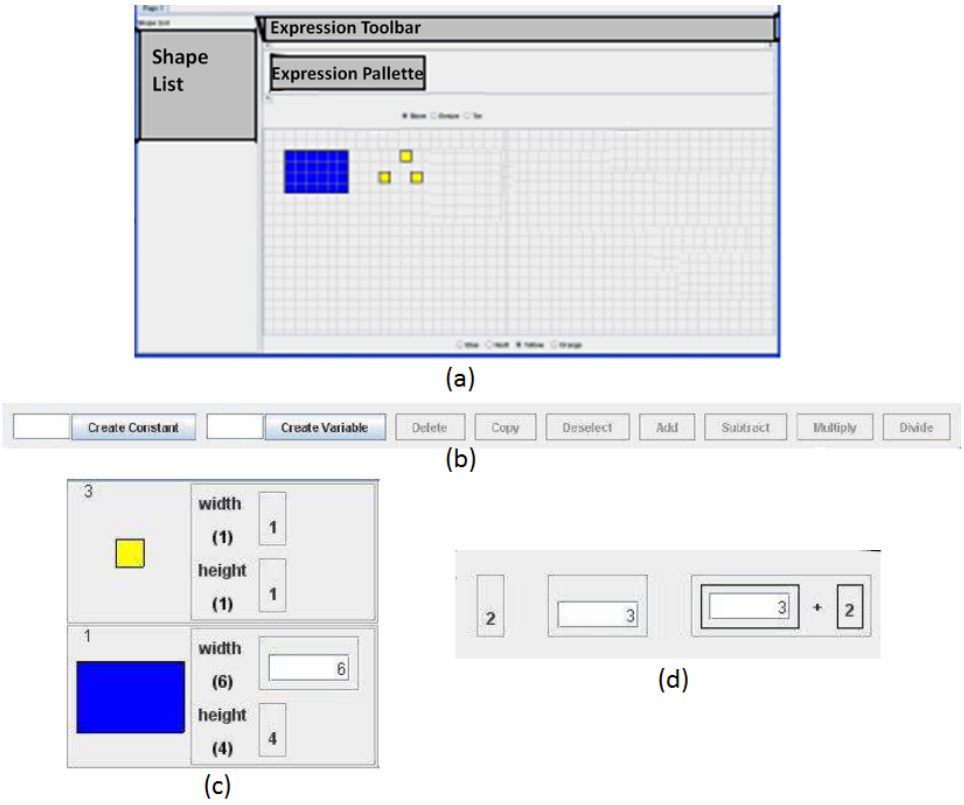


Figure 5: System v0: (a) the overall interface, with the gridded area as interaction canvas; (b) the Expression Toolbar; (c) the ShapeList; (d) the Expression Palette.

Table 2: System affordances and information available.

Affordances	Information available
Shape creation	Shape type (e.g. rectangle, L-Shape, T-Shape) Shape dimensions, e.g. for rectangle, width and height Shape colour
Shape properties modification	New value for dimension or colour
Variable representation	Each dimension of a shape can be represented in three forms: numeric, iconic or symbolic
Linking shapes	The shapes that are linked and The expression that links them
Expression creation	The created expression

Figure 6: A rectangular shape and its icon variable.

ShapeList. The ShapeList displays the shapes that currently exist on the gridded canvas and allows the creation of new shapes. Existing shapes can be manipulated on the interaction canvas - they can be moved and attached to other shapes, and can be resized by either using the mouse or changing their properties in the ShapeList. When a shape has several copies and the properties of one of them is changed, all copies are updated appropriately.

The properties of shapes in the ShapeList facilitate the derivation of the algebraic-like expression for the task at hand by providing parts of the final expression which is formed by putting together various properties of the shapes used in the construction. Constants, variables and numeric expressions lead to *specific* constructions, while icon variables and expressions with icon variables lead to *general* constructions.

In the following, the development of the learner model is presented in accordance with our methodology for each of the three processes involved.

4.1.1. Analysis

Several sources of information were used in the analysis stage: (a) task knowledge from experts and (b) the user interface affordances that allow pupils to solve the tasks in the system.

Generalisation tasks that are typically part of the UK mathematics curriculum and different solutions to these tasks were provided by the experts; these tasks are documented in [44]. To illustrate the different solutions that pupils could adopt for the same task, the solutions provided by the students (from the evaluation study detailed in section 4.1.3) for the pond-tiling task introduced earlier are displayed in Figure 7.

Consequently, the information from experts about solutions to several generalisation tasks and the affordances of the system informed the development of the conceptual model for representing solutions for tasks in *ShapeBuilder*. This is presented in Table 3 and contains properties of each part of the construction and relations between different parts.

Each solution is made of several parts and the relations between the parts are essential in defining an algebraic-like rule. Therefore, the conceptual model should include the "definitions" of parts, but also the relations between them. The "definition" of parts is given by their properties. These properties are either defined through the user interface, or can be derived from what is defined through the user interface.

There are three types of relations: (a) *value relations*, for example the width of component C_2 (top bar) of the 'I' strategy in Figure 7(h) is the width of component C_1 (pond) plus 2; (b) *dependency relations*, when a dimension type of a component is an icon variable of another component; for example, in the example above the the width of component C_2 (top bar) is dependent on the width of component C_1 (pond); (c) *order relations*, previous and next, which specify the components created before and after the current one.

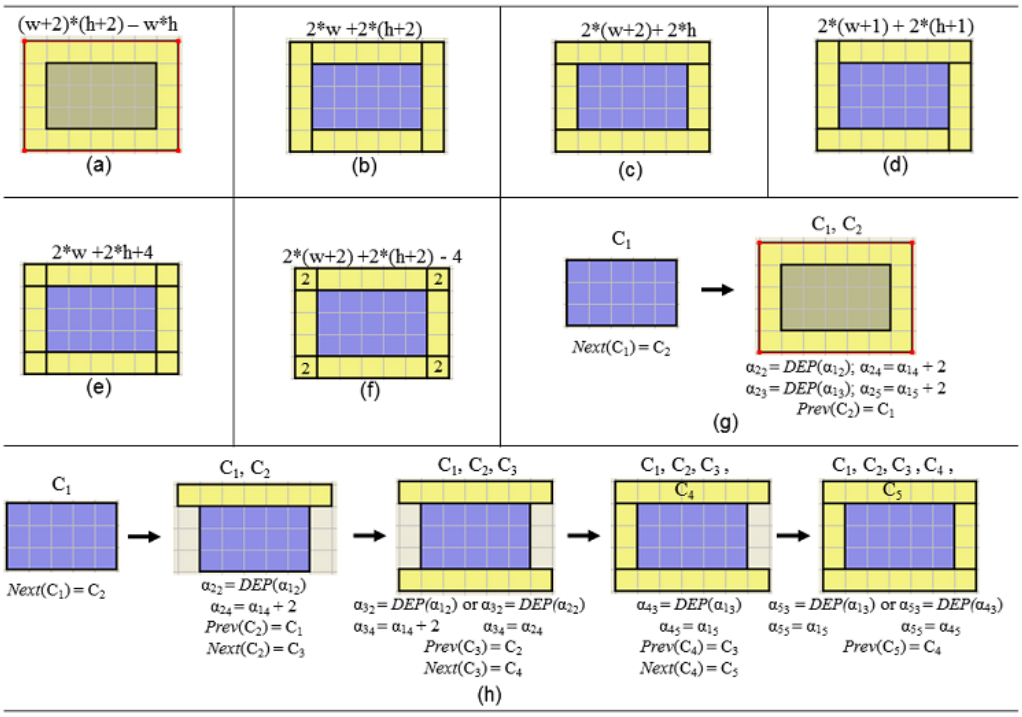


Figure 7: (a) 'Area' strategy; (b) 'I' strategy; (c) 'H' strategy; (d) 'Spiral' strategy; (e) '+4' strategy; (f) '−4' strategy; (g) Steps and relations of 'Area' strategy; (h) Steps and relations of 'I' strategy.

Table 3: Conceptual model for strategies of *ShapeBuider* tasks.

Component	Properties	Possible values	Relations
C1	Shape type	rectangle/L-Shape etc.	Value Relation (VR) 1, VR 2, etc.
	Shape colour	red/ blue/ etc.	Dependency Relation (DR) 1, DR2, etc.
	Each dimension		Order relations: Previous, Next
	- type	constant (c)/ variable (v)/ icon variable (iv) numeric expression (ne)/ expression with IV (eiv)	
	- value	numeric	
C2	Shape type	rectangle/L-Shape etc.	VR1, VR2, etc.
	Shape colour	red/ blue/ etc.	DR1, DR2, etc.
	Each dimension		Previous, Next
	- type	c/v/iv/ne/eiv	
	- value	numeric	
...	....	.....	....
Cn	Shape type	rectangle/L-Shape etc.	VR1, VR2, etc.
	Shape colour	red/ blue/ etc.	DR1, DR2, etc.
	Each dimension		Previous, Next
	- type	c/v/iv/ne/eiv	
	- value	numeric	

To illustrate the conceptual model, Table 4 presents how this translates for the 'I strategy' of the pond tiling task; see also Figure 7(h).

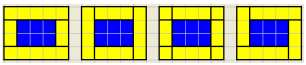
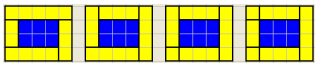
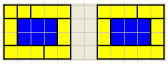
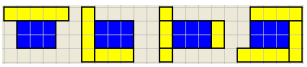
Table 4: Conceptual model for the 'I strategy' of the pond tiling task.

Component	Properties	Possible values	Relations
C1 (pond)	Shape type	rectangle	Previous: null
	Shape colour	blue	Next: C2
	Width type	iv	
	Width value	5	
	Height type	iv	
	Height value	3	
C2 (top bar)	Shape type	rectangle	VR1: C2_width=C1_width + 2
	Shape colour	yellow	DR1: C2_width=DEP(C1_width)
	Width type	eiv	Previous: C1
	Width value	7	Next: C3
	Height type	c	
	Height value	1	
C3 (bottom bar)	Shape type	rectangle	VR1: C3_width=C1_width + 2
	Shape colour	yellow	DR1: C3_width=DEP(C1_width)
	Width type	eiv	Previous: C2
	Width value	7	Next: C4
	Height type	c	
	Height value	1	
C4 (left bar)	Shape type	rectangle	VR1: C4_height=C1_height
	Shape colour	yellow	DR1: C4_heigh=DEP(P1_height)
	Width type	c	Previous: C3
	Width value	1	Next: C5
	Height type	eiv	
	Height value	3	
C5 (right bar)	Shape type	rectangle	VR1: C5_height=C1_height
	Shape colour	yellow	DR1: C5_heigh=DEP(C1_height)
	Width type	c	Previous: C4
	Width value	1	Next: null
	Height type	eiv	
	Height value	3	

From previous knowledge about the difficulties learners face with mathematical generalisation, coupled with knowledge from experts on when support is needed, several scenarios were defined. These scenarios, corresponding to categories of user strategies are given in the first column of Table 5. The second column provides the pedagogical rational for monitoring the particular strategy category, e.g. providing appropriate scaffoldings for users that demonstrate a particular behaviour. The third column displays examples of user constructions that belong to each scenario.

The first scenario, i.e. **complete strategies**, is important for detecting if the learners display behaviours that demonstrate their ability to generalise. In *ShapeBuilder*, the constructions can be built in a *specific*, *partly general* or *general* way. A *general construction* has relations between all its variable parts. For

Table 5: Scenarios from experts.

Scenarios	Pedagogical Rational	Example Constructions
Complete strategies	Identify whether the learner is working with the specific or the general	
Mixed strategies	Identify strategies of learners to guide them towards a particular one if they have difficulties to generalise	
Non-systematic strategies	Guide learners toward a systematic strategy if they have difficulties to generalise	
Partial Strategies	Guide learners by building on the strategy they started with should they be stuck or request help	

example, for the '-4' strategy in Figure 7(f), the top and bottom rows of tiles need to be linked to the width of the pond, thus indicating a *dependency relation* between the width of these rows of tiles and the width of the pond; moreover, the widths of these rows need to be the width of the pond plus 2, thus indicating a *value relation* between the width of these rows of tiles and the width of the pond. Similarly, the left and right columns of tiles need to be linked to the height of the pond and have their height equal to the height of the pond plus 2. If there are no links between the variable parts of a construction, i.e. no dependency relations, the construction is *specific*. If some links are present while others are missing, the construction is *partly general*.

Knowing if the learner is building a *specific*, *partly general* or *completely general* construction is pedagogically important and valuable for providing feedback to the learner, either to confirm that they are doing well, or to provide support if they are not sure how to proceed.

As noticed in Figure 7, all solution have an element of symmetry and minimal elegance, which facilitates the process of generalisation because the dependency and value relations are the same for several components of the construction; consequently, the definition of the algebraic-like rule becomes easier. Learners, however, use a variety of strategies when building their constructions, including combining components from elegant strategies, i.e. **mixed strategies**. Because using this approach adds more complexity to the task, detecting which combinations of strategies the learners are working with enables more personalised feedback in terms of helping the learners extract a general rule or, if that is too difficult, helping them towards using only one elegant strategy that was already partly used in their construction.

Non-systematic strategies contain "bits and pieces", even when other parts of the constructions have

been elegantly constructed. The first non-systematic figure in Table 5 shows an elegant approach for the left and right columns of tiles, which is missing for the top and bottom rows of tiles; for example, the top row is built of three bits and pieces: two single tiles and a bar of four tiles. The presence of "bits and pieces" makes it difficult to identify whether there is an elegant strategy that the learner is partly following. For example, the construction described above has elements of the 'I' and '+4' strategies. Detecting this behaviour, however, is beneficial in terms of identifying learners who try to address the task at hand by merely reproducing the form of the construction (i.e. surround the pond with tiles) without thinking about the generality of their approach. In addition, feedback can be targeted to point out to the learners that this approach is not helpful and to provide guidance towards one of the strategies that they already used.

Partial strategies refer to constructions that are not completed; in the case of the pond-tiling task, this would mean that the pond is not entirely surrounded by tiles, as illustrated in Table 5. Detecting this type of strategy is important for providing help to the learners should they need it. For example, if a learner has started to build their construction using a particular strategy, detecting that they are working with that strategy enables more targeted feedback by providing guidance on how to continue with that particular strategy. This approach is similar to the type of support that teachers would give pupils when that are partly through a task.

All scenarios aim to provide meaningful feedback to the learner during a task in relation to what they have already constructed, by identifying specific difficulties that the learners face when building a general construction and deriving an algebraic-rule from it.

4.1.2. Mapping

With the conceptual model and the scenarios defined, the next step is to map them to the user data, which involves the definition of data collection, knowledge representation and modelling technique.

For data collection, a logging mechanism was used to give us access to user data, which allowed us to test potentially suitable modelling techniques. This was established as a temporary solution for storing user data (in our case using log files), that would later be changed to a more efficient solution. This approach enabled us to discuss what was important to capture from user modelling point of view, i.e. the elements in the conceptual model, and to make sure that the needed data is available.

Based on the conceptual model and the scenarios, we then looked at suitable knowledge representations and modelling mechanisms. The fourth scenario meant that the modelling mechanism should be able to diagnose the learner based on partial information, i.e. incomplete strategies.

The nature of the information from the conceptual model and the requirement to diagnose learners

before they complete a task, indicated Case-based Reasoning (CBR) [57] as a potentially good technique for representation and diagnosis. The knowledge representation is outlined below.

A strategy is defined as $S_i = \{C_i, R_i\}$, where C_i represents a set of cases and R_i represents a set of relations between cases of C_i. Each case of C_i includes attributes with their corresponding values, as in columns 2 and 3 of Table 3, which displays the conceptual model. According to their values, there are 2 types of attributes: numeric (values of dimensions) and variables (e.g. shape type, dimension type).

The set of relations R_i is defined as $R_i = \{RA_i, RC_i\}$. RA_i is a set of relations between attributes of cases (value and dependency relations) and RC_i is a set of relations between cases (order relations). A strategy is specific when it does not have dependency relations and is general when it has all the dependency relations required by the task.

In Case-based Reasoning, similarity metrics are used to measure how close the input case is to the stored ones and to retrieve one or several similar cases from the case-base. In our application, the input case is the construction of the user and the case-base includes the strategies that could be used to solve the task. More specifically, the similarity metrics compare a learner's strategy with all the strategies in the case-base. The aim is to identify the most similar one for the purpose of adaptation and personalisation, as outlined in the scenarios.

The similarity metrics used are displayed in Table 6. Different metrics were used for the different types of information: Euclidean distance was used for numeric attributes, string matching for variables and Jaccard's index for relations. The case comparison metrics were aggregated in four metrics to enable

Table 6: Similarity metrics.

	Similarity Metric	
	Cases / Relations	Strategies
Numeric attributes	$D_{IR} = \sqrt{\sum_{j=v+1}^{w}(\alpha_{I_j} - \alpha_{R_j})^2}$	$F_1 = \begin{cases} \frac{z}{\sum_{i=1}^{z} D_{I_i R_i}} & \text{if } \sum_{i=1}^{z} D_{I_i R_i} \neq 0 \\ z & \text{if } \sum_{i=1}^{z} D_{I_i R_i} = 0 \end{cases}$
Variables	$V_{IR} = \frac{\sum_{j=1}^{v} g(\alpha_{I_j}, \alpha_{R_j})}{v}$ $g(\alpha_{I_j}, \alpha_{R_j}) = \begin{cases} 1 & \text{if } \alpha_{I_j} = \alpha_{R_j} \\ 0 & \text{if } \alpha_{I_j} \neq \alpha_{R_j} \end{cases}$	$F_2 = (\sum_{i=1}^{z} V_{I_i R_i})/z$
Relations between attributes	$A_{IR} = \frac{\lvert RA_I \cap RA_R \rvert}{\lvert RA_I \cup RA_R \rvert}$	$F_3 = (\sum_{i=1}^{z} A_{I_i R_i})/y$
Relations between cases	$B_{IR} = \frac{\lvert RC_I \cap RC_R \rvert}{\lvert RC_I \cup RC_R \rvert}$	$F_4 = (\sum_{i=1}^{z} B_{I_i R_i})/z$

α=attribute; 1 to v are variables; $v+1$ to w are numeric
I=Input Strategy; R=Retrieved Strategy
z=minimum number of cases in I or R
y=number of relations between attributes in R

comparison of strategies, as displayed in the third column of Table 6. The strength of similarity between the input strategy and the various stored strategies is defined as the combined similarity of these four measures: $Sim = F1 + F2 + F3 + F4$.

4.1.3. Evaluation

In the following we present outputs of our mechanism for each scenario, using constructions from classroom trials with pupils solving the pond-tiling task in *ShapeBuilder*. In this evaluation we used data from 10 pupils, where each pupil built one construction. The mechanism used the input from log files and its output (i.e. most similar strategy or strategies) was checked by one expert against screen videos that were collected for all pupils.

To test the specific and partly general complete constructions, as well as the partial constructions, snapshots of user's construction were taken at different point during the task. For example, from the 6 complete strategies, snapshots were taken when the constructions were complete, but with no relations between their parts to extract specific strategies. Table 7 presents the distribution of the 38 user strategies according to the pedagogical scenarios.

Table 7: Distribution of user strategies according to scenarios.

Scenario		No of user strategies
Complete	General	6
	Partly general	6
	Specific	6
Mixed		2
Non-systematic		2
Partial		16

The modelling mechanism successfully identified all 38 tested user strategies. The results of this evaluation indicated that the proposed modelling technique is suitable for the purpose of the user model component.

Observations of pupils were used to evaluate the scenarios. First, as outlined above, behaviours belonging to each scenario were observed. Second, we observed that at the very beginning, when the learners are novices, they build specific constructions and only after having a complete construction they start to think about how to make it general. Moreover, after building a specific construction, many learners found it challenging to create the first link between the components of the construction; in addition, some learners create links between some components, but find it difficult to create links between other components. This behaviour comes under the *complete strategies* scenario, and these observations provided two valuable sources of information: (a) they confirmed that it is important to detect *specific*, *partly general* and *completely general*

constructions and (b) it provided valuable information for the adaptation component in terms of generating feedback for these different behaviours.

Other observed behaviours associated with the novice state were: (a) *mixed strategies*, which were adopted because learners did not yet think about generalisation and did not realise the complexities added by this approach when deriving the algebraic-like rule; we found that no pupil was able to extract a general rule from a mixed construction; (b) *non-systematic strategies*, where pupils focused on reproducing the form of the construction, i.e. a pond completely surrounded by tiles, without thinking about generality.

We also noticed that pupils were unsure how to proceed after building some parts of the construction, and only continued with the task after they were given feedback either to encourage them to continue or to specifically tell them to do something similar to what they have already done (e.g. when they only built one top row of tiles, they were given a suggestion to build the bottom row in a similar way). Consequently, the need for detecting *partial constructions* was confirmed.

4.1.4. Learner model v0 relation to system development

The analysis process of the learner model development was influenced by experts' knowledge of the tasks and the requirements for the adaptation module. The expert's knowledge of mathematical generalisation tasks was used in defining the conceptual model, as well as in identifying relevant scenarios that included requirements for the user model module and the adaptation module. The requirements for the two modules are interlinked, as the adaptation module needs diagnosis information from the user model. For example, the identification of partial strategies was identified as a scenario, as this would trigger intelligent support from the adaptation module.

The mapping process was influenced by the user interface of *ShapeBuilder* in term of data collection, knowledge representation and modelling technique. As mentioned earlier, the user data collection was done through log files which were used in the evaluation process. The knowledge representation and modelling technique were influenced by the user interface in terms of defining the set of attributes for cases and finding appropriate similarity measures for those attributes.

The participatory design of this first version entailed the participation of experts and users in the development of the learner model. The experts participated in several ways: (a) they provided information about generalisation tasks and possible solutions, (b) provided information about what is important to identify to enable intelligent support, i.e. the scenarios, and (c) labeled the solutions of the pupils with the most similar strategy. The pupils participated by: (a) providing data for the evaluation of the learner model, by solving

the pond-tiling task using *ShapeBuilder* and (b) providing information on what they found conceptually difficult such as linking shapes, which was used in the validation of the scenarios.

4.2. Learner model v2

This section presents the last iterative version of the learner modelling mechanism. There was one previous version (v1), based on the same system as the one presented for learner model v0, but with different modifications to the user interface. This previous version is not presented because of its similarity to the last version, as well as for the brevity of the paper.

The modifications in the user modelling mechanism were driven by the evolution of the design of the learning environment in general, and of the interface in particular.

Although the grid-based structure of the environment did not change, several details of the interaction design changed, such as using patterns instead of shapes, property lists instead of menus and introducing "two worlds" in the main screen – a student's world and a general world. These changes were introduced based on the feedback from pupils and teachers, and are described in more detail in the following. From user modelling point of view, these changes meant a change in the attributes of cases and an adjustment of the similarity metrics, which are described in sections 4.2.1 and 4.2.2.

In the following, an overview of the new system is given, outlining the changes from *ShapeBuilder*. Like the first version of the system, this version was designed for classroom use and targets pupils of 11 to 14 year-olds. Also, each task involves two main phases: constructing a model and deriving an algebraic-like rule from it. The features of the new version of the system, named *eXpresser* [70], have been informed by studies with pupils and teachers. Several changes took place that are presented below:

1. *eXpresser* allows the construction of patterns rather than shapes; therefore, *eXpresser* is more general than *ShapeBuilder* in terms of what can be constructed.
2. The ShapeList has been removed and property lists have been "attached" to each pattern that enable their creation and the inspection of their properties.
3. *Icon variables* are replaced by the so-called *T-boxes*; they serve the same purpose as the icon variables, but are defined to represent any of the properties of a pattern. Unlike icon variables that made a dependency relation unidirectional, T-boxes define multi-directional relations, i.e. when the variable defined by the T-box changes, the change is reflected in all related properties.
4. Two 'worlds' are included in *eXpresser* – the student's world where the student builds his/her constructions and rules, and the general world where a different instance of the student's construction is

displayed. Also, the construction in the general world can be animated to display various instances of the same construction.

5. To enable the animation of patterns, in *eXpresser* the rules required by the task need to be defined and at least one dependency relations needs to be in place.
6. *eXpresser* supports collaborative activities, as well as individual ones.

Figure 8 illustrates the system, the *property lists* of two patterns (linked to another ones through the use of a T-box) and examples of rules for two instances of the pond-tiling task. The interface includes two windows: (a) the students' world, where the students build their constructions and (b) the general world that displays the same construction with a different value for the variable(s) involved in the task (h and w in this case), and where students can check the generality of their construction by animating their patterns (using the Play button).

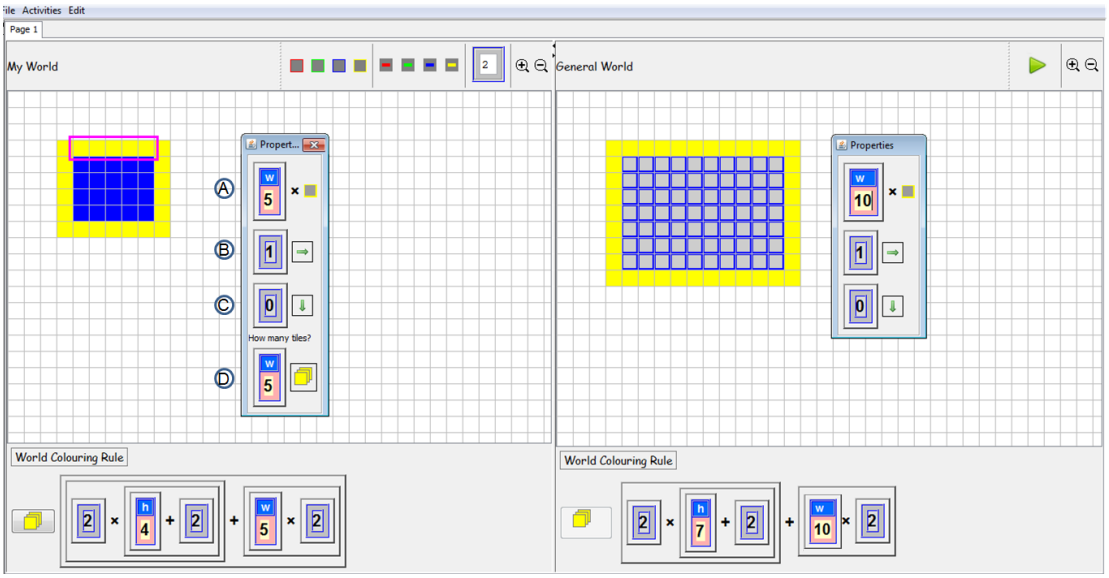


Figure 8: The interface of *eXpresser*. This screenshot includes the display of the the students' world and the general world; the student's construction in the student's world and a different instance for the same construction in the general world; the property list of the top horizontal bar in both worlds; rules for the number of yellow tiles in both worlds.

We illustrate the affordances of *eXpresser* using the pond-tiling task previously introduced for *Shape-Builder* and displayed in the students' world with a 5 by 4 blue (darker colour) pond and in the general world with a 10 by 7 pond. Here we illustrate the 'H' strategy (also displayed in Figure 7c).

The property list of the top horizontal bar is displayed in both worlds. The first property (Ⓐ) specifies the number of iterations of the building-block, i.e. the basic unit of a pattern, which is displayed as an icon; the value for this attribute is set to the value of the width of the pond by using a T-box (that includes

a name and a value); by using a T-box, the two (or more) properties are made dependent, i.e. when the value in the T-box changes in one property, it also changes in the other one(s). The next properties are *move-right* (𝔹), which is set to 1, and *move-down* (ℂ), which is set to 0. The last property (𝔻) establishes the number for colouring all the tiles in the pattern – for this simple pattern the value is the same as the iterations and is also related to the width of the pond through the use of a T-box. The bottom areas of both worlds displays a rule for the number of yellow tiles: $2 * (h + 2) + w * 2$, where h and w stand for the T-boxes used in the property lists of the construction's components; a T-box can be displayed with name only, value only or both. In Figure 8, all T-boxes are displayed with both names and values. If the T-boxes were displayed with names only, the rules in both worlds would be the same, indicating the generality of the solution.

To make a construction general, T-boxes are needed to link the different parts of the construction. Without these links, a construction is *specific*, i.e. it is valid only as a particular instance of the task pattern; a construction can also have some links in place, while others are missing, i.e. the construction is *partly general*. This is essentially the same as in *ShapeBuilder*, except for the replacement of icon variables with T-boxes.

The use of property lists to construct patterns facilitates the derivation of the algebraic-like rule by the presence of the couloring property which refers to the number of tiles needed for certain parts of the construction; the rule is essentially formed by putting together the values of the colouring properties of all parts of a construction.

The following subsections present the last iterative development of the learner model, which is described using the three processes in our proposed methodology: analysis, mapping and evaluation.

4.2.1. Analysis

The analysis in this iteration looked at the changes in the user interface and the implication this changes had on the conceptual model. As the affordances of the user interface changed, the conceptual model was updated to reflect the new way of interacting with the system. The updated conceptual model is presented in Table 8. Each pattern has several properties: iterations, move right, move down and couloring.

We illustrate this version of the conceptual model in Table 9 for the 'H' strategy of the pond tiling task, which is displayed in Figure 8. This also illustrates the concept of "patterns of patterns", where patterns can be constructed by iterating other patterns.

Observations of pupils' interaction with the previous version of the system were used to identify if new scenarios were needed. Several experts looked at new interactive situations and decided that they were

Table 8: Conceptual model for strategies of *eXpresser* tasks.

Component	Properties	Possible Values	Relations
C1	Pattern colour	string	VR1, VR2, etc.
	Each pattern property		DR1, DR2, etc.
	- type:	numeric (n)/ T-box (tb) / numeric expression (ne)/ T-box expression (tbe)	Previous, Next
	- value	numeric	
	width value	numeric	
	height value	numeric	
C2	Pattern colour	string	VR1, VR2, etc.
	Each pattern property		DR1, DR2, etc.
	- type:	n/tb/ne/tbe	Previous, Next
	- value	numeric	
	width value	numeric	
	height value	numeric	
...	....	....	....
Cn	Pattern colour	string	VR1, VR2, etc.
	Each pattern property		DR1, DR2, etc.
	- type:	n/tb/ne/tbe	Previous, Next
	- value	numeric	
	width value	numeric	
	height value	numeric	

already covered in the existing scenarios. Consequently, the scenarios stayed the same as for the previous versions.

4.2.2. Mapping

The mapping process included the revision of data collection, knowledge representation and modelling mechanism. All these changed to reflect the new way pupils interact with the system, i.e. by building patterns instead of shapes. Consequently, all user data that was needed for the conceptual model was stored in this version in a database of user actions.

Two parts of the knowledge representation were updated: the set of attributes and the dependency relations. The set of attributes was updated to reflect the properties of patterns (i.e. iterations, move right, move down, coulouring, width and height). The change in the dependency relation is that the relation is *reciprocal*, while in the previous version the relation was unidirectional (from the dependent part to the independent one). As these are relatively minor changes, the full knowledge representation is not included in this section.

The changes in the knowledge representation triggered changes in the modelling mechanism, and more specifically, in the similarity metrics. Three out of four similarity metrics (F_2, F_3 and F_4) remained the

Table 9: Conceptual model for the 'H strategy' of the pond tiling task in *eXpresser*.

Pattern	Properties	Possible values	Relations
Pond	Made of 2 patterns (P0 and P1)		
P0	Iterations type	tb	VR1: P0_colouring=P0_iterations
	Iterations value	3	DR1: P0_colouring=DEP(P0_iterations)
	Move right type	n	Previous: null
	Move right value	0	Next: P1
	Move down type	n	
	Move down value	1	
	Colouring type	tbe	
	Colouring value	3	
	Colour	blue	
	Width	1	
	Height	3	
P1	Iterations type	tb	VR1: P1_colouring=P0_iterations*P1_iterations
	Iterations value	4	DR1: P1_colouring=DEP(P0_iterations)
	Move right type	n	DR2: P1_colouring=DEP(P1_iterations)
	Move right value	1	Previous: P0
	Move down type	n	Next: P2
	Move down value	0	
	Colouring type	tbe	
	Colouring value	12	
	Width	4	
	Height	3	
P2 (top bar)	Iterations type	tbe	VR1: P2_iterations=P1_iterations
	Iterations value	4	VR1: P2_colouring=P1_iterations
	Move right type	n	DR1: P2_iterations=DEP(P1_iterations)
	Move right value	1	DR2: P2_colouring=DEP(P1_iterations)
	Move down type	n	Previous: P1
	Move down value	0	Next: P3
	Colouring type	tbe	
	Colouring value	4	
	Colour	yellow	
	Width	4	
	Height	1	
P3 (bottom bar)	Properties types and values as in P2 above		VRs and DRs - same as P2 Previous: P2 Next: P4
P4 (left bar)	Iteration type	tbe	VR1: P4_iterations=P0_iterations + 2
	Iterations value	5	VR1: P4_colouring=P0_iterations + 2
	Move right type	n	DR1: P4_iterations=DEP(P0_iterations)
	Move right value	0	DR2: P4_colouring=DEP(P0_iterations)
	Move down type	n	Previous: P3
	Move down value	1	Next: P5
	Colouring type	tbe	
	Colouring value	5	
	Colour	yellow	
	Width	1	
	Height	5	
P5 (right bar)	Properties types and values as in P4 above		VRs and DRs - same as P4 Previous: P4 Next: null

same, F_1 was normalised and the way the four metrics were combined changed. This was due to the change in attributes and how they reflected the structure of a construction. To have a better control over the metrics that influenced the structural similarity, we applied normalisation to the F_1 metric and used weights to emphasize the metrics that reflect the structural similarity.

As the metric for the numeric attributes (F_1) has a different range from the other two similarity metrics, normalisation was applied to have a common measurement scale, i.e. $[0, 1]$. This was done using linear scaling to unit range [2] by applying the following function: $\overline{x} = \frac{x-l}{u-l}$, where x is the value to be normalised, l is the lower bound and u is the upper bound for that particular value. Consequently, as the range of F_1 is $[0, z]$, the normalisation function is: $\overline{F_1} = F_1/z$.

As the structure of the construction is the central aspect that allows identification of strategies, weights were applied to emphasize this aspect. The structure is reflected mostly by the $\overline{F_1}$ metric, and to a lesser degree, by the F_3 metric. Consequently, the similarity between strategies was calculated as $Sim = 6 * \overline{F_1} + F_2 + 2 * F_3 + F_4$.

4.2.3. Evaluation

The modified modelling mechanism was evaluated on unseen user data for each of the scenarios. Data was collected from 36 pupils from classroom use of *eXpresser* to solve a task called stepping-stones [21]. In addition, data was collected from 19 pupils working with *eXpresser* on the pond-tiling task. To increase the data available for testing for partly general constructions and partial constructions, the user data from general/specific constructions was used to extract intermediate steps from the final constructions, i.e. extract the constructions at earlier stages. The distribution of the user strategies by scenario is displayed in Table 10.

Table 10: Distribution of user strategies according to scenarios.

Scenario		No of user strategies
Complete	General	9
	Partly general	5
	Specific	15
Mixed		10
Non-systematic		10
Partial		67

For the first three scenarios, the modelling mechanism successfully identified all user strategies. For the last scenario, we have tested partial strategies with 2 and 3 components. Out of the 67 partial strategies, 40 had 2 components and 27 had 3 components. We calculated the probability of identifying the user strategies from these partial constructions and found the following results: (1) 89% probability of correctly identifying a partial strategy with 2 components and (2) 100% probability of correctly identifying a partial strategy

with 3 components.

Another evaluation study was conducted by asking the experts to rank the similarity of a user's construction to other 3 constructions. Four experts ranked the similarity of 25 user constructions and we compared their ranking with the ranking of the modelling mechanism. A percentage agreement of 90% and a Fleiss kappa of 0.83 were obtained. The disagreement between the experts and the modelling mechanism occurred mainly in relation to non-systematic strategies, which are the most difficult to identify.

In relation to scenarios, similar behaviours of pupils were observed as in the first version, which were described in section 4.1.3. In addition we observed that some pupils who use mixed scenarios were able to generalise their constructions despite the added difficulty involved. This prompted to additional information for the adaptation module to provide feedback on the benefit of elegant strategies.

4.2.4. Learner Model v2 relation to system development

The modifications introduced in the last version of the user model were triggered by the changes in the user interface. This determined changes in the conceptual model, which in turn, prompted the following changes in the knowledge representation and modelling mechanism: (1) new attributes for cases and (2) a new way of aggregating the three similarity metrics.

Although the purpose of the system was the same, the change in the interface, and hence the user interaction, was a significant one. The shift from shapes to patterns led to different attributes, which has the biggest influence on the metric for the numeric attributes. While in the learner model v0 the numeric attributes had the most influence on the strategy similarity without introducing any weighting, in the learner model v2 the introduction of four new numeric properties (iterations, move-right, move down and colouring) led to inconsistency with regards to the influence of the numeric metric on the aggregated similarity. To address this inconsistency, normalisation and weights were used.

Like in the previous versions, experts and users were involved in the participatory design of the last version of the learner model. The experts provided information on: (a) whether the new information collected from users (from the previous version) led to the need to update the current list of scenarios - they decided that all new situations for not previously identified could be classified under one of the existing scenarios, (b) labeled the solutions of pupils with the most similar strategy, and (c) evaluated the learner modeling identification mechanism. The pupils provided data for the evaluation of the learner modelling mechanism by solving two tasks in *eXpresser*.

5. Discussion

Although we illustrated the proposed methodology with a case study in the area of exploratory learning system in which the conceptual model referred to strategies, the methodology is appropriate for other adaptive learning system, where the conceptual model can refer to other common aspects used in current research, such as concepts of the learning domain [18], constraints [65] or competencies/skills [16, 31].

For example, a popular learner modelling approach is the overlay approach (see Section 2.1 and Table 1) in which the principle is that the model of the learner is an overlay of an expert model (i.e. a model of knowledge of an expert). In the following, we outline how the three stages of our methodology would apply to an overlay approach working with concepts of a particular domain:

1. Analysis stage: for this approach, the conceptual model would correspond to the identification of the concepts of the domain overall and for each task, and identification of particular scenarios, which could correspond, for example, to different misconceptions, i.e. mistakes that students typically make in relation to a particular concept. For example, a misconception in the mathematical field is the use of addition when multiplication should be used [47];
2. Mapping stage: at this stage a representation for the concepts and misconceptions of the learning domain needs to be identified based on the conceptual model, scenarios and the interaction design of the system; the modelling technique needs to be decided at this stage as well, and it is likely that the knowledge representation and modelling technique influence each other, as pointed out in Section 3.1. A variety of techniques and representations could be used, such as Bayesian networks [26], fuzzy logic [41] and ontology-based approaches [84];
3. Evaluation stage: this would involve the evaluation of the conceptual model and scenarios in terms of the concepts and misconceptions included, as well as the modelling techniques in terms of how well the learner model reflects the knowledge of the learner. The evaluation will likely lead to changes in the conceptual model, which will trigger another iteration of the methodology.

To further demonstrate how the proposed methodology can be exploited in various domains, we illustrate the steps of the methodology for an area of learner modelling that has received a lot of attention in the last decade, i.e. affect modelling (modelling of emotions and affective states). In terms of user modelling approaches, although overlay models can be used, stereotypes and feature-based models are considered as more suitable. In this domain, the steps of the methodology would involve the following:

1. Analysis stage: the conceptual model corresponds to the emotions or affective states that are of

interest, while the scenarios refer to situations related to the individual emotions or affective states, or their combinations, which should trigger an intervention. For example, a simple intervention is the mirroring of an emotion by an avatar or a robot (e.g. [60, 53]).

2. Mapping stage: in this stage, the method of representation of emotions or affective states needs to be identified; similarly to modelling other learning-related features such as knowledge or goals, the choice of representation is linked to the modelling technique that detects and maintains information on the current (and past) affective states of a learner. This depends on the knowledge elicitation approach used for finding out information about the affective state of a learner. These methods vary from learner/tutor/observers reports [75] to multimodal systems, including a variety of sources such as facial features, gestures, voice and text [33]. In terms of knowledge representation and modelling, network representations, e.g. [25, 74], and machine learning techniques, e.g. [4, 7, 22, 32], are often used.
3. Evaluation stage: this involves the evaluation of the conceptual model (are all affective states of interest included?), the scenarios (are all relevant situations that require intervention covered?) and the knowledge representation and modelling technique (does the model accurately reflect the affective state of a learner?). Changes in one or more of these (conceptual model, scenarios, and representation and modelling technique) will trigger another iteration of the methodology.

Our proposed methodology assumed an iterative development, as this is often the case with system development where users are involved. Moreover, this is especially true with adaptive systems, whose development follow a user-centred approach, in line with the system's purpose, i.e. to offer the users personalised interaction.

In the case study, one of the challenges was to formalise the knowledge elicited from experts, especially for a domain like mathematical generalisation, where the tasks chosen had several equally valid solutions. This led to the need for the conceptual model and the scenarios; the conceptual model covered the information about what the pupils do when they solve the task, while the scenarios covered the situations when they may need guidance. The user studies were helpful in consolidating this formalisation by providing information on how pupils approach the tasks, the processes they go through to reach a solution and where along the way they may need guidance.

The conceptual model facilitates the development of the user model by informing the data collection, i.e. what the system should capture to allow diagnosis. It also facilitates the identification of potentially suitable knowledge representation and modelling techniques. The scenarios provide information about what

the modelling mechanism should be able to diagnose. As this has bearing on the modelling technique, it is best to identify as many of these requirements early.

Users' involvement in the development of the system and the user model in particular, has an interesting bearing on the design of the user model. On one hand, it provides valuable information in terms of how the users approach the tasks and for testing the performance of the user model. On the other hand, it comes with the certainty that the user interaction with the system will change, which will have an implication on the user model. Consequently, the design of the user model needs to account for this expectation of change when weighting the different options for knowledge representation and modelling techniques.

Experts' involvement in the development of the system and the user model is also important. The choice of experts depends on the purpose of the adaptive learning system, and more specifically on its audience. For example, if the system is designed for school children, the experts involved should be teachers. For higher education or professional development, practitioners in the field would be more appropriate. A known issue in many fields is that some problems have multiple solutions and even experts disagree on which is the "best one" – this is often because there is no concept of a "best solution" and forcing experts to choose one leads to confusion for the developers of educational systems. Our case study, although from a field perceived to be precise in its definitions and solutions, addressed problems with multiple equally valid solutions. Consequently, we hope that our case study is useful for researchers in other fields where several perspectives on the same problem are important.

In the presented case study we found two principles that ensured a relatively smooth transition between the various versions of the user model. These principles are early identification of scenarios and modelling technique flexibility. Identifying all or most relevant scenarios at the beginning has an influence on the choice and evaluation of the modelling technique. The more is known at the beginning about the requirements for the modelling technique, the easier it is to choose a method that can address all requirements.

For example, in the case study, if the identification of partial strategies scenario would have been identified in a later iteration, a different technique may have been chosen in the first iteration that would have not needed to deal with partial constructions. Since the choice of using CBR was influenced by this particular requirement, CBR may have not been chosen at that point. This would have had consequences for the subsequent iterations. For example, if this scenario would have been identified in the second (or later) iterations, the chosen technique in the first iteration may have not been able to address this requirement, and, consequently, a new suitable modelling technique would have been required. Consequently, the user model development from the first version would have been lost, and more effort would have been needed to

build a new modelling mechanism rather than make changes to a previous version. Therefore, it is important to identify scenarios early to prevent such problems.

The other principle is flexibility of the modelling technique, which has an influence on the amount of effort needed to adapt it to changes in the user interface. In our case study, CBR had this flexibility due to its case structure and similarity metrics; thus CBR allowed partial diagnosis, which was one of the essential requirements. The advantage of CBR, in our case, is that we could modify the contents of cases and the similarity metrics to account for changes in the interface, while at the same time preserving diagnosis principles, i.e. comparing the construction of a user (the input strategy) with stored constructions (strategies from the case-base). Due to the iterative development of the system, we expected changes in the user interface, which, in our situation, corresponded to changes in the attributes of a case. The changes in the attributes triggered testing of the similarity metrics and adjustments to ensure correct diagnosis.

Another aspect indirectly argued in this paper is that there is an advantage in separating the development of the user model and the other components of the system, including the adaptation module. Some of these advantages are re-usability, easier testing and validation, and more reliable and tractable progress of the system development. Thus, developing the user model for the initial version of the system is not lost for the subsequent versions (re-usability); testing modules separately allows identifications of problems more easily, e.g. if diagnosis and adaptation modules would be tested together and the users do not find the feedback useful, we do not know if this is due to misdiagnosis or unhelpful feedback; finally, progress of system development is easier to track when progress on individual components can be tracked. More arguments for such separation of components can be found in [43].

Despite the separation of the different models, the user studies were planned in such a way as to facilitate the collection of all the information needed by the different components of the system. This was done because access to pupils in UK schools in not easy, as already documented in the literature [48], but also because it facilitated the coordination between all parties involved in the development of the system.

6. Conclusions

In this paper we presented an iterative methodology for user model development called Participatory Learner Modelling Design. The methodology proposes three processes, i.e. analyse, map and evaluate, which are repeated in several iterations that take place in parallel to system development. Our methodology also looked at the link between the user model development and the iterative development of the system in general, and the user interface in particular.

The challenges of designing a user model iteratively, with the users' involvement and in parallel to system development, are both of technical, as well as pedagogical nature: (a) identify information about the tasks and the students' interactions that need to be captured and formalised to allow diagnosis (and feedback); (b) identify an appropriate knowledge representation and modelling mechanism, with the knowledge that the interaction design of the system is likely to change; (c) test and refine the user model iteratively in parallel to system development.

To summarise, the methodology we described helped us systematically address the design of the user model component in the knowledge that the system will evolve and that the user interface may look very different from the initial versions. We hope this would be useful for other researchers involved in user models development where the system is developed iteratively and with users' involvement.

References

[1] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013. ISSN 0924-1868.

[2] S. Aksoy and R. Haralick. Feature normalisation and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, 22:563–582, 2001.

[3] C. Allison, S. A. Cerri, P. Ritrovato, A. Gaeta, and M. Gaeta. Services, semantics, and standards: Elements of a learning grid infrastructure. *Applied Artificial Intelligence*, 19(9-10):861–879, 2005.

[4] N. Altrabsheh, M. Cocea, and S. Fallahkhair. Predicting students emotions using machine learning techniques. In *The 17th International Conference on Artificial Intelligence in Education*, volume 9112 of *LNAI*, pages 538–541. Springer, 2015. doi: 10.1007/978-3-319-19773-956.

[5] D. Benyon and D. Murray. Applying user modeling to human-computer interaction design. *Artificial Intelligence Review*, 7(3-4):199–225, 1993. ISSN 0269-2821. doi: 10.1007/BF00849555. URL `http://dx.doi.org/10.1007/BF00849555`.

[6] R. Bernabei and J. Power. Designing together: end-user collaboration in designing and personalised product. In *10th European Academy of Design Conference*, pages 1 – 12. 17 - 19 April , Gothenburg, Sweden, 2013.

[7] S. Bernardini, K. Porayska-Pomsta, and H. Sampath. Designing an intelligent partner for social communication in autism. In *Proceedings of the Ninth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI Press, 2013.

[8] S. Berry, S. Fazzio, Y. Zhou, B. Scott, and L. Francisco-Revilla. Netflix recommendations for groups. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–3., 2010.

[9] S. Bødker. Scenarios in user-centred design – setting the stage for reflection and action. *Interacting with Computers*, 13 (1):61 – 75, 2000. ISSN 0953-5438.

[10] J. Bollen. Group user models for personalized hyperlink recommendations. In P. Brusilovsky, O. Stock, and C. Strapparava, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 1892 of *Lecture Notes in Computer Science*, pages 38–50. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67910-3.

[11] C. Boyle and A. Encarnacion. Metadoc: An adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, 4(1):1–19, 1994. ISSN 0924-1868. doi: 10.1007/BF01142355. URL http://dx.doi.org/10.1007/BF01142355.

[12] P. Brusilovsky. A framework for intelligent knowledge sequencing and task sequencing. In C. Frasson, G. Gauthier, and G. McCalla, editors, *Intelligent Tutoring Systems*, volume 608 of *Lecture Notes in Computer Science*, pages 499–506. Springer Berlin Heidelberg, 1992. ISBN 978-3-540-55606-0. doi: 10.1007/3-540-55606-0_59. URL http://dx.doi.org/10.1007/3-540-55606-0_59.

[13] P. Brusilovsky. Student model centered architecture for intelligent learning environments. In *Proceedings of Fourth International Conference on User Modeling*, pages 31–36. User Modeling Inc., 1994.

[14] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[15] P. Brusilovsky and E. Millan. User models for adaptive hypermedia and adaptive educational systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 3–53. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-72078-2.

[16] S. Bull, B. Wasson, M. Kickmeier-Rust, M. D. Johnson, E. Moe, C. Hansen, G. Meissl-Egghart, and K. Hammermuller. Assessing english as a second language: From classroom data to a competence-based open learner model. In *International Conference on Computers in Education*, 2012.

[17] F. Carmagnola, F. Cena, and C. Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, 21(3):285–331, 2011. ISSN 0924-1868.

[18] C. Carmona and R. Conejo. A learner model in a distributed environment. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 353–359. Springer, 2004.

[19] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715 – 4729, 2013. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2013.02.007. URL http://www.sciencedirect.com/science/article/pii/S095741741300122X.

[20] M. Cocea and G. D. Magoulas. Hybrid model for learner modelling and feedback prioritisation in exploratory learning. *International Journal of Hybrid Intelligent Systems*, 6(4):211–230, 2009. ISSN 1448-5869.

[21] M. Cocea and G. D. Magoulas. User behaviour-driven group formation through case-based reasoning and clustering. *Expert Systems with Applications*, 39(10):8756 – 8768, 2012. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2012.01.205. URL http://www.sciencedirect.com/science/article/pii/S0957417412002333.

[22] M. Cocea and S. Weibelzahl. Log file analysis for disengagement detection in e-learning environments. *User Modeling and User-Adapted Interaction*, 19(4):341–385, 2009.

[23] M. Cocea and S. Weibelzahl. Disengagement detection in online learning: Validation studies and perspectives. *Learning Technologies, IEEE Transactions on*, 4(2):114–124, 2011. ISSN 1939-1382. doi: 10.1109/TLT.2010.14.

[24] M. Cocea, S. Gutierrez-Santos, and G. Magoulas. Case-based reasoning approach to adaptive modelling in exploratory learning. In T. Watanabe and L. Jain, editors, *Innovations in Intelligent Machines - 2*, volume 376 of *Studies in Computational Intelligence*, pages 167–184. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-23189-6. doi: 10.1007/978-3-642-23190-2_12. URL http://dx.doi.org/10.1007/978-3-642-23190-2_12.

[25] C. Conati. Probabilistic assessment of user's emotions in educational games. *Applied Artificial Intelligence*, 16(7-8):555–575, 2002.

[26] C. Conati, A. Gertner, and K. Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4):371–417, 2002.

[27] T. Conlon and H. Pain. Persistent collaboration: a methodology for applied AIED. *International Journal of Artificial Intelligence in Education*, 7(3):219–252, 1996.

[28] A. I. Cristea and A. de Mooij. LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators. In *WWW03 (The Twelfth International World Wide Web Conference), Alternate Track on Education.*, 2003.

[29] A. I. Cristea, R. Carro, and C. D. Stewart. Advances in authoring of adaptive web-based systems. *Journal of Universal Computer Science*, 16(19):2754–2755, 2010. Special Issue on Advances in Authoring of Adaptive Web-based Systems.

[30] I. Crnkovic, M. Chaudron, and S. Larsson. Component-based development process and component lifecycle. In *Software Engineering Advances, International Conference on*, pages 44–44, 2006. doi: 10.1109/ICSEA.2006.261300.

[31] M. C. Desmarais and R. S. d Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[32] S. D'Mello and A. Graesser. Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence*, 23(2):123–150, 2009.

[33] S. K. D'mello and J. Kory. A review and meta-analysis of multimodal affect detection systems. *ACM Computing Surveys (CSUR)*, 47(3):43, 2015.

[34] A. Druin. Cooperative inquiry: developing new technologies for children with children. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 592–599, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: http://doi.acm.org/10.1145/302979.303166.

[35] A. Druin. The role of children in the design of new technology. *Behaviour and Information Technology*, 21(1):1–25, 2002.

[36] J. Eklund and P. Brusilovsky. The value of adaptivity in hypermedia learning environments: A short review of empirical evidence. In *Proceedings of Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext*, volume 98, pages 11–17, 1998.

[37] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The physiology of the grid. In *Grid computing: making the global infrastructure a reality*, pages 217–249. John Wiley & Sons, 2003.

[38] C. Gaffney, D. Dagger, and V. Wade. Authoring and delivering personalised simulations an innovative approach to adaptive elearning for soft skills. *Journal of Universal Computer Science*, 16(19):2780–2800, 2010.

[39] K. Go and J. M. Carroll. The blind men and the elephant: views of scenario-based system design. *Interactions*, 11:44–53, November 2004. ISSN 1072-5520.

[40] J. Good and J. Robertson. CARSS: A framework for learner-centred design with children. *International Jounal of Artificial Intelligence in Education*, 16(4):381–413, 2006. ISSN 1560-4292.

[41] M. Grigoriadou, H. Kornilakis, K. A. Papanikolaou, and G. D. Magoulas. Fuzzy inference for student diagnosis in adaptive educational hypermedia. In *Methods and applications of artificial intelligence*, pages 191–202. Springer, 2002.

[42] T. Gross, J. Masthoff, and C. Beckmann, editors. *Workshop on Group Recommender Systems: Concepts, Technology, Evaluation (GroupRS)*, 2013. At the 21th Conference on User Modeling, Adaptation and Personalization (UMAP 2013).

[43] S. Gutierrez-Santos, M. Mavrikis, and G. D. Magoulas. A separation of concerns for engineering intelligent support for exploratory learning environments. *Journal of Research and Practice in Information Technology*, 44(3):347–360, 2012.

[44] L. Healy and C. Hoyles. Visual and symbolic reasoning in mathematics: Making connections with computers? *Mathematical Thinking and Learning*, 1(1):59–84, 1999. doi: http://10.1207/s15327833mtl0101_3.

[45] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. Wilamowitz-Moellendorff. GUMO – the general user model ontology. In L. Ardissono, P. Brna, and A. Mitrovic, editors, *User Modeling 2005*, volume 3538 of *Lecture Notes*

in Computer Science, pages 428–432. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-27885-6.

[46] M. Hertzum. Making use of scenarios: a field study of conceptual design. *International Journal of Human-Computer Studies*, 58(2):215 – 239, 2003. ISSN 1071-5819.

[47] V.-L. Holmes, C. Miedema, L. Nieuwkoop, and N. Haugen. Data-driven intervention: Correcting mathematics students' misconceptions, not mistakes. *Mathematics Educator*, 23(1):24–44, 2013.

[48] S. Hossain and L. Brooks. Fuzzy cognitive map modelling educational software adoption. *Computers & Education*, 51: 1569–1588, December 2008. ISSN 0360-1315.

[49] M. Huhns and M. Singh. Service-oriented computing: key concepts and principles. *Internet Computing, IEEE*, 9(1): 75–81, 2005. ISSN 1089-7801. doi: 10.1109/MIC.2005.21.

[50] T. Hurley and S. Weibelzahl. Using MotSaRT to support on-line teachers in student motivation. In E. Duval, R. Klamma, and M. Wolpers, editors, *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 101–111. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-75194-6. doi: 10.1007/978-3-540-75195-3_8. URL `http://dx.doi.org/10.1007/978-3-540-75195-3_8`.

[51] P. Jalote, A. Palit, P. Kurien, and V. Peethamber. Timeboxing: a process model for iterative software development. *Journal of Systems and Software*, 70(1-2):117 – 127, 2004. ISSN 0164-1212.

[52] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72078-2.

[53] H. Jones, N. Sabouret, I. Damian, T. Baur, E. Andr, K. Porayska-Pomsta, and P. Rizzo. Interpreting social cues to generate credible affective reactions of virtual job interviewers. In *IDGEI (Intelligent Digital Games for Empowerment and Inclusion) Workshop*, 2014.

[54] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3):885–899, 2011. doi: http://dx.doi.org/10.1016/j.compedu.2010.11.001.

[55] A. Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2):49–63, 2001. ISSN 0924-1868. doi: 10.1023/A:1011187500863. URL `http://dx.doi.org/10.1023/A%3A1011187500863`.

[56] A. Kobsa. Generic user modeling systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 136–154. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72078-2.

[57] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc., 2nd edition, 1993.

[58] G. Kotonya and I. Sommerville. *Requirements Engineering : Processes and Techniques*, chapter Interactive systems specification, pages 215–250. Wiley, 1998.

[59] C. Larman and V. R. Basili. Iterative and incremental development: A brief history. *Computer*, 36:47–56, June 2003. ISSN 0018-9162.

[60] C. Liu, K. Conn, N. Sarkar, and W. Stone. Online affect detection and robot behavior adaptation for intervention of children with autism. *Robotics, IEEE Transactions on*, 24(4):883–896, Aug 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2001362.

[61] E. Mangina and J. Kilbride. Utilizing vector space models for user modeling within e-learning environments. *Computers & Education*, 51(2):493 – 505, 2008. ISSN 0360-1315.

[62] J. Masthoff. Group recommender systems: Combining individual models. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 677–702. Springer US, 2011. ISBN 978-0-387-85819-7.

[63] M. Mavrikis and S. Gutierrez-Santos. Not all wizards are from oz: Iterative design of intelligent learning environments by communication capacity tapering. *Computers & Education*, 54(3):641 – 651, 2010. ISSN 0360-1315. doi: http://dx.doi.org/10.1016/j.compedu.2009.08.033. URL `http://www.sciencedirect.com/science/article/pii/S0360131509002644`. Learning in Digital Worlds: Selected Contributions from the CAL 09 Conference.

[64] L. Michaud and K. McCoy. Empirical derivation of a sequence of user stereotypes for language learning. *User Modeling and User-Adapted Interaction*, 14(4):317–350, 2004. ISSN 0924-1868.

[65] A. Mitrovic. Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2):39–72, 2012. ISSN 0924-1868. doi: 10.1007/s11257-011-9105-9. URL `http://dx.doi.org/10.1007/s11257-011-9105-9`.

[66] M. J. Muller. *Human-computer interaction: Development process*, chapter Participatory design: the third space in HCI, pages 165–185. 2003.

[67] C. H. Muntean, G.-M. Muntean, J. McManis, and A. I. Cristea. Quality of experience-LAOS: create once, use many, use anywhere. *International Journal of Learning Technology*, 3(3):209–229, 2007.

[68] B. A. Nardi. The use of scenarios in design. *SIGCHI Bulletin*, 24:13–14, October 1992. ISSN 0736-6906.

[69] M. Nebeling, M. Speicher, and M. Norrie. W3Touch: Metrics-based web page adaptation for touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2311–2320, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2481319. URL `http://doi.acm.org/10.1145/2470654.2481319`.

[70] R. Noss, A. Poulovassilis, E. Geraniou, S. Gutierrez-Santos, C. Hoyles, K. Kahn, G. D. Magoulas, and M. Mavrikis. The design of a system to support exploratory learning of algebraic generalisation. *Computers & Education*, 59(1):63 – 81, 2012. ISSN 0360-1315.

[71] D. Pagano and B. Brügge. User involvement in software evolution practice: a case study. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 953–962, Piscataway, NJ, USA, 2013. IEEE Press. ISBN 978-1-4673-3076-3.

[72] C. Paliouras, G.; Karkaletsis V.; Papatheodorou and C. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In J. Kay, editor, *UM99 User Modeling: Proceedings of the Seventh International Conference.*, pages 169–178. Wien, New York (NY): Springer, 1999.

[73] D. Pearce, M. Mavrikis, E. Geraniou, and S. Gutierrez. Issues in the design of an environment to support the learning of mathematical generalisation. In P. Dillenbourg and M. Specht, editors, *Times of Convergence. Technologies Across Learning Contexts*, volume 5192 of *Lecture Notes in Computer Science*, pages 326–337. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-87604-5. doi: 10.1007/978-3-540-87605-2_37. URL `http://dx.doi.org/10.1007/978-3-540-87605-2_37`.

[74] K. Porayska-Pomsta, K. Anderson, I. Damian, T. Baur, E. André, S. Bernardini, and P. Rizzo. Modelling users affect in job interviews: Technological demo. In S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro, editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 353–355. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38843-9. doi: 10.1007/978-3-642-38844-6_37. URL `http://dx.doi.org/10.1007/978-3-642-38844-6_37`.

[75] K. Porayska-Pomsta, M. Mavrikis, S. D'Mello, C. Conati, and R. S. Baker. Knowledge elicitation methods for affect modelling in education. *International Journal of Artificial Intelligence in Education*, 22(3):107–140, 2013.

[76] P. Ritrovato, C. Allison, S. Cerri, T. Dimitrakos, M. Gaeta, and S. Salerno, editors. *Towards the Learning Grid*, volume 127 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2005.

[77] M. Scaife and Y. Rogers. Kids as informants: telling us what we didn't know or confirming what we knew already? In A. Druin, editor, *The design of children's technology*, pages 27–50, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-507-X.

[78] N. R. Shadbolt and P. R. Smart. Knowledge elicitation. In J. R. Wilson and S. Sharples, editors, *Evaluation of Human Work*. CRC Press, Boca Raton, Florida, USA., 4th edition, 2015.

[79] V. J. Shute, E. G. Hansen, and R. G. Almond. You can't fatten a hog by weighing it – or can you? Evaluating an assessment for learning system called ACED. *International Journal of Artificial Intelligence and Education*, 18(4):289–316, 2008.

[80] D. H. Sleeman. UMFE: A user modelling front-end subsystem. *International Journal of Man-Machine Studies*, 23(1):71 – 88, 1985. ISSN 0020-7373.

[81] E. Soloway, M. Guzdial, and K. E. Hay. Learner-centered design: the challenge for HCI in the 21st century. *Interactions*, 1(2):36–48, 1994. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/174809.174813.

[82] E. Soloway, S. L. Jackson, J. Klein, C. Quintana, J. Reed, J. Spitulnik, S. J. Stratford, S. Studer, S. Jul, J. Eng, and N. Scala. Learning theory in practice: Case studies of learner-centered design. In *Proceedings of CHI96*, pages 189–196. ACM Press, 1996.

[83] I. Sommerville. *Software Engineering*. Pearson Education Inc., 2010.

[84] S. Sosnovsky and D. Dicheva. Ontological technologies for user modelling. *International Journal of Metadata, Semantics and Ontologies*, 5(1):32–71, 2010.

[85] M. Specht and A. Kobsa. Interaction of domain expertise and interface design in adaptive educational hypermedia. In P. Brusilovsky and P. de Bra, editors, *Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, page 8993, 1999. Toronto and Banff, Canada,.

[86] K. Stefanidis, N. Shabib, K. Nrvg, and J. Krogstie. Contextual recommendations for groups. In S. Castano, P. Vassiliadis, L. Lakshmanan, and M. Lee, editors, *Advances in Conceptual Modeling*, volume 7518 of *Lecture Notes in Computer Science*, pages 89–97. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33998-1.

[87] F. Tarpin-Bernard and H. Habieb-Mammar. Modeling elementary cognitive abilities for adaptive hypermedia presentation. *User Modeling and User-Adapted Interaction*, 15(5):459–495, 2005. ISSN 0924-1868. doi: 10.1007/s11257-005-2529-3. URL `http://dx.doi.org/10.1007/s11257-005-2529-3`.

[88] E. Wenger. *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. ISBN 0-934613-26-5.

[89] N. Winters and Y. Mor. IDR: A participatory methodology for interdisciplinary design in technology enhanced learning. *Computers & Education*, 50(2):579–600, 2008. ISSN 0360-1315. doi: http://dx.doi.org/10.1016/j.compedu.2007.09.015.

[90] H. Wu. *A reference architecture for Adaptive Hypermedia Applications*. PhD thesis, Technische Universiteit Eindhoven, 2002.

[91] Z. Yu, X. Zhou, Y. Hao, and J. Gu. TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1):63–82, 2006. ISSN 0924-1868.

[92] A. Zimmermann, M. Specht, and A. Lorenz. Personalization and context management. *User Modeling and User-Adapted Interaction*, 15(3-4):275–302, 2005. ISSN 0924-1868.