CO-DESIGN FOR CONCEPTUAL SPACES: AN AGILE DESIGN METHODOLOGY FOR M-LEARNING

David E. Millard, Sue J Faulds*, Lester Gilbert, Yvonne Howard,
**Dan Sparks, Gary B. Wills, and Pei Zhang,

School of Electronics and Computer Science, University of Southampton, UK
{ dem, lg3, ymh,gbw, pz}@ecs.soton.ac.uk}
**School of Nursing and Midwifery, University of Southampton, UK S.J.Faulds@soton.ac.uk*
***Thames Valley University, Dan.Sparks@tvu.ac.uk*

ABSTRACT

M-learning tools function in a number of different spaces, including physical contexts (visiting, at home, at work, etc.), and application areas (location-based information, communication, in-situ recording, etc). We have been exploring m-learning tools that operate in a conceptual space, in our case mobile tools to assist nursing students on placement. These tools are required to connect the user's placement experience with the competency models required by the nursing professional bodies and National Health Service. Designing for conceptual spaces requires developers to work closely with domain specialists in order to model the space correctly, and to create functionally valuable tools. In this paper we present an agile co-design approach to developing mobile tools, which focuses on the requirements and analysis stage. This is largely ignored in current methodologies although it is critical for mobile systems working in conceptual spaces. We demonstrate the use of this methodology through the case study of our nursing placement project.

KEYWORDS

M-learning, co-design, design methodology, healthcare, learning placement

1. INTRODUCTION

In the last decade we have seen many types of m-learning tools, from simple systems that allow access to existing content and behavior on-the-move (Flynn et al. 2000) to more targeted applications, that take advantage of the mobility or locality of the applications users, for example, to provide location-based information (Abowd at al, 1997), or to support fieldtrips (Kravcik et al. 2004). In this paper we are focusing on a different type of application, one that requires the user to navigate, understand and engage with a conceptual space. Typically these spaces are a sophisticated model of some sort; examples include:

- A social space – a model of people, their affiliations, relationships and communications.
- A knowledge or domain map – a network of concepts within a particular domain and their relationships (perhaps including a taxonomy)
- A competency framework – descriptions of skills, abilities, and the relationships between them

The challenge for developers is that working with conceptual spaces requires expert knowledge that is often domain specific. Unfortunately, domain experts are unlikely to have experience of mobile tools and applications, and may not fully appreciate the potential or the limitations of the technology. This creates a tension in the design stages of development.

In this paper we present a co-design methodology that involves domain experts as first-class members of the design team. We describe a number of techniques that brings the design team together, and helps them to converge on a joint understanding of the conceptual space, and focus on tools to tackle real problems in the domain. We have done this by joining techniques found in HCI (personas, scenarios, and storyboarding) with agile software development techniques (iterative and incremental delivery) and lightweight software engineering practice (use cases, activity and iteration diagrams).

2. RELATED WORK

There have been several mobile projects that have developed systems to access information. For example, the Nightingale Tracker (Sloan and Delahoussaye, 2003) used a central database and provides data and communication tools built around it, and the Chawton House project used a sophisticated orchestration engine to drive an outdoors learning experience based on proprietary activity cards (Weal et al, 2006). However, the whole-system approach is inherently heavyweight and inflexible, and has led to criticisms that mobile learning is technology-led and pedagogically naïve (McMillan, 2005). Standalone tools such as the "ME" mobile workbook (de Crom, 2005), or the cell-phone English vocabulary trainer (Collins, 2005) offer a simpler alternative that fit with existing practices more flexibly, but these lack the integration qualities of the whole-system approach.

Design methodologies are therefore required to drive the creation of novel tools, but ensure that they are both useful and sustainable in practice. Many methodologies and models for the design of information and e-learning systems take a layered approach, separating design issues to allow independence (Wills et al, 2003): mapping the domain (in terms of its structure, content, work flow, *etc)*, analysing the associations and relation in that domain, and presenting the information to appropriate users.

Parsons at el. (2006) examined m-learning in particular, and identify four design concerns:

- **generic mobile environment issues** – such as communication support and the device interface
- **learning context** – including the roles of users, the collaborations and activities
- **learning experiences** – how is the learning structured, e.g. through cinematic or game metaphors
- **learning objectives** –the desired goal, for example improved skills or social abilities

These four design concerns are presented as a design framework, but are demonstrated as an analysis framework with which to describe and compare m-learning experiences.

Our approach is much more of a design methodology, and as such is similar to the socio-cognitive engineering approach proposed by Sharples et al. (2002). Socio-cognitive engineering is wary of user-centred design as users are not always able to articulate their own working pattern and methods; instead the method seeks to develop a theory-based framework of user's underlying cognitive and social processes. In practical terms this requires two studies: an investigation into how user activities are performed in their normal contexts, and a theoretical study of the underlying cognitive and social processes.

Our co-design methodology is more lightweight, and does not attempt to build a theoretical model of user's practise, but instead works closely with users as part of the design team in an agile way, in order to build system to support quite subtle requirements – something that is essential for tools in a conceptual space.

3. THE METHODOLOGY

An overview of the four stages in the process is shown in Figure 1. Each stage is supported through workshops and design meetings, attended by both the development and domain experts in the design team. There is a natural flow from stage one through to stage four, however the design methodology is a natural cycle, and design teams will typically undergo several iterations of the last three stages. Each stage, and the techniques used in that stage, is explained in the following four sub-sections.

3.1. Shared Understanding

The methodology begins in the first stage with an effort to create a *Shared Understanding* of the design space within the design team, typically this means that developers and managers learn about the values of the application domain, and that domain experts are introduced to the technology, its scope, potential and limitations. The intention is not to bring everyone in the team to the same level of expertise (this is not possible in sophisticated domains) but to enable an informed conversation to occur in the other stages.

Personas and scenarios are a lightweight method for capturing and recording the requirements of a system from an end user's viewpoint (Cooper and Reimann, 2003). A persona describes an end user in some detail; their background, job function, and situation in the organization. Scenarios are textual descriptions of how a persona interacts with the system and other personas when using a system. The scenarios are independent of any technology and they may represent either current practice or improved practice (as-is or to-be). In stage one they will typically reflect the domain 'as-is'.

Technology Show-and-Tell. It is important that all members of the design team are familiar with both the current technology being used in the domain, and with new technologies that could create new opportunities. We have found that a show-and-tell event is a useful way of bring developers and users together (often for the first time), breaking down social barriers, and creating a common vocabulary for the design team to move forward. Sometimes developers will have to visit key locations to understand the context of technology use, but otherwise a workshop can be held that is focused around a number of key props. Props can include pen and paper based systems, as well as electronic and computing devices.

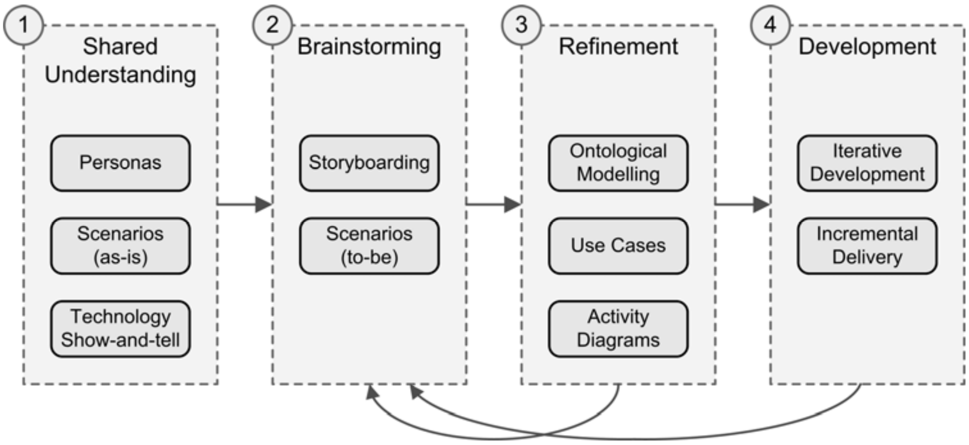


Figure 1: Overview of Agile Co-design Methodology

3.2. Brainstorming

The second stage involves *Brainstorming* ideas for new applications and tools. It relies on the shared understanding established in stage one, and builds a number of initial design artefacts based on the common vocabulary, in particular new scenarios and storyboards of potential applications.

Scenarios (to-be) and Story boarding. A second set of 'to-be' scenarios can be written to capture ideas created by the group. These are textual descriptions that can draw on the same personas used in stage one. The 'to-be' scenarios describe how the personas might interact with potential applications to fulfil an existing need. Using the scenarios, a number of story boards can be created to represent the user interface design (UI) of a given tool. This is a standard technique used in HCI development and is very effective when used in a participatory (or co-design) process. Both the end users and developers (HCI experts) are engaged with designing the UI. During this process, the scenarios can be clarified and modified if required.

3.3. Refinement

The third stage is one of *Refinement*, where the informal ideas captured in stage two are converted into a more concrete set of software requirements and specifications. We take an agile approach, and use lightweight documentation as a means to drive development, rather than as a passive record of activity. We have found three formalisms particularly useful: ontological modelling, use cases and activity diagrams.

Ontological Modelling - Identifying key resources and mapping the relationships between them is a significant part of any co-design process involving conceptual spaces. Often domain experts will not be aware of what type of structures exists within the conceptual space of their domain (for example, is it a taxonomy or just a hierarchy, do richer relationships exist, if so what are the constraints?). In fact, key information structures may have evolved rather chaotically, and modelling them may be a useful point of reflection for domain experts. Ontological modelling is expressive, makes no assumptions about the underlying information models of the domain, and can be easily communicated to domain experts in the form of entity-relationship diagrams.

UML Use Cases and Activity Diagrams - Use Cases are an excellent high-level (and implementation independent) starting point for describing functionality in the context of a given system and user. We use standard UML 2.0 use cases, consisting of a use case diagram, with success scenarios for each case. A brief narrative description is held alongside the diagram as a whole, as well as for each individual use case. From an agile point of view they are effective because they are relatively informal, yet help to define and capture a problem space in detail that can be understood by the whole team, including the end users.

3.4. Development

The fourth stage is *Development*, where all the design artefacts created in stages two and three are used to drive the creation of the tools. We use agile software engineering practises based around iterative development and incremental deployment, we assume that other activities key to the software engineering process (such as architecture design, testing and evaluation) occur within these key principles.

Iterative Development and Incremental Deployment - Agile methods are a number of software development methods which were proposed in the mid 1990s as a reaction to the limitations of traditional software development methodologies Although these methods vary in practice, they share common principles like delivering working software frequently within a short timescale, close communication within the team and with the customer, simplicity and programming over documenting (Larman, 2004). We have applied the spirit of these principles to the earlier stages of co-design, but they are especially important in the final development stage. Development should be focused in a number of small iterations, with the design team (including the users) reviewing the progress of each iteration in a design meeting. The development is shadowed by an incremental deployment of the tool, which first engages users with a simple (but stable) tool from early iterations, and then gradually introduces new functionality. In some cases this will require the team to revisit earlier stages, to reassess assumptions and revise requirements.

4. CASE STUDY

In this section we present a case study of using this agile co-design methodology to help create an m-learning tool for nursing students on placement. This is a conceptual tool as the placement students are constantly asked to reflect on their practise based on a complex competency model that relates skills, proficiencies and attitudes. The Placement Learning and Assessment Toolkit (mPLAT) project aimed to provide a mobile learning toolkit to support practice based learning, mentoring, and assessment to these nursing students. Our belief going into the co-design process was that practice-based learning and the mentoring process would be improved with tools which connected the student in-situ with the competency model against which they were being assessed, and were required to learn.

4.1. Motivation

The following scenario illustrates the problems and the need for such a toolkit:

Pre-registration nursing students spend 50% of their 3-year programme in clinical practice undertaking a series of placements in different areas of the healthcare system. Students are supported by mentors for the duration of their placement. Mentors assess the students' competence in practice against a set of learning outcomes detailed in the practice assessment booklet or practice portfolio. These are summative assessments which students are required to pass in order to register as a nurse at the end of their programme. Students are expected to complete a preliminary, an interim and a final interview with their mentor. The interim interview is crucial as it is at this point that the student who is failing to progress is likely to be identified and *action plans can be put into place. This good practice feature of induction, interim, and final assessment is common to most educational situations where students experience work-based learning situations.*

Issues around ensuring that students are fit for practice at the point of registration were brought home recently following a report by Duffy (2004) which found that mentors were reluctant to fail students due to a number of factors, including lack of confidence, concerns over personal consequences (for student and self), and leaving it too late to implement formal procedures (the preliminary and interim interviews missed or undertaken so late that action plans to assist a student who is not progressing cannot be implemented).

4.2. The Co-Design Process

The co-design process was initiated with a number of workshops. We invited domain experts and stakeholders to join the project team. The main purpose of the first session was bring the co-design team up to speed on our individual expert areas, and to allow us to exchange knowledge so that we built a shared understanding of the domain and technologies in preparation for the following two co-design sessions. The second co-design workshop focused on a brainstorming exercise to facilitate thinking about small novel applications that could help solve some of those problems identified in the first workshop. The third co-design workshop was a refinement process, where we selected three candidate applications and explored their requirements in more detail. It became clear at an early stage that the main concern of the nursing team was in trying to connect student's practice with the conceptual competency framework used by the School of Nursing. As a result we identified a number of potential tools based around this concern:

1. **Profile Placement Tool** – that would provide guidance for how the domain and competency maps to the experiences (opportunities) offered in the placement area (e.g. care delivery in a medical ward).
2. **Learning Contract Builder** – a tool that would draw on the student and mentor's experience and the placement profile to create an action plan concerning what the student will achieve, how they will achieve it, what evidence is required and which learning resources are appropriate

4.2.1. Personas and Scenarios

Personas and scenarios were actually written by the end users, with a little guidance. In addition to the different institutions, there were also the requirements from the professional body (Nursing and Midwifery Council) and the British National Health Service (NHS) to capture. So we developed a number of personas and scenarios for each role. Student surveys have identified the character of a good and poor mentor (Gray and Smith, 2000). A summary of three of our personas for the mentors are:

- The 'gold-standard' mentor: one who facilitates learning appropriate to the student's level of ability and makes an appropriate assessment of the ability.
- The 'toxic' mentor: The term 'toxic mentor' was initially coined by Darling (1984), but discussions with mentors and students currently in practice indicates that this is still relevant. Darling described a gallery of toxic mentors as Avoiders, Dumpers and Blockers
- The 'hero' mentor: is the mentor who refuses help with a student despite potential difficulties such as personality clashes, professional value issues, etc. It appears that these mentors believe that they can turn anyone into a nurse and that if they don't it is their fault and not the students.

These were accompanied by scenarios describing the situations in detail and the problems mentors face while trying to carryout this function and still perform their duties on the ward. These persona and scenarios were developed from a combination of a literature review and interviews with current mentors and students advisors. Those listed above aided the first set of co-design workshops and helped the technical members of the design team get up to speed with the issues faced by the end users.

4.2.2. Storyboarding and Ontological Modelling

From the co-design session, personas, scenarios and the competency profile of the student user we were able to create story boards for the tools quite quickly. We first identified the key features from the scenario and sketched out the initial ideas. In parallel with the storyboarding we were developing our use case and activity diagrams. This helped shape the ideas, which were than transferred from sketches to PowerPoint. Our preliminary story board presents the basic ideas regarding the user interface, for example using tabs to allow users to recognize the available functionality.

We also explored the competency model that lay behind the portfolio (a simplified overview of this is shown in Figure 2). Through the modeling process we discovered that certain terms used were ambiguous or overlapping, this seems to have occurred since the model is the result of combining several other models, from professional bodies, the UK government and the University's own scheme.

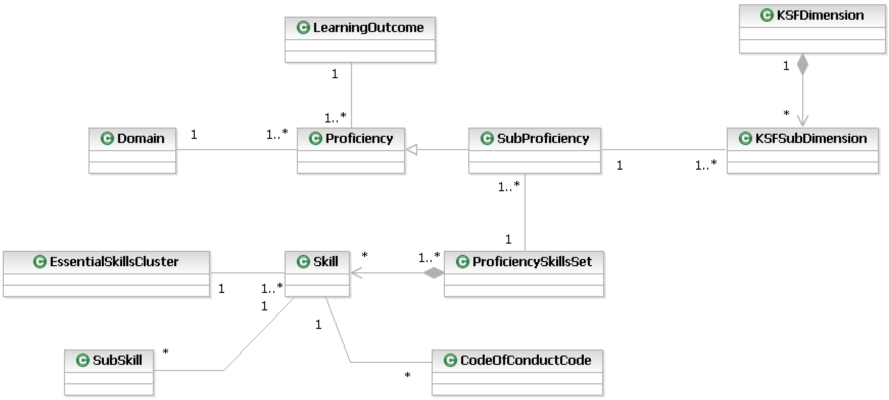


Figure 2: Initial competency model identified through ontological modeling

4.2.3. Use cases and Activity Diagrams

The development of the use case and activity diagrams was again a co-design exercise but on a small scale; just one or two members of the Nursing team joined in the activity. Figure 3 shows the use case diagram for the system. This is supported by normal use and alternative use scenarios for use cases. Figure 4 shows an activity diagram associated with a student using the competency model.

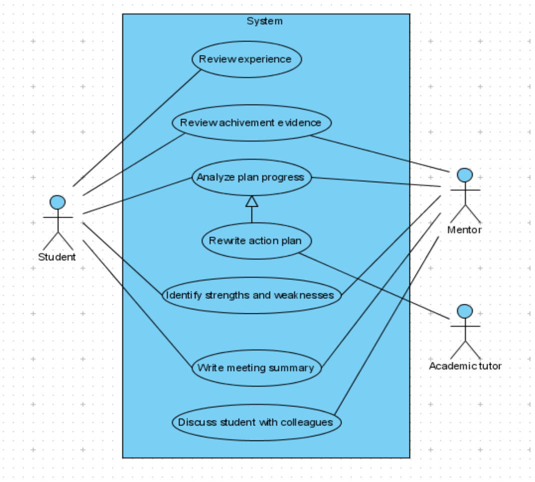


Figure 3: Use case for the system

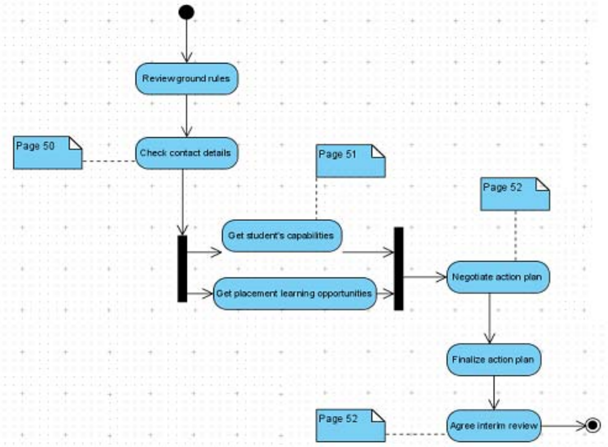


Figure 4: Activity diagram

When supporting students to assess their competency for a task it is necessary to decide how much 'scaffolding' support is required. Should we build in a very structured approach, directing them in the way they should go or a looser approach and let them find out themselves? We decided to adopt something between the two extremes, with a leaning toward the more structured approach. The reasons for this are:

- Part of the learning process involves students becoming reflective practitioners - being too prescriptive will not give students the opportunities they need to self-assess properly.

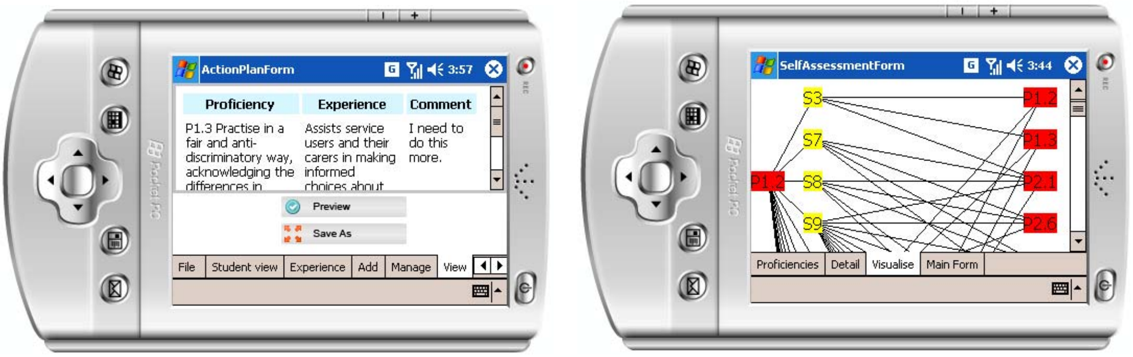
- There is value to a degree of structure, provided students are allowed to reflect on their learning. Giving them some structure will aid them to start the reflection.

As this is a new tool, and the entire system of practice assessment has changed, it was felt that a structured approach would make it easier for users to understand and use. A less structured approach is the goal, but is not realistic for an initial implementation. Rather than just provide a tool that mimics the multiple views of the current paper-based competency system, we wanted to provide functionality that supported the preparation for the initial meeting wherein the Action Plan is co-created by the student and mentor

4.2.4. Deployment

We are currently undergoing our first user trial (the main application is shown in **Figure 5**); this is part of our ongoing iterative development and deployment. We have undertaken a number of iterations so far, and they have changed the shape of the tools. An early change (driven by a design team review) was that the first Learning Contract Tool did not give a good overview of the competency model, and the domain experts felt that this was important; as a result the tool was revised to include a graphical view of the competency network and the student's progress within it. A major revision was needed when we first deployed the tool with the School of Nursing and Midwifery at Southampton, we discovered two things:

1. Around half the students that we gave the device to were unsure of whether they would use the device, they felt that it had potential but were worried over its complexity.
2. The mentors in practice were not happy with using the mobile device at all; they did believe that they needed mobile access to student's progress, and wanted a web based tool instead.



Proficiency	Experience	Comment
P1.3 Practise in a fair and anti-discriminatory way, acknowledging the differences in	Assists service users and their carers in making informed choices about	I need to do this more.

Figure 5: Mobile application showing an Action Plan (left) and Competency Network (right)

As a result of this deployment we revised our plans and our tool. We developed a simple help system that explained to students how the other applications on their mobile device (calendar, audio player, internet access, etc.) might be useful to their studies, and deployed the device *without our applications* for one month before the start of the trial, so that they could familiarize themselves with it. In deference to the mentor's opinions we created a web-based version of our Learning Contract Tool so that they could monitor the progress of their students from their desktop machines. We also worked with a separate group of domain experts to create the Profile Placements ourselves, so that the mentors did not have to.

The next stage of our deployment is to extend the mentor's web-based review tool to allow them to give feedback to the student's mobile tool, and to undergo a second deployment at the School of Nursing and Midwifery at Thames Valley University.

5. CONCLUSIONS

In this paper we have proposed a co-design methodology that enables the development of m-learning tools that operate in complex conceptual spaces. We bring together techniques found in HCI (personas, scenarios, and storyboarding), agile software techniques (iterative development and incremental delivery) and lightweight software engineering practice (use cases, simple ontological modelling and activity diagrams). The resulting methodology represents a modelling and co-design process that is reliant on *mutual learning* and a *fusion of horizons* towards a goal. The methodology is an agile approach that gives us a lightweight method of capturing and recording requirements and feeding these though the design cycle in such a way that they are integral (as opposed to tangential) to the software development process.

We have demonstrated our approach in action through a case study of developing a mobile toolkit for nursing students on placement. The conceptual space in this case is the interconnected competency framework they are required to measure their experiences against; this is particularly challenging as it brings together a number of complex skills frameworks and includes an epistemological dimension. In the future we hope to refine our co-design process, and look at how it might be extended through the deployment and evaluation phases of a mobile systems project.

We have shown how using the co-design methodology can drive the creation of more nuanced tools that help connect physical activities and conceptual spaces and models. Our experiences demonstrate how agile processes throughout design and development ease the co-design experience and allow a flexible and adaptable development process.

6. ACKNOWLEDGEMENTS

The authors would like to thank the extended MPLAT team, in particular our colleagues Hugh Davis and Mary Gobbi at Southampton, and Balbir Barn at Thames Valley. The MPLAT project is funded by JISC.

REFERENCES

Abowd GD, Atkeson CG, Hong J, Long S, Kooper R, Pinkerton M (1997). Cyberguide: a mobile context-aware tour guide. Wireless Networks, 3(5), 421–433.

Cooper, A. and R. Reimann (2003). About Face 2.0: The Essentials of Interaction Design, John Wiley & Sons.

Collins T.G., (2005) "English Class on the Air: Mobile Language Learning with Cell Phones," Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), pp. 402-403.

de Crom N., "The "ME"-Learning Experience: PDA Technology and eLearning in Ecotourism at TUT", , Fourth World Conference on Mobile Learning (mLearn), Oct 2005

Darling, L. (1984) What do Nurses Want in a Mentor? Journal of Nursing Administration 14 (10), October 1994: 42 -44

Duffy, A (2004) Failing students - a qualitative study of factors that influence the decisions regarding assessment of students', article for Nursing and Midwifery Council, http://www.nmc-uk.org/aArticle.aspx?ArticleID=1675 (last accessed Dec, 2007)

Flynn, M. & Pendlebury, D. & Jones, C. & Eldridge, M. & Lamming, M. (2000). The Satchel System Architecture: Mobile Access to Documents and Services. ACM Journal of Mobile Networks and Applications, 5 (4), 243-258.

Gray, M. Smith, L. (2000) The qualities of an effective mentor from the student nurse’s perspective: findings from a longitudinal qualitative study Journal of Advanced Nursing (32) 6:1542-1549

Kravcik M., Kaibel, A., Specht, M., and Terrenghi, L. (2004). Mobile Collector for Field Trips. Educational Technology & Society, 7 (2), 25-33.

Larman, C. (2004). Agile and Iterative Development: A manager’s guide. Pearson Education.

McMillan J. (2005), "Seven Reasons Why mLearning Doesn't Work", Fourth World Conference on Mobile Learning

Sharples M., Jeffery N., du Boulay J. B. H., Teather D., Teather B. and du Boulay G. H. (2002) Socio-cognitive engineering: A methodology for the design of human-centred technology , European Journal of Operational Research Volume 136, Issue 2, 16 January 2002, Pages 310-323, http://dx.doi.org/10.1016/S0377-2217(01)00118-7

Parsons D, Ryu H, Cranshaw M, (2006), "A Study of Design Requirements for Mobile Learning Environments," *icalt*, pp. 96-100, Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)

Sloan HL, Delahoussaye CP (2003). Clinical application of the Omaha system with the Nightingale Tracker: a community health nursing student home visit program. Nurse Educ. 2003 Jan-Feb; 28(1):15-7.

Weal, M. J., Cruickshank, D. G., Michaelides, D. T., Millard, D. E., De Roure, D. C., Halloran, J., Hornecker, E. and Fitzpatrick, G. (2006) A persistent infrastructure for augmented field trips. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, Ed-Media 06 (in press), Orlando, Florida, USA.

Wills G, Woukeu A, Carr L, and Hall H. (2003) The need for semantic in web design. Semantic Web workshop, Fourteenth Conference on Hypertext and Hypermedia Nottingham, UK