Didac Gil de la Iglesia

Uncertainties in Mobile Learning applications:
Software Architecture Challenges

Licentiate Thesis
Computer Science
2012
A thesis for the Degree of Licentiate of Philosophy in Computer Science.

**Uncertainties in Mobile Learning applications:**
Software Architecture Challenges

Didac Gil de la Iglesia

Linnaeus University
School of Computer Science, Physics and Mathematics
SE-351 95 Växjö, Sweden, 2012
http://www.lnu.se/
To Rallas and Hjärter
Abstract

The presence of computer technologies in our daily life is growing by leaps and bounds. One of the recent trends is the use of mobile technologies and cloud services for supporting everyday tasks and the sharing of information between users. The field of education is not absent from these developments and many organizations are adopting Information and Communication Technologies (ICT) in various ways for supporting teaching and learning. The field of Mobile Learning (M-Learning) offers new opportunities for carrying out collaborative educational activities in a variety of settings and situations. The use of mobile technologies for enhancing collaboration provides new opportunities but at the same time new challenges emerge.

One of those challenges is discussed in this thesis and it concerns with uncertainties related to the dynamic aspects that characterized outdoor M-Learning activities. The existence of these uncertainties force software developers to make assumptions in their developments. However, these uncertainties are the cause of risks that may affect the required outcomes for M-Learning activities. Mitigations mechanisms can be developed and included to reduce the risks’ impact during the different phases of development. However, uncertainties which are present at runtime require adaptation mechanisms to mitigate the resulting risks.

This thesis analyzes the current state of the art in self-adaptation in Technology-Enhanced Learning (TEL) and M-Learning. The results of an extensive literature survey in the field and the outcomes of the Geometry Mobile (GEM) research project are reported. A list of uncertainties in collaborative M-Learning activities and the associated risks that threaten the critical QoS outcomes for collaboration are identified and discussed. A detailed elaboration addressing mitigation mechanisms to cope with these problems is elaborated and presented. The results of these efforts provide valuable insights and the basis towards the design of a multi-agent self-adaptive architecture for multiple concerns that is illustrated with a prototype implementation. The proposed conceptual architecture is an initial cornerstone towards the creation of a decentralized distributed self-adaptive system for multiple concerns to guarantee collaboration in M-Learning.

Keywords: TEL, Mobile Learning, Distributed Systems, Uncertainties, Quality of Service, Self-Adaptation.
Acknowledgments

As I am approaching the final lines of this thesis, I would like to go through the reasons that made this work possible. When I was a kid, my parents transmitted me the idea that we are ephemeral in this world, so there are three MUSTs in every person’s life: to plant a tree, to have a child and to write a book. To plant a tree, you need a seed. To have a child, you need a partner. To write this book, I needed the help of several important people. In these lines I would like to acknowledge all the efforts you have put so this thesis comes out.

I want to thank my supervisors, Marcelo Milrad and Jesper Andersson, for the numerous and long hours they have spent giving me guidance, comments, suggestions and new points of view. I want to thank them as well for the efforts to improve the thesis readability and to reduce the ambiguities it used to have.

Thanks to my colleagues in CeLeKT and DFM, for all the moments we have spent together during this period, and the moments to come. Because, more than colleagues, we are friends here. Thanks to Miranda, Tonia, Yuliya, Bato, David, Ilir, Lars, Oleg, Osama and Oskar. Special thanks to Sadaf, for all the fikas and long conversations that kept me awake and helped me, more than caffeine, to keep working during the nights. And thanks to Maria, for frequently reminding me that ”la vida es un carnaval”.

Thanks to my friends Gloria, Puchu, Juanma, Janosch, Juan Pablo, Xavi and all the troupe in Spain. Thanks for motivating me from abroad; not an easy task. Thank you all for keeping in touch and wishing me success. You all know that I have you on my mind more than I have been able to show.

Thanks to the teachers and students involved in the research trials, besides the data, I collected good experiences with you.

My gratitude to my family who keep encouraging me to bring something to this world of ours, even if this means to be away from them. Thanks, papa i mama for raising me to be critical and logical, so much needed nowadays. And thanks, Alba, for always sending me energies when I am running out of them. I take your life as an example to follow. I love you!

Finally, the biggest thanks to you, Yuliya. Yes, you again, because you are not only a friend. Thanks for joining me during this last part of the thesis, listening to my worries and concerns and giving me good suggestions at all times. But thank you even more for supporting me even when I was not taking your advises. I hope I can pay it back. Thank You!

Växjö, Sweden
February 29, 2012
## Contents

1 Introduction ................................. 1  
1.1 Motivation ................................. 2  
1.2 Problem Definition .......................... 4  
1.3 Limitations .................................. 4  
1.4 Thesis Overview .............................. 5  
2 Research Approach ......................... 7  
2.1 Research Context ............................ 7  
2.2 Research Goal ............................... 8  
2.3 Research Methods ............................ 8  
2.4 Results ..................................... 10  
2.5 Validity of Results ........................... 11  
3 Theoretical Foundations ..................... 13  
3.1 Pervasive, Mobile, Ubiquitous Computing and Learning  ... 13  
3.2 Achieving Goals ............................. 16  
3.3 Self-Adaptation .............................. 18  
3.4 Resilience .................................. 21  
3.5 Self-Adaptation in TEL/M-Learning .............. 22  
3.6 Framework for Understanding Uncertainties .......... 27  
4 The GEM Iterations ......................... 29  
4.1 GEM1: Växjö ................................. 29  
4.2 GEM2: Stockholm ............................ 31  
4.3 GEM3: Vederslöv ............................. 33  
4.4 Summary .................................... 35  
5 Discussion .................................... 37  
5.1 Goals revisited ............................... 37  
5.2 Required QoS outcomes for collaboration ............ 38  
5.3 Uncertainties ................................. 38  
5.4 Risks ....................................... 42  
5.5 Mitigations and Outcomes ..................... 46  
5.6 Approach for Mitigation ..................... 49  
6 Conclusions and future work ............... 61  
6.1 Conclusions .................................. 61  
6.2 Future Work .................................. 63  
Bibliography .................................. 67  
A Glossary of Terms .............................. 77  
B Publications .................................. 79
List of Figures

1.1 Relation between Resource Share, Dynamism and Concerns . . . 3
1.2 Disposition of the thesis . . . . . . . . . . . . . . . . . . . . . 6
2.1 Context in the evolution of the research claim . . . . . . . . . 12
3.1 Dimensions of Ubiquitous Computing (Lyytinen and Yoo, 2002) 14
3.2 Design alternatives evaluation based on goals and environment . 17
3.3 MAPE-K. Model for self-adaptive systems (IBM, 2006) . . . . 19
3.4 Framework for handling of uncertainties and their effects (Hastings and McManus, 2006) . . . . . . . . . . . . . . . 28
5.1 Potential risks for collaboration due to knowledge-uncertainties-assumptions basis . . . . . . . . . . . . . . . . . . . . . . . 45
5.2 Modularity and Redundancy mitigations to address failure and degradation risks . . . . . . . . . . . . . . . . . . . . . . . 47
5.3 Modularity and Serviceability mitigations to address failure, need shifts and cost increase risks . . . . . . . . . . . . . . . 48
5.4 Design Choices and Serviceability mitigations to address degradation and cost increase risks . . . . . . . . . . . . . . . . 48
5.5 Generality mitigation to address market shift risks . . . . . . . 49
5.6 Upgradeability mitigation to address cost increase risk . . . . . 49
5.7 Simplified MAS in a mobile device scenario . . . . . . . . . . . 51
5.8 Components for a MAPE-K control loop design . . . . . . . . . 52
5.9 MVDs covering multiple concerns . . . . . . . . . . . . . . . . 53
5.10 Components view for the self-adaptive MAS prototype . . . . . 54
5.11 Component mapping to FORMS representation model . . . . . 56
List of Tables

3.1  Modeling dimensions in Self-Adaptive systems (Andersson et al., 2009) .................................................... 20
3.2  Overview of the use of concern, adaptation and resilience concepts in TEL/M-Learning and Mobile Computing projects ........ 26
4.1  Requirements identified in the GEM project ......................... 36
5.1  Collaboration properties ............................................. 37
5.2  Uncertainties and risks relationship in the GEM trials ............ 46
5.3  Mitigations for potential risks in GEM ........................... 50
Chapter 1

Introduction

The spread in the use of computers is reaching levels at which they can be found in situations where they would not have been considered before. Their use started with a few computers, which were so costly that they were mostly reserved for research or military purposes. Education has always encompassed the use of technology with the evolution of this adopting its improvements, as expressed in the literature (Herrington et al., 2009; Suppes, 1966). The application domain in which the thesis is situated is within the field of Technology Enhanced Learning (TEL).

A critical aspect in education is collaboration, one of the cornerstones of the learning process (Chiu, 2000; Dillenbourg, 1999). Chiu (2000) states that groupware activities provide several benefits in comparison to outcomes obtained from traditional individual activities. Thus, collaboration is a critical feature and supporting it should be considered in the design and implementation of TEL applications.

With the evolution and spread of computing devices, new paradigms have come into play in TEL, for instance pervasive and mobile computing technologies. Pervasive computing units are characterized by becoming invisible to the user, so the user is not aware of their existence, which emphasizes the focus on the activity itself (Weiser, 1991). Mobile computing technologies allow the user mobility while carrying a computational device. However, according to Lyytinen and Yoo’s definition (2002), they exclude the capability to “seamlessly and flexibly obtain information about the context”. The use of mobile technologies in education is known as M-Learning (Sharples et al., 2011).

The use of these new technologies offers new possibilities to education, but, as a drawback, it implies the addition of some new problems. The dynamic environment where M-Learning activities take place, in many cases in outdoor settings, entails a level of uncertainty that threatens collaboration. This uncertainty may be explained by the lack of knowledge about the events and environment during development and at runtime (Hastings and McManus, 2006).

In this thesis the problems that can affect the level of collaboration when mobile technologies are used in TEL activities in dynamic settings are investigated. Efforts described in this thesis include the characterization of different scenarios that involve mobile technologies in collaborative outdoor
TEL activities. This characterization brings up uncertainties present in M-Learning activities that may put collaboration at risk.

After having set the boundaries that describe these scenarios through the identified uncertainties and risks that may affect collaboration, this study searches for mitigation approaches to reduce the identified risks in order to achieve the desired software system outcomes. The identification of the uncertainties, risks and mitigations for a software system leads to the definition of a research claim and an agenda for future research.

### 1.1 Motivation

The use of ICT (Information and Communication Technologies) in education is not new but advancing at a pace that is driven by advances in other areas (Sharples and Roshchelle, 2010). Recently, the use of ICT has expanded to encompass a more complex work environment than the traditional setting. Outside the classrooms, and located in more authentic environments, mobile technologies allow the student to explore alternative ways to understand the subject matter. Consequently, the use of mobile and ubiquitous computing technologies in the field of education, known as M-Learning and Ubiquitous learning (U-Learning) respectively (Caballe et al., 2010; Jones and Jo, 2004), is progressing significantly. Ubiquitous computing technologies are those that, being mobile, can gather information from and model the environment to configure their services appropriately and invisibly to the user (Lyytimen and Yoo, 2002).

Commonly there are various stakeholders involved in the creation of the educational activities. The first group of stakeholders includes the students and teachers whose well-marked target is completing a specific teaching and learning activity, individually or collaboratively (Ames, 1992; Spikol et al., 2009). An activity includes a set of functional objectives to be accomplished that may vary depending on the learning activity.

However, there are several uncertainties that cause risks that may affect the achievement of these learning activities. An initial experience is the GEM1 (GEometry Mobile), the first of a set of activities further elaborated in Chapter 4. In this activity, the users worked in teams measuring distances in order to perform geometrical calculations in relation to lengths, areas and volumes using mobile devices and customized mobile applications. During the activity, some factors were detected that could have, and in some cases did, put the activity in a state that threatened the successful completion of the activity. One of these factors is the mobile device’s dependence on the communication infrastructure. Another factor is the risk of a student closing the M-Learning application while the activity is being carried out. These events affect not only the student carrying the mobile device but also the collaboration in the group. The physical environment brings uncertainties that can affect the activity as well. For example, meteorological conditions can affect the accuracy of GPS devices, which in turn
provokes a degradation of the functionalities needed for the collaborative activities. The existence of uncertainties and risks is confirmed by many similar projects, for instance, the Manhattan Story Mashup (Tuulos et al., 2007). The experience consisted of a collaborative story telling activity where the participants shared pictures taken from their mobile devices to be presented on public displays. The authors experienced a problem on the server that led to 30 minutes interruption in which all players were affected.

From the pedagogical point of view, in addition to the functional objectives defined by teachers and students, there are non-functional requirements for collaboration that claim attention. It is of interest to provide a certain level of Quality of Service (QoS) in the M-Learning activities aimed at collaboration. The activity flow in formal indoor activities can be assured to some extent by adapting the formulation of the activity, changing some parameters on the blackboard, applying different formulas, and so on. However, this can become an arduous task in collaborative outdoor activities with the use of mobile technologies, as it would require new system reconfiguration and possibly redesign of the application and redeployment on devices. As it has been shown in the above examples, incidents during outdoor collaborative M-learning activities can put at risk the execution of the lecture and, consequently, the expected pedagogical outcomes.

Therefore the motivation behind the efforts that guide this thesis is the creation of a software architecture to support and guarantee collaboration in M-Learning activities where the conditions are dynamic and uncertain and hence collaboration is difficult to guarantee. Figure 1.1, below, presents the intersection of the three components involved in this thesis.

![Figure 1.1: Relation between Technology, Environment Dynamism and Collaboration Concerns in M-Learning](image)

The collaboration concern involves a set of additional requirements. In their work, Neyem et al. (2009) initiate a list of requirements needed in order to support collaborative M-Learning activities. These include flexibility of the system, support to guarantee shared information, preparation of the system for mobile awareness, safety integration to provide privacy
and protection, communication mechanisms required for the collaboration tasks, and a networking infrastructure to offer the previous communication mechanisms. A derived motivation for this thesis is to identify the set of requirements that are necessary in collaborative M-Learning activities to guarantee that the required collaboration can be offered, as a step to determine the related risks and how these can be minimized.

### 1.2 Problem Definition

As mentioned previously, the focus of this research is on the use of mobile technologies in education to guarantee that technology supports collaborative learning activities. From a technological point of view, there are two separate components that need to be shared to support collaboration in mobile learning activities, namely the activity and the resources. To be able to follow a collaborative activity, it is necessary that the members involved in it have a common understanding of the tasks and actions to execute. This is to enable them to work coordinately and in cooperation. On the other hand, collaboration implies resource sharing. Resources can be hardware (such as the camera, GPS, memory space, processor unit, etc.), software (such as activities, media files, activity outcomes, messages, etc.) or communication resources (access to the Internet, access to internal servers, etc.).

In the field of M-Learning several efforts have applied technology to support collaboration. However, most of these solutions have looked at the field from an idealistic perspective (Herrington et al., 2009; Herskovic et al., 2009; Kurkovsky, 2010; Neyem et al., 2009; Rogers et al., 2010; Tarkoma, 2009; Tuulos et al., 2007), as suggested by Neyem et al. (2011). In most of these solutions view, they have considered the application to be used under optimal conditions, which is far from reality.

Summarizing, the environment in which M-Learning activities take place is dynamic, which leads to the existence of risks during the activity. These uncertainties can be the cause of risks that perturb collaboration or even result in collaboration not being achieved.

The challenge that this thesis addresses is to identify required features for a system that should provide assurances for collaboration in M-learning activities. Collaboration is assumed to be concerned with resource sharing and activities sharing among devices and participants. The assurances for collaboration must be valid for all dynamic M-learning environments.

### 1.3 Limitations

In the previous section we defined the problem we address in this thesis. This section describes the limitations and constraints of this thesis.

The thesis analyzes problems identified in the field of M-Learning related to collaboration. Collaboration in M-Learning embraces several aspects.
However, this thesis focuses on technological aspects related to the availability and reliability of shared resources (information, activities, etc.) in M-Learning activities. User interface evaluation and pedagogy aspects are not analyzed and are beyond the scope of this work. The research in this thesis does not tackle aspects related to security issues or privacy of the information shared between students involved in the study. Moreover, the thesis does not consider a work environment with a large amount of mobile devices, but considered a total of 30 devices, as this is the typical size of a class.

1.4 Thesis Overview

This thesis is based on a collection of five scientific peer-reviewed conference articles. These publications present the evolution of our initial concerns in relation to collaboration assurance in M-Learning activities where the environment is characterized by a level of uncertainty. As outcomes, a specific subset of functional and non-functional requirements for these scenarios is selected and the foundations to construct a system to assure this collaboration are conceived. Figure 1.2 illustrates the structure of the thesis. The research question is based on observations from previous M-Learning projects in CeLeKT (Center for Learning and Knowledge Technologies\(^1\)) and the GEM project.

Chapter 2 details the context of this research and identifies the research questions of the thesis. It also describes the methodological approaches applied. Chapter 3 presents the theoretical foundations of this research. It provides a description of concepts to be used in this thesis and presents the state of the art in the field of TEL and M-Learning in relation to collaboration assurance. A detailed description of the GEM project is given in Chapter 4 to reveal the identified requirements and to connect the outcomes with the publications attached to the thesis. The discussion that addresses the research question is provided in Chapter 5. An analysis of outcomes and challenges is presented as well. To conclude, Chapter 6 provides a summary of this thesis and a description of future research challenges.

\(^1\) http://celekt.info
Figure 1.2: Disposition of the thesis
Chapter 2
Research Approach

Chapter 1 presented the field of work for this thesis and the problem to be studied. This chapter presents the approaches applied during the research process from which this thesis derives. The chapter starts by providing the context to the reader to identify the research goal of this thesis. A description of the research methods used is provided. At the end of the chapter we provide a research map giving an overview of the research that was carried out including the research problems addressed, research efforts conducted and concrete research results.

2.1 Research Context

The efforts presented in the thesis are part of projects with a focus on M-Learning done in the CeLeKT research group. It is relevant to specify the context of this thesis to understand the origin of the researcher’s experience, how the initial hypothesis emerged, and how the research question was derived from it.

Projects in CeLeKT are multidisciplinary, combining new technologies in education. These projects aim to analyze the use of new tools and methods to support teachers and students. Within the studied tools, augmented-reality, inquiry-based learning, collaboration, and mobile technologies can be found. Within the last two, lies the project where this thesis focuses its efforts and which is introduced below.

AMULETS (Advanced Mobile and Ubiquitous Learning Environments for Teachers and Students) is a project in progress at CeLeKT. This project examines the use of mobile and desktop computers to create educational activities combining outdoor and indoor environments. This thesis is contextualized in GEM, one of the AMULETS sub-projects. GEM brings concepts from the field of geometry to students in a more authentic scenario. More concretely, the use of mobile devices in outdoor settings provides an alternative way to carry out mathematics learning activities within the geometry theme.

The GEM project involves 12–14-year-old students. In these activities students estimate and calculate figure measurements (lengths, areas, etc.) in outdoor settings employing mobile technologies. The figure dimensions become an incentive to use wireless technologies. The platform used for the
activities allows information to be shared between participants to strengthen collaboration. The main shared resource in the activities is the device location to perform distance calculations (Sollervall et al., 2011).

The use of mobile technologies to support collaboration in the learning activity is the major characteristic of the GEM project and is shared with many other projects. However, as it has been mentioned in Chapter 1, the constantly changing environment introduces a set of uncertainties that lead to certain risks in the activity.

2.2 Research Goal

The mobile learning activities executed in GEM require a system that supports collaboration. Our initial claims (C) were that

C from a technological viewpoint, collaboration can be guaranteed in outdoor collaborative M-Learning activities.

However, the experiences gained in GEM1 revealed several challenges with respect to the uncertain system environment, which may lead to putting collaboration at risk. These experiences resulted in the definition of the principal research question (RQ).

RQ What are the features of a system that can make it possible to guarantee collaboration in M-Learning activities?

A set of additional research sub-questions (Q) has been defined to complement the main research question.

Q1 Which architectural concerns are critical for collaboration assurance in M-Learning systems and which factors affect these concerns?

Q2 Which architectural patterns and mechanisms exist that enable collaboration guarantees in M-learning systems?

Q3 What is the state of research and state of practice in M-Learning for collaboration assurances?

2.3 Research Methods

The field in which this study is carried out has influenced the selection of the methodologies applied. Given the nature of this research, different approaches have been used at different stages of the study. That is, different techniques have been used depending on the types of outcomes to be achieved and the situation of the research question (Shaw, 2002). Desired outcomes were to understand the current state of the problem under analysis, explore the feasibility of the research claim and further evaluate concepts developed during the research. This section presents the methods used during the development of this thesis and the reasons that have led to their use.
2.3 Research Methods

2.3.1 Exploratory Case Study

This thesis describes the use of a number of customized applications to integrate different technological components addressing the concerns of the thesis. However, the first iteration of the platform was conducted on a previously existing experimental platform, adding the required functionality to provide autonomous positioning data exchange among devices in the activity. The researcher was present during the execution of the activity, which became an exploratory case study (Flyvbjerg, 2006), in order to collect the necessary information that would provide the basis for the study and identification of requirements (Winston Tellis, 1997) in relation to the collaboration concern. The intended outcomes of this method were to use a platform for mobile collaboration and identify the concerns that are linked to the "collaboration guarantee" aspect. This analysis is conducted in more detail in Chapter 4. In Chapter 5 the case studies are recapitulated to present the discussion of the thesis.

2.3.2 Survey

A survey method (Babbie, 1990; Wohlin, 2000) has been applied to provide a view of the concepts concerning this research in the field of Software Engineering, TEL, and M-Learning. This survey was carried out by collecting data from publications in six well-known journals and conference proceedings in the field of educational technologies and mobile communications technologies. To obtain a representative sample for the study, recent publications have been selected and a second filter has been applied based on the topics addressed in the publications.

2.3.3 Prototyping and Experiment

Experiments analyze the impact that causes the change of variables in a study by comparing different activities where the researcher has directly influenced the behavior of the system (Wohlin, 2000). Each one of the GEM iterations is an experiment that examines the impact of adding new technological functionalities into the M-Learning system, therefore carrying out the analysis through the comparison of the two versions (Shaw, 2003). For each iteration, a prototype has been developed considering the functionalities to be incorporated in order to study the prototype’s impact on the system. Prototypes allow the rapid introduction of new methods and features without too much interest in the refinement but allowing a better understanding of how to include and implement fuzzy requirements (Rogers et al., 2010).

Thus, in the GEM iterations it can be said that the variables under study have been the existence or absence of the features introduced on the platform in order to compare its merits with respect to previous versions.

It is worth mentioning that an evaluation has not been done to study different alternatives to provide functionalities to the system. The reasons
include the time spent in creating the platform and the frequency at which new requirements are added.

Requirements that have led to the final conclusions of this thesis are presented in the next section.

2.3.4 Theoretical Framework

The theoretical aspects related to the problem of this thesis have been explored using the framework presented by Hastings and McManus (2006). The framework is presented as a tool for researchers and developers to relate the outcomes desired in a project with the set of uncertainties that can be found during the design, implementation and use of these systems. The authors describe the following four concepts: uncertainties, risks, mitigations and outcomes, explaining their relationship with the following statement:

\[ \text{Uncertainty causes Risk handled by Mitigation resulting in Outcome} \] (Hastings and McManus, 2006).

It is important to differentiate between the expected outcomes in a software application and the outcomes of this research. The proposed framework provides a method to discuss the uncertainties in engineering systems and how they can be addressed. The intended outcome of this thesis aims the assurance of collaboration in M-Learning activities. As it has been mentioned before, uncertainties are present in M-Learning activities that may put collaboration at risk. The intended outcome of this thesis is to initiate the study of an architecture design that addresses the collaboration concern and mitigates the risks that the selected uncertainties may influence. It is therefore suitable to use Hastings and McManus’ framework (2006) as a guide for this study.

2.4 Results

The results obtained in this thesis are listed in this section. The results have been presented to relate them with the questions defined in Section 2.2. The argumentation behind each result is presented in more details in Chapter 5. Their presence in this section is to facilitate the development of the argumentation logic towards the thesis results.

R1 Report

Q1 Identify a set of concerns in relation with collaboration in dynamic collaborative environments and provide a view on the current efforts made in these settings

Q3 Description of current efforts in the field of M-Learning related to collaboration guarantees

R2 Descriptive conceptual architecture
2.5 Validity of Results

Once a model is created, even the proper research methods have been applied during its development, it is essential to check the model’s validity providing evidences of the claims (Shaw, 2002). IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990) defines the validation process as:

"The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements." (IEEE, 1990)

The conceptual model presented in this thesis must also be validated to evidence that it accomplishes the requirements identified in the research questions. Model validation can be done by example, where the model is put into practice to prove its claims. The validation process, in this case for a conceptual model, can also be done by implementation (proof of concept), evaluation, persuasion, analysis and experience (Glass et al., 2002; Shaw, 2002). In this thesis, initial steps have been taken by applying an evaluation validation using an implemented application.

The validation of the presented model is done by evaluating the results that could be obtained by applying the approach to our internal project. A model can be subjective when applied. Moreover, we do not want to fall into the error of creating a tailored model that benefits the activities presented in GEM, which is known as internal validity thread (Cook et al., 1979), but is not scalable to other external projects. Therefore, the resulting model presented in the thesis is also validated on a selected external outdoor collaborative mobile application (Tuulos et al., 2007) to describe the potential impact it would bring. The validation process is elaborated in Chapter 5. However, deeper analysis, development and validation in relation to the suggested conceptual architecture needs to be further elaborated in the coming years by testing the related implementations in more and larger scenarios (experience) and by studying the acquired data (analysis).
Figure 2.1: Context in the evolution of the research claim
Chapter 3

Theoretical Foundations

This chapter provides the theoretical foundations for this thesis. It begins with an overview of Pervasive, Mobile and Ubiquitous Computing and their relationship to Learning. After this overview, an introduction to achieving goals, self-adaptation and resilience is given to present the basic concepts behind them. The chapter reviews the state of the art in TEL and M-Learning with respect to the concerns of this thesis, narrowing down the efforts to the scope where this thesis is applied. The chapter ends with a presentation of the framework used during the discussion of this thesis to understand uncertainties, risks and mitigations.

The field of M-Learning offers the required characteristics to become an interesting field for exploring the different aspects related to the main research questions of this thesis. This field presents a relevant level of dynamism and uncertainty due to its mobile and distributed context.

3.1 Pervasive, Mobile, Ubiquitous Computing and Learning

The existing and increasing amount of technologies fulfill an important function in connection with nowadays’ emerging highly interactive computing environments. Weiser (1993) envisioned these technological environments in his pioneering work about pervasive and ubiquitous computing. He defined the notion of ubiquitous computing as "the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user" (Weiser, 1993). It can be inferred from his definition that his understanding of ubiquitous computing did not differentiate between on-place and mobile commuting technologies; it only differentiated between levels of visibility.

Lyytinen and Yoo (2002) made a major contribution to this field by introducing two dimensions of pervasive and ubiquitous computing environments: mobility and embeddedness of the service. Based on these two dimensions, the difference between pervasive and ubiquitous computing is on the level of mobility. The authors, based on the level of embeddedness dimension, propose another category: the mobile computing. In their work, mobile computing is described as: "In mobile computing, however, an im-
The important limitation is that the computing model does not considerably change while we move." (Lyytinen and Yoo, 2002). This description distinguishes the mobile computer field from ubiquitous computing by the invisibility of the used technologies in a moving environment.

The concept of mobile computing was already described by Imielinski and Korth (1996) and is aligned with the definitions provided by Lyytinen:

"Smaller units, often called personal digital assistants or personal communicators, will run on AA batteries and may have only a small memory; larger ones will be powerful laptop computers with large memories and powerful processors. Regardless of size, most mobile computers will be equipped with a wireless connection to the fixed part of the network, and, perhaps, to other mobile computers. The resulting computing environment, which is often referred to as mobile or nomadic computing, no longer requires users to maintain a fixed and universally known position in the network and enables almost unrestricted mobility." (Imielinski and Korth, 1996)

From this description, it can be noted that the characteristics of a mobile computing technology imply the use of not wired devices, both for communications and power supply. However, they do not imply that these technologies need to be transparent to the users.

According to Lyytinen and Yoo, "the main challenges in ubiquitous computing originate from integrating large-scale mobility with the pervasive computing functionalities" (Lyytinen and Yoo, 2002).

The complementary nature of pervasive and ubiquitous computing is clear from these definitions. According to Dargie et al. (2009), the goal of pervasive computing is to create smart environments that embed computation and communication in a manner that contextually interacts with the users.
in order to facilitate and support their daily tasks. One example of pervasive computing technologies is exemplified by the sensor grids (Lopes et al., 2009).

Embedding pervasive computing technology into physical environments, with the help of different sensors and actuators, brings computational power closer to user needs and actions. According to Ferscha et al. (2009), these trends will result on seamless integration of computers into our environment, thus enabling service delivery to be adapted to the users’ needs and their context.

Technological advances in sensing, computation, storage, visualization and mobile communications are turning modern mobile phones into a global mobile sensing device that can be used in a wide variety of settings and across contexts (Campbell et al., 2008). Contemporary mobile devices incorporate features that allow acquisition of environment information from sensors. Thus, these devices offer different possibilities to support adaptive mechanisms that meet the users’ needs while considering their environment. Integrating mobile computing technologies into everyday’s activities will facilitate their execution anywhere and at any time, by providing access to information and interacting with the users’ physical context.

With mobile computers, each device belongs to one individual and is used for his/her own purposes. These purposes are individual or shared along several users. This brings an opportunity in the field of education. Nowadays, there is a large amount of mobile applications to provide users with the tools to achieve their goals. To exemplify it, Apple Store offers, only in the field of education, more than 26,000 applications.1

The use of technologies for learning purposes, when the learner is mobile or when using mobile technologies, is known as M-learning (Sharples et al., 2011). Based on the use of mobile computing technologies, M-Learning does not consider the invisibility of applied technologies in learning activities. Yahya et al. (2010) provide a definition for U-Learning, which is based on the definitions for ubiquitous computing technologies. We agree with this definition. To the best of our knowledge, existing efforts in TEL have not fulfilled all criteria to comply with the definition of U-Learning. This is also emphasized by Yahya et al.

"we move from conventional learning to electronic-learning (e-learning) and from e-learning to mobile-learning (m-learning) and now we are shifting to u-learning.” (Yahya et al., 2010)

Further, Yahya et al. consider context-awareness as a corner-stone characteristic for U-Learning.

"The environment can adapt to the learners real situation to provide adequate information for the learners.” (Yahya et al., 2010)

Some M-Learning projects have considered learning content adaptation, but

---

1 Apple Store Education Apps by the end of 2011
have ignored the system adaptation to address the dynamism of the environment. This argument is further elaborated in Section 3.5.

3.2 Achieving Goals

Software requirements provide the specifications which a software system needs to implement. Therefore, it is important that software requirements elicit the objectives the stakeholders want to obtain with the product. In 1977, Ross and Shoman highlighted the importance of goals in the definition of requirements:

"Requirement definition is a careful assessment of the needs that a system is to fulfill. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed. Thus, requirement definition must deal with three subjects: context analysis, functional specification, designs constraints." (Ross and Shoman, 1977)

It is in the context analysis process where Ross and Schoman considered the goals to be analyzed in order to explain the why a requirement needs to be included in the project (Ross and Shoman, 1977).

Due to the importance of goals in the design of software solutions, as they refer to intended properties to be ensured by the system (van Lamsweerde, 2001), several studies have used goals in the process of requirement engineering (Dardenne et al., 1991; Robinson, 1989; Zave, 1997; Zendagui, 2009).

Goals describe both functional concerns and non-functional concerns which the involved stakeholders have in the project. Therefore, goals will commonly require the cooperation of several agents involved in the system: humans, devices and software (van Lamsweerde, 2001).

The IEEE Recommended Practice for Software Requirements Specifications (Engineering and Committee, 1998) and Systems and software engineering - Architecture description standard (Hilliard, 2010) are an attempt to alleviate the lack of requirements elicitation and goals definition issues in software engineering. The first describes the process of software requirements specification (SRS) for a software to be developed or even for the selection of commercial software products. The basic issues which a SRS should address are (1) functionality of the system, (2) external interfaces, (3) performance, (4) attributes and (5) designed constraints imposed on an implementation (Engineering and Committee, 1998). The first and fifth issues directly involve the description of goals of the stakeholders, describing properties to be ensured by the system and limitations to be considered. More recent is the Systems and software engineering - Architecture description standard (Hilliard, 2010), which proposes a methodology to describe
the system functionalities depending on the different views the stakeholders have. In turn, views are defined according to each one of the goals the stakeholders hold. These can be detailed system functionalities, such as user login process, and non-functional goals, such as QoS (availability, reliability, flexibility, performance, etc.), security and so forth. The standard is an attempt to do a separation of concerns, so the architecture can be described from each one of the viewpoints and provide the required, related and relevant information about the architecture for the addressed goal. Applications provided in the standard, to the date of this thesis (Hilliard, 2010), consider the architecture models, such as component models or deployment models, as static solutions which represent the concerns.

A language to describe the architecture, based on different concerns, views and viewpoints, becomes a useful tool to evaluate and compare possible design alternatives for the software solution and select the best candidate to cover these concerns. It is also a mechanism to analyze and document missing features in the candidates and facilitate the description of changes to be done on the design alternatives during the development cycle.

A set of assumptions are made for the design and development of a software solution. The assumptions are taken based on the knowledge and uncertainties present at each instant. During design and development, some uncertainties may get solved, amplifying the knowledge base and strengthening or redefining the previous taken assumptions. Therefore, it is a common practice to apply several iterations in the development cycle (Figure 3.2).

![Figure 3.2: Design alternatives evaluation based on goals and environment](image)

However, some mitigations cannot be achieved during development, but are postponed due to the existence of remaining uncertainties. Therefore, it is becoming recently more accentuated that systems need to be adaptive to changes in the environment. Systems are becoming more complex, and are making more components vulnerable to failure. Some examples are distributed systems, the adoption of external services in software solutions
and the incursion of services in the Cloud. Mobile and Ubiquitous Computing are fields in which environment changes are not rare and are potential causes for failures in the system, which in turn can lead to get one or several goals uncovered (Musolesi, 2009; Thomas, 2010; Wang and Liu, 2009). These goals include resource availability, reliability and performance.

Therefore a static approach is not always a sufficient solution. Instead, an adaptive solution should be considered. In changing environments, goal definition and requirement engineering processes to address these goals are more difficult to achieve. Uncertainty is a factor to be considered in evolving environments, but it was not reflected in the standards presented above. SeMaPS is an ontology-based language that attempts to address uncertainty in the context of the execution environment to infer the situation and provide an understanding for the system to react (Zhang et al., 2010). RELAX is another language that addresses uncertainty in system requirements, allowing them to be either mandatory to be covered or to apply relaxation on them, in case they become into conflict with other goals due to changes in the environment (Whittle et al., 2010).

The environment, however, is not the only dimension capable of changing during run-time. Goals are also candidates to be modified, removed or added during execution time. Ideally, all the concerns will be static and defined in development time, but changes in the environment or in the activity being carried out itself can force modifications in the defined goals. These aspects will be more elaborated in Chapter 5.

### 3.3 Self-Adaptation

Following the discussion from the previous section, new risks may surface during runtime. A system that attempts to mitigate these risks requires adapting to new circumstances. Formulas to provide system adaptations at runtime are known as *Self-adaptation* mechanisms. Huebscher and McCann (2008) make a connection between the concept of self-adaptation and biology. The human body has capabilities to react to events in our environment through our autonomic nervous sense, where the term autonomic comes from. Self-adaptation receives, as well, the names of autonomic computing and self-management. The authors describe the concept as follows:

"The system self-adapts to the environment to overcome challenges that can produce failure, but it is not measuring its own performance and adapting how it carries out its task to better fit some sort of performative goals." (Huebscher and McCann, 2008)

Huebscher and McCann agree with Horn (2001) in his description of self-adaptation as a collection of properties that can be enumerated as self-configuration, self-optimization, self-healing and self-protection. Self-configuration property allows a system to adapt the system’s configuration to address high-level goals. Self-optimization enhances the performance of
the system by adapting how its resources should be used. By self-healing, we understand that the system is capable of adapting its behavior to redirect its functionalities towards the defined goals in case of failures. Examples are mirroring systems and heartbeat solutions, just to name a few. Finally, self-protection offers the system with the capabilities to maintain its functionalities in case of attacks or misuse by the client, which could compromise the security of the system (Huebscher and McCann, 2008). These four properties, existing in self-adaptive systems, derive from the characteristics in software agents: autonomy, social ability, reactivity and pro-activeness (Wooldridge and Jennings, 1995).

A self-adaptive system was introduced by IBM in 2003 (IBM, 2003) and further developed in 2006 (IBM, 2006). MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) contains five components required by a system to become self-adaptive. A Monitoring component (1) acquires information from the system, and serializes it to be transferred to the Analysis component (2). Based on a base of knowledge (5), the Analyze component will infer from the data provided by the outside to be able to understand the contextual situation. Subsequently, the Plan component (3) will design a set of events that will be Executed (4) to apply the adaptation actions (Figure 3.3).

![Figure 3.3: MAPE-K. Model for self-adaptive systems (IBM, 2006)](image)

The knowledge a system possesses will affect the Plan of actions. Huebscher and McCann (2008) describe three different kind of knowledge representation used in self-adaptive systems: Concepts of Utility, Reinforcement Learning and Bayesian Techniques.

However, these three representations are based on the idea of a close world, where all possible values of the environment have been considered in the Knowledge base and, therefore, a plan can always be prepared. However, uncertainty, as explained in the previous section, is a common element in complex systems. Addressing the unforeseen is the next step in the evolution of self-adaptation, which would make a system resilient to changes not only under foreseen and foreseeable events, but also under the unforeseen events.

Andersson et al. (2009) provide a complete list of modeling dimensions for self-adaptive systems. The list includes Aspects of Goals, Changes occurred, Mechanisms for self-adaptation and Effects of Self-Adaptation (Table 3.1).
### Aspects in System Goals

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibility</td>
<td>Rigid. Constrained. Unconstrained</td>
</tr>
<tr>
<td>Duration</td>
<td>Temporary. Persistent</td>
</tr>
<tr>
<td>Dependence</td>
<td>Independent. Dependent—Conflicting. Dependent—Complementary</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>Single. Multiple</td>
</tr>
<tr>
<td>Evolution</td>
<td>Static. Dynamic</td>
</tr>
</tbody>
</table>

### Changes for adaptation

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>External. Internal</td>
</tr>
<tr>
<td>Frequency</td>
<td>Rare. Frequent</td>
</tr>
<tr>
<td>Type</td>
<td>Functional. Non-functional. Technological</td>
</tr>
<tr>
<td>Anticipation</td>
<td>Foreseen. Foreseeable. Unforeseen</td>
</tr>
</tbody>
</table>

### Mechanisms for adaptation

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Parametric. Structural</td>
</tr>
<tr>
<td>Organization</td>
<td>Centralized. Decentralized</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Autonomous. Assisted</td>
</tr>
<tr>
<td>Scope</td>
<td>Local. Global</td>
</tr>
<tr>
<td>Duration</td>
<td>Short Term. Medium Term. Long Term</td>
</tr>
<tr>
<td>Timeliness</td>
<td>Best effort. Guaranteed</td>
</tr>
<tr>
<td>Triggering</td>
<td>Event. Time</td>
</tr>
</tbody>
</table>

### Effects on the system

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criticality</td>
<td>Harmless. Mission-critical. Safety-critical</td>
</tr>
<tr>
<td>Overheard</td>
<td>Insignificant. Failure</td>
</tr>
<tr>
<td>Predictability</td>
<td>Non-deterministic. Deterministic</td>
</tr>
<tr>
<td>Resilience</td>
<td>Resilient. Vulnerable</td>
</tr>
</tbody>
</table>

**Table 3.1:** Modeling dimensions in Self-Adaptive systems (Andersson et al., 2009)
Goal aspects are variables that need to be targeted in self-adaptive systems, both in the knowledge database, to identify conflicts between interdependent goals, and for the plan component to design actions according to their flexibility (constraints to be considered) and evolution. Changes that provoke adaptation are a second dimension in self-adaptive systems. The type of adaptation presents changes that have been either functional, non-functional or technological. In other words, these changes are related to the system goals. It either needs to provide new functionalities, the QoS needs to be changed, or some components in the system require to be switched, just to enumerate some examples. The anticipation of a change is another relevant variable. Preparing a system to react to unforeseen changes would make it resilient to potential failures.

Andersson et al. (2009) present a classification of these dimensions applied in a traffic-jam monitoring system and in an Emergency Deployment System (EDS) to illustrate how this classification can be applied and to evaluate the completeness of the list.

3.4 Resilience

Resilience has been presented as a feature that a self-adaptive system can provide. The term resilience is mainly used in the field of physics. Oxford Advanced Learner’s Dictionary defines it as "the ability of a substance or object to spring back into shape; elasticity"\(^{1}\). This elasticity ability that allows an object to spring back into shape can be also desired in software engineering, making software products get healed and bring back their desired behaviors, as presented in the previous section. The second provided definitions is "the capacity to recover quickly from difficulties; toughness", which illustrates the desired outcomes in applying self-adaptation in software engineering solutions.

In SE, resilience has its origin in the dependability community (Avizienis et al., 2001). Resilience with complex goals has not been implemented in complex software systems. The few projects that have provided resilience have done it for a single system goal, such as connectivity (Wang and Liu, 2009). In the SE field, Laprie describes resilience as:

"the act or action of springing back. The ability to successfully accommodate unforeseen environmental perturbations or disturbances" (Laprie, 2008).

To understand resilience, it is worth mentioning the concept of dependability. Avizienis (2000) described dependability as the ability to deliver service that can justifiably be trusted. This trustworthiness requires the development cycle to fix faults identified during the design and development phases. However, faults can also appear during the execution phase. Causes of faults can be found in development, explained by deficiencies in

\(^{1}\)http://oxforddictionary.com/
the software code due to limitations in the design process, human errors or third-party software errors, such as bugs existing in libraries included in the project. Commonly, these issues could be identified during a testing phase, allowing the system to evolve by following one or more iterations in the development cycle process. However, faults, degradations and other system risks can be found in the physical environment and interactions, too (Avizienis, 2000). Examples of these include faults in hardware components, GPS reception problems and connection issues to remote systems such as services in the Cloud. A reliable system should provide fault-prevention, fault-tolerance, fault-removal and fault-forecasting to avoid failures in its services in case of facing development-, physical- or interaction-provoked faults. Dependable systems are those systems that address faults considering the world as a deterministic world. This means that the system is constructed to overcome faults that can be identified, monitored and classified, and a system reaction is designed for the system to spring back to a fair working behavior (Laprie, 2008).

However, reality is not deterministic. There are potential faults that cannot be considered beforehand. Therefore, a level of uncertainty is present and needs to be addressed by the system to overcome potential risks. In the light of the previous definitions, resilience can be defined as adding robustness to dependability:

\[
\text{Resilience} = \text{dependability} + \text{robustness}
\]

To sum up, a dependable system should provide self-adaptation mechanisms to make the solution resilient to risks caused by the unforeseen events (uncertainties that cannot be previously addressed during design and development phases). In this manner, the system retains its ability to deliver its services in conditions which are beyond its normal operation domain.

### 3.5 Self-Adaptation in TEL/M-Learning

This section provides an overview of the state of the art in the application of self-adaptation towards resilience in the TEL and M-Learning fields. Six major journal and conference proceedings have been selected, based on their impact in the field of TEL, Mobile Learning and Mobile Computing. The selection of articles was made in June 2011. The following list presents the first filtering criterion applied for the selection of the publications under analysis. It also provides the years and the conferences publications and journals under study.

- IEEE TLT (Transactions on Learning Technologies)
  2007-2011
- IEEE TMC (Transactions on Mobile Computing)
  2007-2011
The literature on Software Engineering identifies the concepts of concern, self-adaptation and resilience as key aspects to consider in development based on distributed architectures. Based on these three concepts, the following filters were applied on the full text articles to proceed to a second filter based on the abstract.

- "goals" OR "concerns"
- "self-adaptation" OR "self-healing" OR "self-management" OR "autonomic computing"
- "resilience" OR "dependability" OR "fault prevention" OR "fault tolerance" OR "fault removal" OR "fault forecasting"

A total of 88 articles fit with the filter criteria in TLT, 246 articles in TMC, 5 articles in WMUTE, 139 in PMC and 85 in PUC. The 199 articles published in EC-TEL were not filtered but included to be selected based on the abstract. Due to the high amount of articles that passed the filtration process, and the broad use of the word goal in research, it was decided to apply an additional filter based on the 762 articles’ abstracts. Forty articles were selected in this second process where at least one of the following considerations was met:

1. The concept of concerns was a relevant aspect in the article.
2. Self-adaptation was considered as a requirement in the described application or scenario.
3. Resilience was expressively addressed in the abstract or the system suggested its use.

Table 3.2 presents the selection of articles used for this study and their considerations of addressed goals, self-adaptation mechanism application and resilience considerations.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Goal</th>
<th>Self-Adaptation</th>
<th>Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROME (Hambach and Martens, 2008)</td>
<td>Static multiple goals (Model for codesign)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WHURLE 2.0 (Meccawy et al., 2008)</td>
<td>Static single goals.</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>Peer Assessment Authoring Tool (Miao et al., 2008)</td>
<td>Static single goals.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ShapeBuilder (Pearce et al., 2008)</td>
<td>Static multiple goals (codesign).</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ScenEdit (Pernin and Emin, 2008)</td>
<td>Static multiple goals (codesign)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Generic Service Integration (de la Fuente Valentin et al., 2008)</td>
<td>Static single goals.</td>
<td>Deployment adaptation</td>
<td>-</td>
</tr>
<tr>
<td>LAG (Cristea et al., 2009)</td>
<td>Static multiple goals (simple rule-based goals).</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>Dynamic Item Framework (Ulrich and Lu, 2009)</td>
<td>Static single goals.</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>APOSDLE (Lindstaedt et al., 2009)</td>
<td>Static single goals.</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>ISiS Model (Pernin and Emin, 2008)</td>
<td>Static multiple goals (codesign)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CAMera (Schmitz et al., 2009)</td>
<td>Static single goals.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Goal-management tool (Scholl et al., 2009)</td>
<td>Static single goals.</td>
<td>Resource adaptation (selection)</td>
<td>-</td>
</tr>
<tr>
<td>Adaptive VR for GRAPPLE (Troyer et al., 2009)</td>
<td>Static single goals.</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>User-Adaptive Recommendation Techniques</td>
<td>Static single goals.</td>
<td>Resource adaptation (selection)</td>
<td>-</td>
</tr>
<tr>
<td>(Ruiz-Inieta et al., 2009)</td>
<td>Activity goal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
### 3.5 Self-Adaptation in TEL/M-Learning

<table>
<thead>
<tr>
<th>Reference</th>
<th>Goal</th>
<th>Self-Adaptation</th>
<th>Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDiM (Zendagui, 2009)</td>
<td>Static single goals. Activity goal</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multi-agent systems (Aseere et al., 2010)</td>
<td>Dynamic multiple goals. (Just concepts)</td>
<td>Activity flow adaptation. Resource adaptation. Organization adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>Adaptive extension for IMS LD (König and Paramythis, 2010)</td>
<td>Static single goals. Activity goal</td>
<td>Activity flow adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>CAVIar (Melia, 2009)</td>
<td>Static multiple goals (dependence)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Annie (Thomas, 2010)</td>
<td>Static single goals. Activity goal</td>
<td>Activity flow adaptation. Optimal path. Uncertainty is used</td>
<td>-</td>
</tr>
<tr>
<td>EAP framework (Zemirline et al., 2011)</td>
<td>Dynamic multiple goals. (Language). No conflict evaluation</td>
<td>Multiple dimensions</td>
<td>-</td>
</tr>
<tr>
<td>JIL (Vargas et al., 2011)</td>
<td>Static single goals. Activity goal</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Formulas for mood recognition (Moridis, 2009)</td>
<td>Static single goals. Activity goal</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SOPN (Lin, 2008)</td>
<td>Static single goals. Activity navigation evaluation</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Traffic algorithm (Hoh et al., 2011)</td>
<td>Static multiple goals (specific solution)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Connectivity-Guarantee for MSN (Tan et al., 2009)</td>
<td>Static single goals. Connectivity goal</td>
<td>Path adaptation. Considered for future work</td>
<td>-</td>
</tr>
<tr>
<td>Uncertainty model for movement (Li and Wu, 2010)</td>
<td>Static single goals. Connectivity goal</td>
<td>Path adaptation.</td>
<td>-</td>
</tr>
<tr>
<td>Hedonic coalitions formation (Saad et al., 2009)</td>
<td>Static single goals. Connectivity goal</td>
<td>Organization adaptation. Behavior adaptation</td>
<td>-</td>
</tr>
<tr>
<td>Fuzzy Set framework (Anagnostopoulos and Hadjiefthymiades, 2008)</td>
<td>Dynamic multiple goals. (Language). No conflict evaluation</td>
<td>Multiple dimensions</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Reference</th>
<th>Goal</th>
<th>Self-Adaptation</th>
<th>Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chameleon (Liu and Shenoy, 2008)</td>
<td>Static single goals.</td>
<td>Energy goal</td>
<td>-</td>
</tr>
<tr>
<td>LET’S GO! (Maldonado and Pea, 2010)</td>
<td>Static single goals.</td>
<td>Activity goal</td>
<td>-</td>
</tr>
<tr>
<td>Optimal Accounting Policies (Zaghloul, 2010)</td>
<td>Static multiple goals (specific solution)</td>
<td>Hypermedia adaptation (QoS)</td>
<td>-</td>
</tr>
<tr>
<td>Robust routing (Wang and Liu, 2009)</td>
<td>Static single goals.</td>
<td>Path adaptation.</td>
<td>Communication</td>
</tr>
<tr>
<td>CAR (Musolesi, 2009)</td>
<td>Static single goals.</td>
<td>Path adaptation.</td>
<td>Communication</td>
</tr>
<tr>
<td>AMULETS (Pettersson and Gil de la Iglesia, 2010)</td>
<td>Dynamic multiple goals. (Just concepts)</td>
<td>Deployment adaptation</td>
<td>Communication</td>
</tr>
<tr>
<td>DIY (Do it yourself) (Kohtake et al., 2007)</td>
<td>Static single goals.</td>
<td>Behavior adaptation</td>
<td>-</td>
</tr>
<tr>
<td>Lily Arbor (Rogers et al., 2010)</td>
<td>Static single goals.</td>
<td>Activity goal</td>
<td>-</td>
</tr>
<tr>
<td>Hyperperform (Weller et al., 2011)</td>
<td>Static single goals</td>
<td>Physical adaptation</td>
<td>-</td>
</tr>
<tr>
<td>SeMaPS (Zhang et al., 2010)</td>
<td>Static single goals</td>
<td>Technological and Logical Reasoning language</td>
<td>-</td>
</tr>
<tr>
<td>PervML (Serral et al., 2010)</td>
<td>Static multiple goals (independent)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multimedia and QoS in WMN (Liu et al., 2009)</td>
<td>Static single goals</td>
<td>Hypermedia adaptation (QoS)</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.2: Overview of the use of concern, adaptation and resilience concepts in TEL/M-Learning and Mobile Computing projects
The study of the forty articles reveals that research in TEL and M-Learning mostly addresses projects where a single goal has been considered. Most often the single goals was completing the learning activity. Even multiple stakeholders (teachers, students, developers, researchers) were involved in the projects, the activity goal is the only one expressed and considered. Minor are the cases where several goals have been analyzed. In the cases where several goals have been analyzed, two or more goals are mostly considered during the design phase and a solution is obtained in consensus through a co-design process. Only some studies in M-Learning show cases where multiple goals get into conflict and a new specific solution is designed to address both of them. The cases explained above consider goals as static elements that can be analyzed during the design phase and implemented in the development phase. However, it has been recently evidenced that goals can not be static, but they must be modifiable, removable and add-able at runtime (Aseere et al., 2010). These include the criteria for group creation, the learning path to use in the field or even some technical aspects like the need for data mirroring. This challenge is not exclusive of TEL, but it is present in Mobile Computing and other fields as mentioned before.

A system that allows stakeholders to change goals during run-time should possess self-adaptation features. Self-adaptation is starting to be considered in the TEL and M-Learning fields. Due to the pedagogical perspective, the activity flow has been the most considered aspect to include self-adaptation mechanisms. The study of the students’ interactions (behavior, answers, actions), in individual aspects or in group collaborations, could trigger the adaptation of the activity flow in order to optimize the students’ outcomes. There are also recent studies, in Mobile Computing, that focus on self-adaptation oriented to technological aspects. These include the protocols and compression mechanisms to provide the proper QoS in hypermedia delivery, and paths to use to maintain network communications.

Resilience also played a role in technology-oriented studies. In these studies, self-adaptation mechanisms have been applied to provide resilience towards the connectivity among mobile devices by transmission route adaptation. However, as previously mentioned in Section 1.2, our predictions became true when we did not find studies in which resilience has been applied to assure functional requirements.

### 3.6 Framework for Understanding Uncertainties

This section provides an overview of Hastings and McManus’ framework (2006) to understand uncertainties in development and the mitigations that can be applied to them. The framework describes uncertainties and their relationships with system risks which affect the desired system outcomes. In addition, it discusses mitigations that may reduce or remove the consequences of such risks. The framework, however, does not show the re-
The relationship between uncertainties and mitigations. Therefore, it becomes a descriptive representation to "aid in the understanding of uncertainties and techniques for mitigation and even taking positive advantage of them" (Hastings and McManus, 2006).

The framework provides a classification with a brief definition of each one of the four concepts (see Figure 3.4) to help in the understanding of the uncertainties and their impact.

![Figure 3.4: Framework for handling of uncertainties and their effects (Hastings and McManus, 2006)](image)

This framework can be used to describe the uncertainty solving process both during design, development and execution phases, as different mitigation mechanisms can be placed to operate in each of them.

This chapter has provided the foundations of this thesis, necessary to understand the relevant aspects to be discussed in Chapter 5 with respect to self-adaptation in TEL and M-Learning. An overview on TEL, M-Learning and Mobile Computing has been provided to illustrate the current state of the art in these areas. In the following chapter, these efforts are further explored by analyzing three different GEM case studies in order to identify the requirements to support collaboration in M-Learning activities in dynamic settings.
Chapter 4

The GEM Iterations

The designs, concepts and conclusions to be discussed in Chapter 5 arise from the practical work done in this research, which has been an evolutionary process applied to a M-Learning system. As described in section 1.3, a set of outdoor activities have been executed in GEM.

The GEM project makes use of mobile computing technologies in a way that allows acquisition of sensory information from the field to be used in indoors settings. In addition, it studies the share of information in the field to provide a collaborative outdoors environment and to increase the learning experience in the field. GEM was initiated in 2009. The three iterations, presented below, represent the main milestones of the project during the two years that followed, and the experiments corresponding to each one of these iterations.

4.1 GEM1: Växjö

The first iteration of GEM began during spring 2009. The project involved sixth grader students (12–13-year-old) in an outdoor activity to apply geometry concepts. The GEM1 activity took place in autumn in Växjö, located in south east of Sweden, within the proximity of Linnaeus University.

4.1.1 Objectives and Tasks

An objective of GEM, from the researchers’ perspective, was to identify the benefits of and limitations in the use of different information and communication technologies to support outdoor activities in this mathematics field and to motivate students’ collaboration. Another desired outcome, this time from a pedagogical perspective, was to enhance the students’ understanding of different measures by, through discussion and collaboration, building a relationship between spatial skills and the measurements acquired. The outdoor activity should encourage the students to reason about the validity of the measurements, reject the results that are unreasonable and study on the results that could appear as valid.

Another activity objective was to enforce the involvement of the participants to avoid a situation in which some students would feel excluded from the activity.
During the activity, the students were required to use GPS-enabled mobile devices to gather location data, as a variable in the calculation process of distances, areas and volumes. To sum up, and within the scope of this thesis, the following is a list of the objectives of GEM1:

- Guarantee group collaboration
- Insure the students’ involvement in the activity
- Provide tools for distance measurements

### 4.1.2 Identified Requirements

The experiment goals definition is a prior step to determine the requirements that a software solution should provide. The students would work in 3-peer groups and each of them would play a different role. The main task in the activity required a distance calculation. Therefore, one of the requirements for the system was to provide two devices that would acquire their location to be able to calculate the distance between them. The involvement of two devices, to get locations, aimed also at motivating the participation of the students inside the teamworks, since one of the goals was to allow student collaboration and participation in the experiment. Below is the list of requirements:

- Group collaboration
- Location acquisition
- Share of GPS coordinates among mobile devices
  - Remote access to GPS coordinates over long distances

### 4.1.3 Publications

Initial results, obtained from GEM1, were presented in the first publication (Pettersson and Gil de la Iglesia, 2010) attached to this thesis. One result was the need of reusability in M-Learning applications. We argue that reusability can be applied both in the contents of the activity as well as in the TEL components. The GEM1 showed a need of resource sharing, an approach to provide reusability of resources within the activity. The publication presented three concepts that are central for reusability, namely Logic, Resource and Agent. The Resource concept considers the hardware (HW) and software (SW) elements to be reused for future iterations. The first publication already introduced the need for resilience in M-Learning applications, as an evolutionary feature to facilitate the reuse of components among the evolutionary development process.

Reuse can also be considered during the exercise, by sharing device resources among students. This aspect was described in the second publication (Gil de la Iglesia and Pettersson, 2010). The paper focuses on the share of resources with the aim of providing extended functionalities in M-Learning activities. The publication initiated the idea of providing a flexible
solution to cover the variety of needs in mobile environments by sharing resources. This concern narrows down the requirement of a mechanism for real-time information sharing and the potential need for information creation on-demand. The publication described an initial architecture used to allow student collaboration by resource sharing. Further details will be given in Chapter 5.

Publication 1


Publication 2


4.2 GEM2: Stockholm

The second GEM iteration took place in Stockholm during winter 2009. This experiment, internally called MULLE, was based on the results of the first GEM iteration, and was conducted in collaboration with human and computer interaction (HCI) researchers from the Data och Systemvetenskap (DSV) Department at Stockholm University. This time, the experiment was conducted on fifth grade students, and the original idea of working in a collaborative manner was kept to study several aspects in the field of geometry.

4.2.1 Objectives and Tasks

The objectives of GEM2 were partially altered from those defined in the first GEM iteration. The students in the experiment were one course younger than the subjects in GEM1. Concepts, such as area, had been taught, but the perimeter and volume calculations had not been covered in the lectures in which the trials took place. Therefore, the learning objectives of the GEM2 experiment were limited to the topics of rectangle area calculations.

From the experience gained from the GEM1, HCI researchers identified a lack of involvement by the students that did not carry the main mobile device. For this reason, one objective in the GEM2 experiment was to increase the participation of the students by intensifying their role in the tasks. The application interface used in GEM1 showed that there was a lack of space to provide enough and relevant information to the students. Therefore, a new approach was used to display information among the two devices in each group, duplicating the display area and allowing to increase the amount of information that could be presented to the students. The
objective was to study the implications and benefits of the use of multiple User Interfaces to increase the amount of information that the users can access at a time, and analyze the technological implications of sharing the resources in the group. Additional GEM2 objectives can be listed as follows:

- Potentiate the involvement of the students in the group
- Extend the amount of information to be offered to the student
  - Split the information among multiple displays

### 4.2.2 Identified Requirements

The GEM2 implementation made use of the concepts and tools created for the GEM1 trial. However, some modifications were introduced to this version. The display resolution on mobile devices was low to provide all the information desired to be shown during the activity. Therefore, it was decided to distribute the information and display it among the two devices used in each group. This means that displays became a resource to be shared between mobile devices in the group. Moreover, videos, as a software resource for the activity, required sharing as well. Below is the list of requirements:

- Share of display resources
  - Remote access to a display on a mobile device
- Share of video files
  - Remote access to media files

The resource sharing became more relevant to GEM2, where new software and hardware resources were needed for sharing. This increase evidenced the rising complexity in the application design where resources needed to be shared. The concept of Mobile Virtual Device (MVD) was one of the new introductions in GEM2. The idea behind the MVD was to design mobile applications considering the functionalities required for the application, in spite of being in one physical device or distributed among several physical devices. Two examples were the need for large displays on mobile devices or the capability to acquire two coordinates at the same time. The first would demand the use of a high-resolution mobile device, or the combination of multiple devices to extend the display area. However, in the second example, it was found that some features could not be found in one single mobile device. The MVD approach aimed at addressing the lack of mobile technology that could provide all the needed features for the M-Learning activity.

### 4.2.3 Publications

The third attached publication (Gil de la Iglesia et al., 2010b) presented the MVD concept and how it could be used in M-Learning activities, with
the goal of facilitating the design of collaborative M-Learning applications and the implementation of the required systems.

The fourth publication (Gil de la Iglesia et al., 2010a) elaborated on the MVD implementation. It described the required components which constituted the MVD: Service Provision and Consumption, Service Discovery and Publication, Orchestration and Activity Interpreter.


4.3 GEM3: Vederslöv

The GEM3 iteration kept the focus on geometry, but adapted the activities towards space orientation and geometrical shape constructions (Sollervall et al., 2011; Yau et al., 2011). The scenario used in this iteration involved fifth grade students in Vederslöv, south east of Sweden. Triangle-based structures were created to motivate collaboration and strategy construction, and reason on basic trigonometry aspects.

4.3.1 Objective and Tasks

In the last iteration, the geometry activity was envisioned to motivate the student discussion and reason about concepts in the class. The first pedagogical objective in GEM3 was to explore the spatial orientation of the students by the creation of triangles on the field based only on the measurements of their sides. Secondly, a different approach was used to strengthen the involvement of the students in the groups. Instead of assigning a role to each student in the group, the approach would be to switch the student role at different stages of the activity. In this way, it would potentiate the coordination between students and the collaboration required to achieve the final goal.

Two experiments were conducted in GEM3, one in December 2010 and the second in February 2011. From a technical perspective, an Activity Server was present in the system, offering tasks instructions and following up the students’ performance during the activity. During the first experiment (GEM3-a), and due to technical issues, the Activity Server became unavailable. Fortunately, it had no noticeable effects on the students, as
they already had downloaded the tasks instructions. However, the activity tracks were not correctly saved. This drew attention to resilience aspects, and gave evidence that components may fail in M-Learning activities.

Information about the activity performance was recorded in activity logs. These logs contained information about the subsystem, individual and group student information. This information was critical to evaluate the system and the students, from developer’s and teacher’s perspectives respectively. Therefore, log availability needed to be ensured in spite of dynamic conditions issues.

Initially, the activity was envisioned to involve three groups participating in the activity per turns. However, due to time limitations and affected by the weather conditions, the activity would have better run with the groups working in parallel. This demanded a system that allowed changes in the activity flow during execution time. Additional objectives that were included in GEM for this iteration can be enumerated as:

- Dynamic behavior depending on students’ role
- Access to the activity script
- Access to the activity logs
- Dynamic activity script

During a second GEM3 execution (GEM3-b), the activity server failure risk was taken into account. This particular situation is detailed in Chapter 5, and the scenario is described in Section 5.6.5.

### 4.3.2 Identified Requirements

A major change in the technological infrastructure used for these trials was made in GEM3. The mobile device platform switched from Symbian to Android devices. The Mobile Web Server (MWS) (Wikman, 2006) project was closed down by Nokia in January 2010, offering no further updates and closing the gateway service that provided remote access to mobile Web servers. The system used in GEM3 shifted towards a multi-agent architecture based on JADE (Bellifemine, 2001). Further details are provided in section 5.6.1

GEM3 added new identified requirements to the previous iterations. The incidents detected in the GEM3-a suggested a deeper analysis to provide service script and log availability, the first as a necessary resource for the collaborative M-Learning activity, and the second as one of the produced outcomes. Moreover, the weather conditions proved that a system might be needed to change the activity flow during execution time. Therefore, the solution should be tolerant to changes in the activity during running time. Finally, the devices were required to provide multiple behaviors, depending on the services requested to offer in the organization (MVD), to support the fact that students would change their roles during the activity. Below is the list of requirements:

- Multiple behavior device
• Activity Script availability
• Log availability
• System tolerant to changes in the environment
• System tolerant to changes in the activity

4.3.3 Publications
The fifth publication (Gil de la Iglesia et al., 2012), attached in this thesis, recapitulates the requirements identified in the three GEM iterations. Once the requirements are listed, an analysis of architecture patterns is presented, comparing centralized systems, centralized distributed systems, decentralized distributed systems and decentralized distributed systems with self-adaptation mechanisms. The study describes the related drawbacks of each one of the first three approaches, and hypothesizes the use of the last one. This discussion is further elaborated in Chapter 5.

Publication V

4.4 Summary
This chapter has presented the three main trials run in GEM. The elaboration on these experiments allowed us to specify a set of requirements. A subset of the requirements listed in this section is common in the TEL field. However, we have identified requirements that are crucial for the M-Learning activities.

Due to the collaborative aims, the lack of functionalities in the mobile devices used in M-Learning activities, and the dynamism of the environment in which these activities take place, there are aspects that require a high level of attention in the field. These can be classified into the following three categories: Service Sharing, Organization Management, Resilience to availability issues. The following table summarizes the requirements (R) presented in this section, while classifying them into four categories.

The table above presents 10 requirements identified during the different GEM trials. These requirements are highly connected with the required outcomes to guarantee collaboration in M-Learning activities. Availability, Reliability and Performance in collaboration are concerns that are included in the 10 presented requirements. For example, remote access to GPS coordinates implies the availability of the GPS resource as well as the reliability of the information the GPS service provides. Moreover, to maintain the
flow in collaboration, performance in supplying the GPS coordinates must be guaranteed. This explanation also applies to other requirements, such as R4, R5, R7, R8. More complex is R10, but it still demands the availability, reliability and performance outcomes. Changes in the activity can affect collaboration in the activity, and the list of resources to be shared. An adaptation in the list of shared resources needs to be provided so that the new changes would be available and reliable to the students, and the collaboration performance would not be substantially affected.

Table 4.1: Requirements identified in the GEM project
Chapter 5

Discussion

The goal of this chapter is to provide a solid argumentation in order to demonstrate that we may mitigate the uncertainties identified in the GEM M-Learning field trials. We begin by recapitulating the collaboration assurances we set as our goal in Chapter 1 and discuss assurances for collaboration. Having those aspects as our starting point, we provide an in-depth discussion of uncertainties identified in the field trials and proceed by connecting them to concrete system risks. Each risk is further analyzed and a number of possible mitigation strategies are discussed. The chapter concludes by exemplifying the strategies with cases from the GEM trials and additional case studies.

5.1 Goals revisited

Previous research efforts confirm the benefits of collaboration in education (Chiu, 2000; Dillenbourg, 1999). Based on our experience from multiple case studies and an extensive literature survey conducted in the field, we have identified a set of properties that are necessary to guarantee collaboration in M-Learning activities, which we describe below in table 5.1.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperation</td>
<td>2 or more people need to work together to achieve their personal goals (Kling, 1991).</td>
</tr>
<tr>
<td>Coordination</td>
<td>The actors need to coordinate their actions, &quot;bring up activities into proper relation&quot; (Neyem et al., 2011), to direct their efforts towards their goal. It implies orchestration.</td>
</tr>
<tr>
<td>Resource sharing</td>
<td>Information, knowledge, activity, device components need to be shared.</td>
</tr>
</tbody>
</table>

Table 5.1: Collaboration properties

The properties emphasize the need for resource sharing. Moreover, the communication channel must guarantee support for collaboration. Such guarantees are described in terms of Quality of Service levels that a system is expected to provide. In the following subsection we will discuss the quality
of service concerns to be considered and, to some degree, the desired service levels to be achieved.

5.2 Required QoS outcomes for collaboration

The collaboration properties underline the importance of providing quality of service with respect to resource sharing, including resource availability and their accessibility to other participants in the collaborative activity. To provide an acceptable quality of service in collaboration we consider three specific quality concerns: Availability, Reliability and Performance. Obviously, there are other important quality concerns that contribute to the overall quality of service for collaboration, but they have not been the focus of this study. We describe the three aspects below. The description captures the informally specified QoS levels accepted in the GEM project.

- **Availability** Services provided by devices in the system need to be available to be shared among the participants and other devices. Chung et al. (2000) define it as "Assured Service". Availability is achieved if the user can connect to, invoke and receive the desired services so the activity can be completed. In this thesis, the concept of accessibility is included as part of this concern.

- **Reliability** The system must provide the functionalities it has been designed for (Blanchard, 1992). An acceptable level of reliability in M-learning settings would be that the system will always supply services that provide the necessary information to achieve the collaborative activity.

- **Performance** The achievement of a requested job is under a certain lapse of time (Chung et al., 2000). We consider only performance in terms of response times. In our M-learning context, a tolerable service performance would be to provide an answer to the user within 10 seconds.

5.3 Uncertainties

In the process of a software product creation, some uncertainties appear early in development stages and may be mitigated immediately. For example, the activity content is unknown to the developer at an early stage, but this uncertainty can be mitigated by interviewing stakeholders, such as teachers, and by increasing the knowledge base during the design and development phase. Other uncertainties appearing early may be left aside and addressed later. Uncertainties in interface design concerning usability, for example, are often identified early in the design stage. However, they can often be left aside and addressed late in the development phase. Some other uncertainties remain until deployment time. One such uncertainty is the number of devices in which the application will be installed. Finally,
some uncertainties remain or surface at run time and thus, cannot be addressed until after deployment.

Hastings and McManus’ framework (2006) defines five uncertainty classes as presented in Chapter 3. In some cases, the boundaries between these classes are not strictly defined. Therefore, some uncertainties may be placed crisp and in more that one category. Nevertheless, the classification is not critical. In our research context, the critical elements are the uncertainties that cause risks threatening the activities.

As explained in Section 1.2, the outdoor M-Learning settings are inherently uncertain. Different uncertainties cause different risk depending on the scenario where they are identified. We present the uncertainties that have been identified in the different activities, a first step towards detecting collaboration risks. The classification below arranges the uncertainties according to their proximity to a category. However, as mentioned above, some uncertainties may be considered as candidates for being part of other categories as well.

5.3.1 Lack of Knowledge

Several examples of lack of knowledge were experienced during the activities performed in GEM. These suffered from the absence or insufficiency of information. Five uncertainties have been selected from the GEM field trials to exemplify this category.

**U1- Which hardware will be present in the activity?** Despite it was known which mobile devices had been prepared to be used in the GEM trials, not all of them were actually used. Furthermore, not all the devices had the same capabilities and configurations, differentiating characteristics were the model (Nokia N95, Nokia 5800, HTC Hero and HTC Tattoo), memory space and SIM-Card, to mention some. Some devices were left as spare units. Therefore, during design and deployment time, it was not known which devices would be used during each activity.

**U2- How many students will attend to the activity?** The GEM activities were performed during the Swedish winter season. In such conditions, it is high probable to get into a situation in which one or more students became ill. This information cannot be know in advance and becomes a lack of knowledge with respect to the number of participating students in the collaborative activity.

**U3- How many groups will be formed and how many students will be in each one?** This uncertainty is related to U1 and U2. An average of 12 students were involved in each one of the GEM trials. However, as a consequence of U2, it was not possible to know, in the design phase, how many groups would be formed. This was due to the lack of knowledge of the number of students available for participation.

**U4- How many devices are going to be available in each group?** Due to the two previous uncertainties, another uncertainty surface. We don’t
know the number of devices each group will use in the trial. This is due to the other uncertainties, U1-U3.

**U5- What data communication throughput will be available?** Another factor that cannot be known prior to runtime is the quality of service in the wireless network. Due to different factors, there is a chance that a desired GSM connectivity service, such as 3G or UMTS, will not be available during the field trial, and the network may only provide for GPRS based data connectivity.

### 5.3.2 Lack of Definition

Within this category, two uncertainties have been identified. A lack of definition occurs when aspects have not been decided or specified (Hastings and McManus, 2006). In our case, these absences occurred during the design, development and deployment phases, but did not become evident until during execution.

**U6- What is the required GPS accuracy to properly calculate distances?** The distance calculation relies on the accuracy of the GPS modules. However, the measured values in the device can vary based on different factors, such as meteorological conditions. The definition of the tolerance error was rather optimistic. In worse case scenarios, the tolerance value would be considered as incorrectly defined, which will lead to implications similar to what is described in Section 5.4.

**U7- Serial or parallel activity sequence flow?** Multiple groups were involved during the trials. However, it was not clearly defined during the application design phase in which order they would complete the activity. The groups could be organized and work either in serial (one group after another) or in parallel mode. In this case, the decision had not been taken prior to the trial.

**U8- Where will the activity will take place?** Location coordinates were used in the activity task. The activity may be initially designed to work in a specific location. However, there can be causes that force the activity location reassignment. These causes could be due to weather conditions, the harvesting season or system reuse by external learning teams in other cities and countries.

### 5.3.3 Statistically Characterized Variables

Statistically characterized variables are those that cannot be known in detail, but their values can be bounded after statistical analysis (Hastings and McManus, 2006). The following are two uncertainties that belong to this category.

**U9- Will the battery last to support the activity?** Statistically, the battery life expectancy is given to a certain number of working hours. The M-Learning activities are designed to cover one or two lecture hours. In a
normal distribution, a large percentage represents mobile devices that can work long enough to support the entire activity. However, there are a remaining percentage of cases where the batteries will run out. This value can be statistically calculated.

\textit{U10- Will the used memory cards have malfunctions?} Hardware components may fail during their use. One of the used HW components in the mobile device is the Memory Card. Commonly, the manufacturers provide the reliability of their hardware components, including a correctness guarantee.

### 5.3.4 Known Unknowns

There are variables that are known to exist in the activity settings, but the values they are going to have remain unknown. These are defined as Known Unknowns (Hastings and McManus, 2006). Below we enumerate some examples from our experiments.

\textit{U11- Will the GSM connectivity be available throughout the activity?} Prior to our first trial, during a testing session, it was discovered that the GSM connectivity was down due to the technical problems of the telecom provider. It is not possible to know in advance whether external providers will experience technical difficulties or change their communication standards.

\textit{U12- Will the activity server be available all the time during the activity?} One of the components in the system to support the collaborative M-Learning activity is the Activity Server. This situation is similar to what is described in U11, the activity server may become unavailable due to multiple factors. Such factors include hardware failures, software bugs and even a power outage or Internet failures.

\textit{U13- Will a student close the application?} Human behavior is difficult to predict. During the M-Learning activities, the students carried mobile devices with them and had limited supervision on the actions they took. One example was turning off the device or accidentally closing the application.

### 5.3.5 Unknown Unknowns

To complete the picture, there are uncertainties that are not known (Hastings and McManus, 2006). Uncertainties in this category can be found by putting some efforts towards their identification and analysis. These unknown unknowns are not considered in this project, hence, we do not provide an example. However, it is not possible to identify the entire set of uncertainties that lie under this category, because they are infinite.
5.4 Risks

In the previous section we elaborated on the uncertainties that can be attributed to the settings in which our M-Learning activities took place. A set of thirteen uncertainties in relation to our GEM experiments were identified and classified. This section presents the risks that these uncertainties contribute with. Risks must be seen as potential threats to the desired QoS outcomes in relation to collaboration in M-Learning activities presented in Section 5.2. It is important to mention that uncertainties can cause multiple risks, and each risk could be caused by more than one uncertainty.

5.4.1 Failure

When the system does not provide the functionalities it was designed for, a failure occurs. Sometimes, the obtained results are not the expected ones. Several uncertainties, as those identified in the previous section, cause a risk of failure on the system provision for resource sharing, availability and reliability of resources. Below we present a list of uncertainties that could provoke failure in achieving the desired collaboration outcomes.

- **U6**- In case the GPS module was not accurate enough, the activity would be under risk of failure. High inaccuracy rates in the distance measurements, would threaten the learning activity, as it would affect the reliability for collaborative actions. An example would be to get 100m inaccuracy measurements in 30m distance calculations.

- **U7**- If one execution sequence was taken (i.e. a serial order) and the alternative is required (parallel sequence), the activity might lead to a failure, due to a lack of time, devices, etc.

- **U8**- A new location, in where to perform the activity, may have not been known and thus considered during the design phase. Changing the location of the activity may lead to situations where the application does not support the activity, i.e. if location coordinates were used in activity tasks.

- **U9**- If they could not be replaced, an empty battery would become a risk of activity failure.

- **U10**- A malfunction in the memory cards would become a risk of failure.

- **U11**- The lack of a GSM communication channel becomes a risk for a collaborative mobile activity, as it disables the required physical communication channel.

- **U12**- A failure in accessing or retrieving data from the activity server would cause a failure. For instance, the activity script would not offered to the students.
5.4.2 Degradation / Unexpected capacity

Degradation occurs when the system does not fulfill the expectations and when the capacities to provide system’s features decrease. On the other hand, some uncertainties can lead to unexpected capacities when the expected conditions are surpassed. Of the uncertainties presented, five become a risk for system degradation. The following is a list of the identified uncertainties that may lead to degradation.

- **U4**- Being uncertain about the amount of devices available in each group could become a risk of degradation if the number became low. However, it can become an opportunity for unexpected capacities, in case the number of devices is more than required, as it would provide more tools for collaboration in the group.

- **U5**- Degradation on the communication channel, which would occur when switching from 3G to GPRS, would become a degradation on collaboration due to reduced data throughput.

- **U6**- Previously it was stated that inaccuracies in GPS measurements could become a risk of failure towards reliability, one of the outcomes needed for collaboration. In case these inaccuracies are small, it would result in a degradation of resource reliability. To illustrate this risk, if the student was allowed to have a 5% error in his calculations, but the GPS inaccuracy exceeded this limit when providing a distance, the student would be required to retake the task, delaying progress.

- **U9**- The batteries can be replaced or charged in case they are emptied during the collaborative activity. However, this difficulty could delay the collaboration, as its availability is degraded during this period of time.

- **U13**- In case the mobile application got turned off. Degradation in resource availability would be found during the process of reloading the mobile application.

5.4.3 Need shifts

In some situations, uncertainty may cause shifts in stakeholders’ needs (Hastings and McManus, 2006). The following two uncertainties are examples that may cause risks to need shifts.

- **U7**- Initially, a M-Learning activity may have been designed to run on a serial sequence. However, due to a lack of definition during the design phase, there is a risk for shifts in the stakeholder’s need, which may result in requiring the activity to work in parallel. This risk affects the performance of the overall activity, due a lack of time to carry out the activities.

- **U8**- Similar risks may surface when the activity location has not been completely defined. A new area to perform the activity may not have
been known and is thus considered during the design phase. This may lead to need shifts and therefore required shifts to the application design. In this case, the shift may affect the availability of the resources to be used.

5.4.4 Market shifts

Market shifts are risks that may occur. The system may have assumed some requirements based on the current market scenario. However, uncertainties in relation to the market force developers to define assumptions about the market that could shift. One example has been found in the GEM trials.

- U11- New communications channels are appearing. Based on the assumption that the system should use GPRS or 3G GSM channels may face the risk of new technologies appearing in the future, not being supported by the initial design. Availability of resources may be affected by this risk.

5.4.5 Costs Increase

Uncertainties during design and implementation phases may force developers to make assumptions that risk, for instance, the application budget. This cost increase includes the need for extra budget and working hours. Uncertainties that may directly cost increase of risks with the aim of maintaining the resource availability are listed below.

- U1 - It has been mentioned in Chapter 4 that several devices were prepared to be used during the M-Learning activities, assigned to groups. However, not all the devices were actually used at the same time and it was not known which devices would be assigned to which group. This becomes a cost increase risk for the development phase, as it could lead to consideration of all possible hardware combinations present during the activity. The fact that it is not possible to know everything beforehand becomes a risk for the project budget.

- U2, U3, U4 - A comparable scenario represents the participants involved in the groups. A lack of knowledge of the number of groups, participants in each group and devices in each of the groups causes a risk of cost increase by demanding the study of multiple combinations.

- U8- A new area to perform the activity may have not been known and thus considered during the design phase. This may lead to an increase of the time needed to create a new version where the new area is supported, in order to maintain the availability of the resources.

5.4.6 Uncertainty and Assumptions in Design

The existence of uncertainties is due to knowledge limitations. Some uncertainties are identified, studied and solved during design and development time. If designers lack knowledge, the affected design decision must be
delayed or reasonable assumptions must be taken. Based on the current knowledge base, given uncertainties and assumptions, a best candidate solution may be selected from a set of possible solution candidates (Figure 5.1-a). The best candidate selection is based on the assumptions taken. Therefore, the best candidate is revised for each of the taken decisions, as new knowledge is derived and assumptions may be removed. The following formula represents the selection of a solution candidate to implement.

\[ f(K, U, A) = \text{BestCandidate} \]

During execution time, new information may be collected from the environment, the activity or the system. Assumptions, made during design, development and deployment, may turn out wrong or they are invalidated due to changes in the system or its environment. Such information can resolve some of the uncertainties present during design and development. In these new settings, the assumptions made during those phases can be questioned. A consequence is that the system built on these assumptions may not provide sufficient guarantees for collaboration in M-learning activities.

Figure 5.1-b illustrates the effect that uncertainties and knowledge limitation can cause by deriving to a questionable best candidate solution.

Table 5.2 presents the relation between uncertainties identified in the GEM trials with risks that they may cause. The rows present the identified uncertainties, which have been classified inside five different uncertainty categories. The rows in the table present the risks that can be caused: F (failure), D (degradation or unexpected capacities), N (need shifts), M (market shifts) and C (cost increase).
5 Discussion

<table>
<thead>
<tr>
<th>Category</th>
<th>Uncertainty</th>
<th>F</th>
<th>D</th>
<th>N</th>
<th>M</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of Knowledge</td>
<td>U1- Hardware in the activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>U2- Number of students</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>U3- Number of groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>U4- Devices per group</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U5- Communication performance</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Definition</td>
<td>U6- GPS accuracy</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U7- Sequence in task</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U8- Activity location</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variables</td>
<td>U10- Memory Card correctness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Known</td>
<td>U11- GSM availability</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>U12- Activity Server availability</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U13- Application turned off</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Uncertainties and risks relationship in the GEM trials

5.5 Mitigations and Outcomes

The previous section presented the risks that uncertainties may cause. These risks can directly affect the outcomes that are required for achieving our goals: collaboration. Pertaining to the scope of this thesis, risks have been considered in terms of their effect on the availability, reliability and performance outcomes.

In order to minimize the potential risks, organizational, methodological, and technological strategies can be applied as mitigations mechanisms. Mitigations mechanisms are solutions that provide some degree of assurance that some quality concerns are met. It is critical to motivate and demonstrate the importance to a system’s behavior, as mitigations often increase project cost (Hastings and McManus, 2006). Risks are mitigated throughout development or deployment, when knowledge is attained and decisions based on knowledge and assumptions are made. However, some risks always remain. The dynamicity in a M-Learning system and its environment also contributes with new risks that may surface due to changes in the environment, in the system itself, or in the system’s goals.

The existing mitigation mechanisms have properties that make them capable to addressing one or more risks in order to provide assurances for the affected outcomes. The use of multiple mitigations mechanisms addresses several risks. Using multiple mitigation mechanisms requires more studies to better understand problems and possible strategies to resolve them. However, this is not explicitly addressed in our work until now.

5.5.1 Modularity

A modularity mechanism allows the grouping of functionalities in interconnected modules. Modules with crisp and stable interfaces may provide for easier replacement of modules, as long as interfaces do not change. Standard interfaces are common in modular approaches, making it easier to connect new components to the system. A modular approach minimizes the risk of a complete failure, as it facilitates replacement of failing components.
by alternative ones. This approach is, therefore, suitable to combine with redundancy.

### 5.5.2 Redundancy

Redundancy introduces extra capacity to a system by providing two or more instances of a component or a feature. When combined with modularity, redundancy facilitates replacement of malfunctioning components to mitigate risk of failure or degradation.

Failures and degradation in services provided by the activity server (U12), or by other hardware components in the system (U6, U9, U10, U11, U13) can be mitigated by the combination of modularity and redundancy mechanisms. With the use of modularity and redundancy, the assurance of availability and reliability on shared resources can be improved (Figure 5.2).

![Figure 5.2: Modularity and Redundancy mitigations to address failure and degradation risks](image)

### 5.5.3 Serviceability

Serviceability is the separation of a module’s interface and its implementation. Supplying separate and stable interfaces provides for transparent service evolution. That is, serviceability and modularity combined allow the replacement of a module that provides one implementation with a revision. Replacing a module can be performed without creating ripple effects on any other service in the system. Some of the risks of failure, need shifts and cost increases from uncertainties identified in GEM may be addressed by a Serviceability & Modularity mechanism. For example, a replaceable service providing the activity script can mitigate such risks caused by U7 and U8 (Figure 5.3 below). These risks threaten the flexibility of a system and its evolvability by extension. Therefore, availability is indirectly affected.

Moreover, serviceability can be integrated with design choices mechanisms to provide for additional mitigation strategies.
5.5.4 Design Choices

Design choices offer the capability of having two or more alternatives to provide system features. This strategy, combined with a serviceability mechanism, may offer a system that provides a solution that better fits the system environment. Figure 5.4 illustrates the effects of applying design choice and serviceability strategies to mitigate risks in GEM. The added benefits, from multiple design choices, can mitigate the costs of creating several solutions for multiple students-group combinations (uncertainties U2, U3, U4). For example, different design choices may be selected to arrange the resources needed in U4 to allow distance calculations: three non-shared devices, more than three devices or three shared-with-other-groups devices.

5.5.5 Generality

The generality mechanism is characterized as using "standard systems and interfaces, rather than specialized ones" (Hastings and McManus, 2006). This mechanism mitigates the risk of market changes by using standard protocols and de-factor standard platforms. A risk caused by changes to the GSM infrastructure may threaten the application QoS. We may mitigate the market change effect on a system by providing a more general implementation that allows for system evolution to address the changes in the system environment.
5.6 Approach for Mitigation

Figure 5.5: Generality mitigation to address market shift risks

5.5.6 Upgradeability

The upgradeability mechanism facilitates system feature upgrades to provide for improved functionality in terms of performance, extensibility, reliability, etc. For example, in U1, the addition of new devices in the system can lead to an increase of cost in the development phase. However, upgradeability would make the system capable to upgrade and, by that, provide support for introducing new hardware in the system.

Figure 5.6: Upgradeability mitigation to address cost increase risk

5.5.7 Mitigation strategies in GEM

In the previous subsections we have discussed how mitigation mechanisms contribute to a system by reducing risks due to specific uncertainties in development, deployment, and at runtime. Risks can be addressed by one or more mitigation mechanisms. In addition, combination of mitigation mechanisms may further reduce the impact of such risks. Table 5.3 illustrates how mitigation mechanisms can be applied to the system used in GEM in order to guarantee support for collaboration in the activities. The table shows some risks that have not been addressed due to external actors, such as the GSM provider. This is the case for U5 and U11. M-Learning application designers have no or few opportunities to influence the GSM infrastructure it is dependent on.

5.6 Approach for Mitigation

This section presents the application of mitigation mechanisms in the GEM trials’ systems. We begin with an overview of the strategies applied in the systems. These are then connected and described in the context of a complete, initial solution proposal which has informally tested in controlled environments to study the solution’s feasibility and viability. We wrap up
5 Discussion

<table>
<thead>
<tr>
<th>Category</th>
<th>Uncertainty</th>
<th>F</th>
<th>D</th>
<th>N</th>
<th>M</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of Knowledge</td>
<td>U1- Hardware in the activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>♦</td>
</tr>
<tr>
<td></td>
<td>U2- Number of students</td>
<td></td>
<td>△</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U3- Number of groups</td>
<td></td>
<td>△</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U4- Devices per group</td>
<td>△</td>
<td></td>
<td>△</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U5- Communication performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Lack of Definition</td>
<td>U6- GPS accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>U7- Sequence in task</td>
<td></td>
<td></td>
<td></td>
<td>△</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U8- Activity location</td>
<td></td>
<td></td>
<td></td>
<td>△</td>
<td></td>
</tr>
<tr>
<td>Variables</td>
<td>U10- Memory Card correctness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>Known</td>
<td>U11- GSM availability</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>□</td>
</tr>
<tr>
<td>Unknown</td>
<td>U12- Activity Server availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>U13- Application turned off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

Legend
- ♦ - Upgradeability
- △ - Design Choice + Serviceability
- O - Modularity + Redundancy
- △ - Modularity + Serviceability
- □ - Generality

Table 5.3: Mitigations for potential risks in GEM

the section with a conceptual evaluation of the system, applying it to three different scenarios.

5.6.1 Mitigation strategy I - Multi-Agent System

Chapter 3 provided an architectural perspective on multi-agent systems. One agent autonomously performs the actions it was designed for in a "self-contained, interactive and concurrently" (Nwana, 1996) manner. The use of multi agent systems (MAS) provides a channel for communication between agents. With such infrastructure in place, a system becomes modular in terms of agents providing services to local components and to other agents. Redundancy mechanisms can be created on top of a MAS, combining the modularity provided by the software agents and utilizing of the communication channels provided by the infrastructure.


Another approach is the use of the JADE solution (Bellifemine, 2001; Bellifemine et al., 2007). JADE stands for Java Agent DEvelopment framework and, as the name suggests, JADE may be used to design and implement a MAS. The agent framework enables mobile devices to run one or more behaviors utilizing remote message calls, timers and events. Moreover, the platform facilitates service publication and discovery, by a Yellow Pages service, provided by a dedicated JADE agent (Bellifemine et al., 2007).

Figure 5.7 illustrates a simplified MAS in a mobile device scenario. The software-agent component can be implemented by the previously mentioned MWS or JADE approaches. Moreover, the communication mechanisms
use the HTTP protocol. The figure below depicts how the system uses redundancy, replicating a service on multiple devices.

![Simplified MAS in a mobile device scenario](image)

**Figure 5.7:** Simplified MAS in a mobile device scenario

Central nodes in a distributed system often become system bottlenecks. A MAS solution constitutes the foundations for a decentralized distributed system. Internal agents replication is possible in JADE (Bellifemine et al., 2010; Vogt, 2008). This is an important property to provide for replication, thus reducing the risk of MAS failure.

A MAS solution provides for modularity, redundancy, serviceability and generality. However, it lacks features that provide capabilities to detect and fix issues autonomously at runtime. This requires further investigations, which is presented in the subsequent section.

### 5.6.2 Mitigation Strategy II: Self-Adaptation

A self-adaptation mechanism enables a system to perform changes in its composition or behavior in an autonomous manner. Redundancy, for example, can be achieved with self-adaptive mechanisms as it allows autonomous changes in the system composition. Moreover, design choices can be offered to implement self-adaptive mechanisms in, for instance, a MAS. This may be achieved by switching agents’ behaviors or the overall MAS composition at runtime. Such changes are always triggered and governed by the system’s and environment’s current state.

Internal agents provided in JADE facilitate system and subsystem monitoring. An Agent Management System (Bellifemine et al., 2007) provides a list of the agents registered in the system. On the other hand, a Directory Facilitator agent provides a list of the services provided by the agents in the MAS. This knowledge is critical to a self-adaptive MAS as it describes which devices can be accessed in the system and, consequently, which services and resources can be consumed and shared.

Figure 5.8 presents a schematic self-adaptive mechanism based on the MAPE-K control loop (Horn, 2001). Agents in the devices monitor the environment and the subsystem. In a second phase, an analyzer agent determines whether the current status of the subsystem accomplishes the
activity requirements or not. A Planner agent determines a set of potential MVDs. In the example below we adapt Mobile Virtual Devices (MVD) (Gil de la Iglesia et al., 2010b) to provide them with sufficient capability to complete an activity. A MVD represents the composition of multiple mobile devices sharing services and resources. Finally, an Execute agent performs the required changes to the MVDs to adapt the subsystem to the current environment and system conditions and to guarantee a subset of the desired QoS outcomes.

**Figure 5.8:** Components for a MAPE-K control loop design

### 5.6.3 Mitigation Strategy III: Multiple Concerns

A self-adaptation mechanism is not self-sufficient in a M-learning context. A solution for a collaborative M-Learning system must consider multiple concerns. Multiple concerns may require additional analysis and adaptation, as two or more concerns may conflict. Applying self-adaptation mechanisms when dealing with multiple concerns is indeed a risk in itself and such risks must be mitigated. The execution of adaptation actions to resolve some issues related to one concern could negatively affect other concerns (Gil de la Iglesia et al., 2012). Figure 5.9 illustrates a set of MVDs arranged to cover three concerns and potential risk that may exist due to dependences between them.

### 5.6.4 Conceptual Architecture

We have introduced a set of components to mitigate the risks identified in GEM. We may now describe a conceptual architecture to depict the participating components, roles, behaviors and relationships to other system components. This conceptual architecture is a result of the experiences, investigations, and analysis reported in the previous sections. We have used it as the basis for a prototype implementation. The implementation
5.6 Approach for Mitigation

Figure 5.9: MVDs covering multiple concerns

has been used in some controlled GEM setting to provide initial feedback. This evaluation further discussed in subsequent sections.

The conceptual architecture, illustrated in Figure 5.10, combines the mitigations presented above. Based on the MAPE-K control loop, four components categories are introduced to allow for Monitoring, Analysis, Plan and Execution of subsystem arrangements to provide self-adaptation.

Agents in the MAS monitor the environment and provide information about their status in the subsystem. This information feeds the knowledge base during execution, providing an abstraction view of the environment and the subsystem. An MVD manager aggregates this information by informing about the existing MVDs and their members. This data is used by the Analyst agent. The role of the analyst agent is to verify whether the subsystem accomplishes the desired outcomes or not. It checks whether the existing MVDs in the system are capable of providing the necessary services and resources for the activity. An Activity Manager is responsible to declare the required services and resources for the activity, so the self-adaptation process can be achieved.

A Planner agent explores new, revised, MVD deployment candidates when required and supplies best-fit candidates to an Executor agent. The Executor agent executes modifications on the existing MVDs to adapt the system to the changed environment and subsystem conditions. An Executor agent can obtain information from one or multiple Planner agents, allowing the system to, in principle, deal with multiple concerns in parallel. The conceptual architecture is designed to be scalable. It provides capabilities for multiple Analyst and Planner agents, allowing for parallel concern analysis.

The architecture is completed with the Activity Manager agent and the Activity Server agent. The first actor’s role is to provide activity scripts to the system so that tailored MVDs can be created. It is important to mention that this service can be provided by an agent behavior and may, therefore, be replicated to other agents in the system. The Activity Server agent offers the information necessary to determine the services each MVD must contain.

In Figure 5.11 a FORMS (Weyns et al., 2011) representation is provided. It illustrates the self-adaptation mechanism. The MAPE-K control loop is
Figure 5.10: Components view for the self-adaptive MAS prototype
5.6 Approach for Mitigation

provided by JADE agents that monitor the environment, the JADE Directory Facilitator, a Concern Database provided by an Activity Manager and the Executor agent that controls the MVD Manager.

Multiple concerns are represented by the reciprocity loop on the Subsystem Control. We believe the system control loop can be extended to include multiple MAPE loops, which separate control loops based on concern. Moreover, the model presents the actors that maintain knowledge used in the reflection process. This knowledge is represented by the subsystem and environment models (Malenfant et al., 1996).

5.6.5 Conceptual Architecture Evaluation

In order to provide an initial validation of the proposed software architecture, three collaborative mobile scenarios are presented where collaboration is at risk due to design and development phase uncertainties.

- **GEM, resource disconnection**

This particular collaborative M-Learning activity requires a minimum of three mobile devices in each group and an agent providing the activity script to them. During execution time, the $N$ groups are supported by $N$ MVDs, which contain three mobile devices and one activity server, provided by a desktop computer. No restrictions are defined with respect to sharing the activity server in the groups; therefore, this is present in all $N$ MVDs.

During the trial, due to a software bug, the activity server stops providing the activity script service. The change is registered in the Directory Facilitator, as the service is not available in the system when the agent is disconnected. The Subsystem monitor identifies the change in the system, and notifies the Analyst agent, which verifies whether the change has affected the collaboration to a degree that the current MVD no longer complies with the activity’s QoS concerns. However, the activity script has been downloaded by one or more mobile agents, promoting them to become potential providers of the activity script service. The Planner agent proposes a set of candidate MVDs deployments where the Activity Script Server service has been substituted by a mobile device agent. The Executor agent selects one of the proposed candidate deployments and implements the changes through the Group Manager agent.

- **Manhattan Story Mashup: activity server failure**

The Manhattan Story Mashup (Tuulos et al., 2007) involved several players carrying personal mobile devices. Putting it simple, the participants in the activity got an embryonic story that they would extend by adding a new sentence. Thereafter, the story was transferred to another user to repeat the process until the story was considered completed. An activity server was in charge of distributing the stories and to request participants to get involved. In a second phase, the participants were requested to take pictures that related to terms used in the story.
Figure 5.11: Component mapping to FORMS representation model.
At one point during the activity, the activity server stopped. The problem persisted for some time until it was identified and the service was relaunched. During the down-period, a degradation/failure in the collaborative activity was experienced, as the participants could not interact with the system.

Let us now study this scenario from a different perspective, assuming that a MVD architecture was used to deploy the story mashup system. In this scenario, there would be a single MVD, in which all the mobile devices would be members, together with the Activity and Repository servers. The failure on the activity server would have been detected by the Monitor agent. Therefore, the Interpreter agent would have requested a Planner to look for agents capable of providing the missing services in the MVD. A mobile device could have taken the role of requesting interactions reducing the system degradation while the activity server was down. The Executor agent would select one of the MVD candidates where a mobile device would take on the server tasks.

In order to modify the original implementation of the mashup system to become self-adaptive, the mobile clients should be modified. Including and configuring an agent in each device would instrument them with sufficient means to deal with the loss of an activity server.

- English training for Chilean students: microphone failure

This scenario presents an envisioned scenario for conducting collaborative English training activities for native Spanish speaking students. This scenario will take place in the coming months as part of an on-going research project. The activity goals are to practice how to pronounce and recognize numbers in the English language. The activity considers 3-peer groups, where the participants have to sort three given numbers. The interaction between the students and the number in the activity is done using a Speech-recognition service. Therefore, the students are required to pronounce the numbers correctly to place them in order. Each student is responsible for pronouncing one of the numbers. This aspect motivates the students to correctly pronounce the numbers. To motivate the correct understanding of the number, the students only see the number that they have been assigned. They have no access to their colleges’ numbers. Therefore, this information must be shared before the sorting phase. So, the other two numbers should be communicated between the students, but only oral communication through the mobile phone application would be allowed.

The activity provides three mobile devices to the students as their tool for visualizing the given number, recording their pronunciation (microphone) and playing the peers’ recordings (speaker). To be able to carry the activity out, three microphone services, three speaker services, an activity script service and a speech-recognition service are required. This becomes the criterion for the creation of the MVDs.

The activity script is divided into two phases. The first one assigned to create a knowledge base containing the three numbers in the activity. The
second phase is assigned to sort the three numbers. The activity script is as follows:

1. Information sharing phase
   (a) A number is presented to each student.
   (b) Each student reads the given number in English through the mobile device to share the given number with his/her group.
   (c) A speech-recognition service validates that the number has been correctly pronounced before it is shared with the other two members.
   (d) At this stage, the students know the three numbers in the activity and so they know their turn in the sorting phase.

2. Number sorting phase
   (a) Given his or her turn, each student pronounces the given number again.
   (b) A speech-recognition service validates that the number has been correctly pronounced before it is shared with the other two members.
   (c) An activity server validates that the numbers have been correctly sorted.

The failure of one mobile device becomes a risk of failure for the collaboration within the activity in the affected group. However, using our suggested architecture, this risk can be mitigated. Once the device fails in providing its services, the Monitor agent will detect changes in the affected MVD. A notification to the Analyst agent is performed, to evidence that the affected MVD does not fulfill the concerns set for the collaborative activity. Therefore, the Planner agent will intervene and plan alternative configurations where the MVD recovers the missing services. A potential solution is to substitute the lost services in the affected MVD by including a mobile device that is already assigned to another group. The Executor agent will communicate the changes to the MVD manager, so the new configuration takes place.

It is important to notice that this situation implies that a mobile device may belong to two different MVDs, and this mobile device is then shared by two students groups. Under these conditions, a failure in supporting collaboration in one group is avoided. However, the configuration represents degradation in the collaboration of the two involved groups, as one mobile device needs to be shared. With this illustration, we want to emphasize that under different conditions and having new knowledge and uncertainties, a different design choice becomes the best candidate solution for the system, even though it represents degradation from the previous status. In this setting degradation is preferred over failure.
5.6 Approach for Mitigation

An activity server scenario has been put in practice in GEM under controlled settings to validate the viability of the solution. As exemplified by the Manhattan Story Mashup, this scenario is not unique. However, deeper analysis is required to evaluate the implications of self-adaptation mechanisms with multiple concerns. The third scenario is in its current for a plan. Future evaluations will demonstrate if MVDs are a viable approach when we loose connections to mobile devices, which, as a consequence, may result in multiple MVDs including the same mobile device.
Chapter 6

Conclusions and future work

Chapter 5 showed how, during design and development time, the knowledge is limited due to the existence of uncertainties. These uncertainties may force designers and developers to define a set of assumptions in the creation of a software product. However, these uncertainties are the cause of risks that can affect the desired application outcomes during execution time. It is at runtime when some of the previous uncertainties can be solved. New information is then gathered, thus increasing the knowledge base and consequently decreasing the number of uncertainties. Mitigation mechanisms address the risks that threaten the activity outcomes. In the previous chapter we have also described how a self-adaptive based solution can help in mitigating risks for collaborative M-Learning applications, as they will allow updating the system composition and behavior based on the new information obtained at runtime.

Based on the analysis performed on projects in the M-Learning field and through the experiences gained during the GEM project, a hypothesis for the continued research in this line of exploration has been formulated.

A self-adaptive MAS system with support for multiple concerns provides assurances for collaboration in M-Learning activities.

6.1 Conclusions

This section connects the research questions that guide the thesis with the results we obtained. The results are discussed under each of the research question presented below.

Q1 Which architectural concerns are critical for collaboration assurance in M-Learning systems and which factors affect these concerns?

This question searched for the technological foundations to support collaboration in M-Learning activities. This is an initial step necessary for the formulation of the coming research questions. In Chapter 3, an analysis of the collaboration process was provided and some technological aspects to support collaboration were described. We identified the need for supporting cooperation, coordination and resource sharing in M-Learning activities. Coordination and cooperation aspects imply that the system needs to support communication channels which participants can use to share resources
(e.g., activities, knowledge, activity outcomes). Moreover, the need for resource sharing implies the importance of QoS attributes such as availability, reliability and performance.

**Q2** Which architectural patterns and mechanisms exist that enable collaboration guarantees in M-learning systems?

To guarantee collaboration in M-Learning systems means that availability, reliability and performance of resource sharing need to be ensured. Chapter 5 elaborated on uncertainties that cause risks that threaten outcomes. A description of uncertainties and risks identified in GEM was provided to characterize mitigation mechanisms that reduce and sometimes even remove risks. Uncertainties can be identified throughout development. If possible, some can be resolved during initial stages of development, or they remain unsolved for long. However, some uncertainties remain at runtime, becoming a risk for the requirements that support collaboration. An in-depth analysis, step-by-step, lead to the identification of a principle solution candidate, a self-adaptive multi-agent based system with support for multiple concerns. Inspired by this, we proposed a decentralized distributed system to mitigate the risks in M-Learning activities and presented three preliminary validation scenarios.

**Q3** What is the state of research and state of practice in M-Learning for collaboration assurances?

A review of the state of the art in TEL and M-Learning projects was presented in Chapter 3. A selection of forty publications from six journals and conference proceedings was the basis for the review. The analysis showed that current research efforts lack assurances for activity collaboration by a supporting system solution. The study found that it is a common practice to ignore uncertainties in learning activity environments, describing idealistic scenarios. Few studies have shown interest in dealing with risks that may arise during execution time. However, the study maintains that tailored solutions have been designed to address single uncertainties, such as device connectivity. However, no studies appear to address multiple concerns.

The results constitute the underpinnings for our answer to the main Research Question (RQ).

**RQ** What are the features in a system that can make it possible to guarantee collaboration in M-Learning activities?

A system that guarantees collaboration in M-Learning activities needs to be able to alleviate risks caused by uncertainties present during design, development and runtime. The following list summarizes the features, which we identified so far, that a system requires to ensure collaboration in such settings:
• **Support for Resource sharing**
  Resource sharing is a key feature for collaboration; therefore, it needs to be guaranteed during the activity.

• **Alleviate risks at design, development, deployment and execution time**
  Uncertainties lead to risks that affect the guarantee. Therefore, mitigation mechanisms should be in place to reduce the risk influence and to alleviate risks with respect to resource availability, reliability and performance. Mitigation mechanisms can be applied during all phases in development, but uncertainties that are present until runtime need special attention.

• **Avoid central nodes**
  A chain is only as trustworthy as the weakest of its links. The presence of central nodes becomes a potential bottleneck and point of failure. A reliable system must avoid the existence of central nodes, but consider decentralization.

• **Monitor the environment and subsystem**
  Dynamicity at runtime entails uncertainties. To decrease the risks consequences, caused by uncertainties present at runtime, the system and its environment require monitoring mechanisms to identify changes from the taken assumptions.

• **Provide mechanisms to adapt the system at runtime to maintain collaboration aspects**
  In addition to a monitoring system, self-adaptation mechanisms must be included to react to threats and provide assurances for the QoS attributes needed for collaboration.

• **Provide mechanisms to adapt the system for multiple concerns**
  Moreover, the self-adaptation mechanisms should pay attention to multiple concerns, such as multiple resource availability, reliability and performance.

### 6.2 Future Work

The conceptual architecture proposed in this thesis represents the results from the different explorations that were carried out. Further studies and design work are required to further develop the reference architecture. We plan to focus our attention on self-adaptation mechanisms for multiple concerns. Future directions for this research include:

1. **Multiple-concerns**
   Each single concern is indeed a challenge that requires years of work to fully understand and master. In our future work we plan to address aspects of multiple concerns. This means that we focus on the multiple concerns problem without providing complete support for specific
concerns. Mitigation alternatives should be studied as well as the challenges and benefits from their interactions to cover the combination to ensure availability, reliability and performance.

(a) Availability
(b) Reliability
(c) Performance
(d) Combination of the three

2. Refine the conceptual architecture

The proposed architecture in its current stage requires a complete redesign to meet our quality standards. The studies and design work on multiple concerns will provide insights that we will include in the reference architecture. Aligned with the aspects explored in the study, the refined conceptual architecture should include mitigation improvements for availability, reliability, performance and their interactions. We also plan for a more standardized architecture description, addressing stakeholder concerns in separate views.

(a) Refine and document the reference architecture
(b) Improve mechanisms for multiple concerns
(c) Improve mechanisms for self-adaptation

3. Refine implementation

In this thesis a prototype has been created to strength the validity of the model. After identifying modifications on the conceptual model to improve the placed mitigation mechanism, an upgraded and advanced version of the prototype will be required.

4. Evaluation

Finally, an evaluation of the suggested approaches needs to be accomplished. Based on the modifications implemented in an improved prototype version, three different evaluation phases are required.

(a) Under controlled settings

In the first phase, a laboratory evaluation should be performed to analyze the outcomes when applying mitigation mechanisms under controlled settings.

(b) In real settings

Thereafter, an evaluation under real scenarios is crucial to validate the suggested solution. In a real scenario, the settings are not well-controlled, but fuzzy, and the solution can be tested towards a large number of existing uncertainties.

(c) In other domains

Aligned to this last evaluation, an alternative path is to investigate whether this architecture is useful in other domains. It is important to notice that the aspects and results presented in this thesis are
not unique to the field of TEL and M-Learning, but can be applied to other fields such as e-health (Eysenbach, 2001), military and autonomous vehicles (Garbinato et al., 2009; Whittle et al., 2010) among others, where collaboration and distributed systems are also present.
Bibliography


BIBLIOGRAPHY


Vogt, J. (2008). *Jess to JADE Toolkit (J2J)*, volume 1. Department of Informatics, University of Fribourg (Switzerland), Switzerland.


Yau, J., Gil de la Iglesia, D., and Milrad, M. (2011). Identifying the potential needs to provide mobile context-aware hints to support students’ learning. *International Conference on Computers in Education*.


Appendix A

Glossary of Terms

Cloud Component that, through the use of a network and deployment-transparently to the user, provides services such as computational unit, shared resources and information.

FORMS A FOrmal Reference Model for Self-adaptation (Weyns et al., 2011).

JADE Java Agent DEvelopment framework (Bellifemine et al., 2007).

MAS Multi-agent system.

M-Learning Mobile Learning. Use of mobile technologies in education.

Mobile Computing Use of technology being mobile (Weiser, 1993).

MVD Mobile Virtual Device. Virtual composition of mobile devices to create an entity in where resources are transparently shared.


Pervasive Computing The use of technology which effectively invisible to the user (Weiser, 1993)

QoS Quality of Service.

Resilience Feature that brings elasticity to a system to overcome undesired states and conditions.

Self-adaptation Characteristic in a system to autonomously change internal features based on external conditions and events.

TEL Technology Enhanced Learning. Support of learning through the use of any kind of technology.


Ubiquitous Computing The use of technology which is mobile and effectively invisible to the user (Weiser, 1993).

Uncertainty Aspect that is not known, or known only imprecisely (Hastings and McManus, 2006).
Appendix B

Publications
Publicaiton I

On the Issue of Reusability and Adaptability in M-learning Systems

On the Issue of Reusability and Adaptability in M-learning Systems

Oskar Pettersson & Didac Gil
Center for Learning and Knowledge Technologies (CeLeKT)
Linnaeus University
Växjö, Sweden
oskar.pettersson@lnu.se, didac.gil-de-la-iglesia@lnu.se

Abstract— This paper presents a brief overview over some of the conceptual and technical issues associated with reusability and adaptability. The specific orientation of our efforts is oriented towards systems supporting Mobile Learning. The field of M-learning domain has during recent years been suffering from severe configuration problems caused by the plethora of mobile devices present on the market. This paper presents a conceptual architecture that has been conceived to remedy some of these issues. Furthermore, a possible instantiation of this architecture with self-adaptivity features is discussed and a first prototype implementation is presented. The notion of ecosystems is introduced, as we believe it is an important building block towards increasing the reusability of components and content in this field where composition is one of the major issues to be addressed.

Keywords: M-learning; Self-adaptation; Software Ecosystem;

I. INTRODUCTION

The field of mobile learning (M-learning) has been rapidly developing during the last few years. Recent mobile learning projects now utilize a wide array of devices including mobile media players like iPods, PDAs, and mobile phones. Many of the efforts in this field also focus on different techniques ranging from using SMS to teach language, to using “smart phones” for teaching primary teachers energy management [1].

The variation among devices and other technologies on one hand, and the usage on the other, creates a massive configuration problem finding the best technology supported by a device for a given activity in a specific context. The variation of devices also has an effect on M-learning. Generic portable devices are extended with new technology, such as sensors. In parallel, dedicated portable “science platforms” like NOVA 5000 and Pascos’ SPARC are introduced. These developments make technology more versatile than ever.

Broadening the scope in this fashion causes increased variability, which introduces a number of challenging problems. For instance, devices are required to use specific protocols in order to utilize different services supported. This fact has the unfortunate side effect that the majority of M-learning activities developed today are tailored to specific devices or platform such as Symbian or Windows Mobile and specific versions thereof.

The notion of reusability has gained a lot of attention in the Technology-Enhanced Learning (TEL) research community. Recent efforts in this specific area of research mainly focus around adapting learning content via different approaches in various contexts [2, 3, 4]. Some of the suggested approaches provide significant advances in the field but still several problems remain. For example, orchestration of different devices and device specific deployment are areas that have not been left deeply investigated.

The contribution of this paper is two-fold. First, we discuss reusability in M-learning including both content reuse and reuse of M-learning components identifying a number of challenging problems. Second, we propose a conceptual model that leverages reuse of M-learning components and content.

This paper is structured as follows; sections II will elaborate on the motivation and background for the notions presented and discussed in this paper. Section III and IV present a conceptual architecture and an instantiation of this architecture which is then, followed by a depiction of a first test implementations of this architecture. Section V briefly discusses some of the related efforts exploring similar directions of research while the last section provides a discussion and elaborates on our future lines of work.

II. REUSE IN M-LEARNING SYSTEMS

Content reuse in the field TEL is a rather mature area with proven methodologies and tool support. In M-learning, content reuse is necessary but not sufficient for developing a complete system. In this section, we present and elaborate on a few examples of M-learning activities related to our recent research efforts in order to further validate our approach, enumerating a number of critical issues.

The set of learning activities to be presented relates to the trials conducted as part of the Advanced Mobile and Ubiquitous Learning Environments for Teachers and Students (AMULETS) project [5]. These trials have been conducted during a period of three years with different age groups of students. The first of these trials was conducted in 2006. It utilized S60 Nokia phones, GPS devices and used a
standard web browser to support different activities and tasks in the field of natural science. This implementation included a browser component, a server component a component for recording GPS data and a visualization component for the post-activity. It also had a component for interpreting visual tags called Semacodes. The subsequent activity was conducted later the same year and focused on the subject of history. This trial had similar components but lacked a GPS component. However, it did include a camera component which enabled the groups to document the activity, a chat component a collaborative component that enabled an indoor and an outdoor group to collaborate on questions and another component for the indoor client. The third activity we carried out was on the subject of biology and the target group was university students. This activity did not utilize GPS either but expanded upon the notion of content creation and included a module for audio recording, which could be listened to in real-time by the indoor group.

These activities, when split into modules, do to some extent seem similar. The fact is that they share very little code even though they have been developed from the same family of learning activities supported by similar technologies. Between the first and second activity the entire code base was changed. Between the second and third trial, the functionality of the activity server was refined several times, the components for the picture taking application changed entirely due to move to a newer mobile platform and the indoor client had to be remade from scratch.

The problem with reusability of software components has been evident in all three cases. We discuss two types of reuse, reuse of content and reuse of TEL components. TEL Components can be characterized among other features by the behavioral logic capturing activity flow and the presentation logic for content. In all cases, clients TEL Components had to be tailored to specific devices and would not be robust if any other TEL Component would change. This specific issue in M-learning is to some extent coupled with the fact that mobile devices and software are not standardized to the extent desktop computers and applications are. The variations in devices’ capabilities can only be solved by a configuration process that assembles TEL Components based on current device capabilities.

M-learning is often criticized for being too technology centered but the resolution of this configuration problem is of software centric nature. Providing highly configurable systems implies that devices hosting these systems must provide provision for flexible deployment and instantiation of TEL Components. In more traditional systems, TEL Components are statically bound which results in a reduced flexibility.

Issues with tightly coupled systems have been identified long ago. One of the most popular notions used today to handle complex distributed systems is Service Oriented Computing (SOC) [3]. In SOC, a system is divided up into cohesive units and processes are deployed onto these units via Web Services or other means. The notion of SOC fits well with the observations made and the requirements for loosely coupled systems. Further, SOC claims to support reusability of services since these are cohesive units with a stable interface. Changing one of these services should not affect any other as long as the interface is preserved. As [3] points out, there are currently very few systems in the TEL field that utilizes this approach.

Another concept that would positively contribute to the development of systems in this area is the notion of Software Ecosystems and Ecosystems in general as pointed out in [6]. Here the focus is on facilitation of other external actors and to enable them to modify and expand parts of an existing system. Something would be beneficial for this kind of system as is argued in [6].

We already discussed the required support of device variability. In addition, the field of M-learning faces a number of challenges related to reuse of TEL Components, for instance we must address efficient reuse of behavioral components and dependencies in-between content and TEL Components. With these notions in mind, the next section of the paper will depict the ideas and concepts introduced in this section in order to address some of these issues and problems identified in the earlier sections.

III. A CONCEPTUAL ARCHITECTURE

The previous sections have highlighted problems in TEL Component reuse, for instance how activity and device orchestration are engineered in most current M-learning efforts. To remedy these issues, reusability should be promoted and better supported in M-Learning. To that end, we propose a conceptual architecture which models the M-Learning domain. A domain model is the first step towards a better understanding of the field, which in the long term will support the development of additional concepts and technologies that leverages reuse.

![Figure 1: A Conceptual Architecture for M-learning](image-url)
A. The concepts

The conceptual architecture described in figure 1 captures our understanding of the M-learning concepts. The concepts were conceived with reusability and extendibility in mind. Devising and constructing a M-Learning ecosystem based on this architecture is one of our prioritized goals.

The conceptual architecture is build around three main concepts: logic, resources and agents. The key notion is the concept of Resource. A resource is basically a system building block that needs to be orchestrated. A resource may also be reflexive, which implies that a resource can contain other resources. One example of a resource is an Activity. An activity can group tasks, media and other objects, even other activities. Figure 1 lists a few resource concepts but this list can (must) be extended with other concepts as required. Formalized Learning Objects [6] could be one type of resource, a photography-module or a GPS module could be another.

Continuing the description of the conceptual architecture, the Agent concept depicts different aspects of users, the grouping of these and their devices. This concept also deals with different agent roles, which can be seen for example in IMS-LD [7]. In the context of our efforts, the term Agent should not to be confused with the term agent or intelligent agent used in fields like artificial intelligence. The term agent here is derived from the ontology FOAF (Friend Of A Friend).

Finally, we have Logic as the third concept. Logic is what connects resources with agents and, to some extent, resources with resources. Two or more resources may be orchestrated via the Connector concept, which offers and extended orchestration mechanism over different concepts in the system. Logic also offers additional control over the different parts of the system. The notion of connectors is, to our knowledge, a novel approach in this context. The approach of putting more emphasis on the connections between different parts of the system stems from the fact that device interoperability is a critical concern in M-learning contexts. Another reason for this novelty is that the highly contextualized nature of M-learning, where for instance resources might need to be re-initialized dynamically. This element of self-adaptability will be further addressed below. The next section depicts the way that the conceptual architecture could be instantiated.

B. Instantiating the Conceptual Architecture

The previous section introduced a number of M-Learning concepts and structured them in a conceptual architecture. This section further elaborates on the proposed conceptual architecture and the potential benefits system designers could obtain when using it as a basis for their designs. In figure 2, we depict a representation of the learning environment ecosystem and its different components. This ecosystem architecture is the result of an instantiation of the concepts found in the conceptual architecture.

One key feature of this architecture is the use of the two central concepts, namely Resource and Connector. Now resources can be independent, grouped or coupled to form larger resources and even new applications with connectors. This increases reusability, as even small pieces of content can be included in several different larger applications or aggregations.

There is also the notion of Agent, which deals with the users and how these are modeled. The Device concept is also present. It provides a meta-model of devices and is used internally in the system. This is useful introducing the context of the system as an aspect. There are also the Rule and Logic concepts, which orchestrate parts of the system and provides additional controls and features. Rule is a central notion for the connector orchestration and for defining rules that govern the flow of instantiated activities, which is crucial for the activity. To utilize the variability to a maximum, the ecosystem architecture makes use of a dedicated self-adaptation agent. This agent is conceived and inspired by the theories proposed in [9], where a monitoring cycle is combined with an evolution process. Briefly, it works by monitoring instantiations of system concepts for specific state, evaluates call for actions and if something is found measures in the form of new or modified instantiations, and adjusted connectors are taken before the modifications are deployed.

This approach makes the architecture resilient to unexpected irregularities, as it is able to compensate and correct these changes when possible. It also facilitates smoothing of the coupling between resources and components which are required when external actors are brought into the Ecosystem. This notion of self-adaptation also makes the architecture more resilient to changes in the resources over time, which means that resources and components can be reused in later aggregations resources and components.
The architecture also supports the usage of external clients via an API that can also be used by external services that can for example visualize data provided by the system. Furthermore, one of the main benefits of adhering to this conceptual architecture with its self-adaptation features is that constructing an authoring tool for teachers to build their own activities should prove far easier. The self-adaptation lets the system compensate for many of the faults in configuration that could emerge from such a tool.

The next sections go on to elaborate and describe the first test implementation based on this conceptual architecture.

IV. A PROTOTYPE IMPLEMENTATION

The first test implementation of the ideas introduced in the previous sections was conducted as part of the AMULETS project and is referred to as GeM (Geo-Math). The activity was carried out by a group of 12 – 14 year olds students during a whole day. This learning activity was split into three parts consisting of an introduction, one outdoors activity for collecting data and measurements and one session in the classroom to utilize the data from the outdoors activity in order to construct a 3D model of a building with the Sketch-up modeling software. The architecture presented in this paper was implemented and utilized in the outdoor stage and will only depict that one further.

The outdoor activity was conducted using two different mobile devices where one was used as an activity client to receive and answer tasks and to receive measurements from the other phone. The other phone was mainly used as a point of measurement. An example of this is when the students were tasked with measuring the area of a large circular field on the campus. This was done by positioning one device at one end of the field and the other device at the opposite end, measuring the distance via GPS. The activity went without any trouble in the architectural parts of the system.

The implementation made for the GeM activity was far from feature complete with respect to the instantiation of the conceptual architecture presented in the previous section. The central notions and concepts described in section III were tried and even though no self-adaptive features were implemented, the notion of them and how they can further improve the system proposed in this article seem like a viable option for future implementations. This will be further discussed in section six where future lines work will be presented. This implementation represents the first step towards the deployment of the conceptual ideas presented in this paper.

V. RELATED WORK

The efforts outlined in this paper are not isolated and share characteristics with other research projects in this area. One early effort of adaptation in E-learning is the work by Brusilowsky[11]. He describes a web-based portal for providing personalized learning content called KnowledgeTree. This is achieved with distributed activity servers that are used for serving educational activities. The application also has a student model server, which is used to collect student performance data. This effort share some common features with the ideas presented in this paper. A few of the most notable similarities are the focus on reusability of content and the distributed nature of the
system. The differences are quite significant, while KnowledgeTree focuses on E-learning means that its domain differs slightly from the architecture presented in this paper that is leaned towards M-learning. Another difference is that KnowledgeTree is focused on content adaptivity whilst our architecture is focused on content and component adaptability.

Zhao and Okamoto [2] presents an approach to adaptive mobile learning that focuses on utilizing the user agent that is supplied by web browsers. This user agent is then used to modify content to better fit the device it is being sent to. This approach focuses on adaptation of existing content. Compared to the approach proposed in this paper, it is different but it introduces some interesting notions. Another interesting effort conducted as part of the SCY project by Lejune and colleagues [12] uses a more flexible approach to learning scenarios. It also introduces the notion of “Learning Activity Space”, which realizes this flexibility. This work has similarities to the one presented in this paper but it lacks self-adaptive features.

VI. DISCUSSION AND FUTURE WORK

The findings in this paper suggest that reusability and self-adaptation are not prioritized notions in most M-learning projects. In the general field of TEL and to some extent M-learning there are some notable efforts in this direction, some of which have been brought up in this paper. More are likely to follow as this is an emergent part of the field. The main concerns addressed in this work are reusability and adaptability in the domain of TEL, focused primarily on M-learning. The conceptual architecture and its instantiation presented in this paper are a first step towards trying to find some viable solutions to these issues. Several novel notions have been discussed for the domain in order to achieve this. Self-adaptation is one of the most prominent. The fact that this is inherent in the Ecosystem mindset makes it a promising path to follow for future development. As brought up in the brief related work section, no system is currently handling the configuration problems inherit in M-learning and this is one of the aspects that this system promise to deal with.

The initial findings presented in this paper are only the first steps for the developing architecture. The scope of this paper is therefore limited to the conceptual architecture and some of the benefits that can be derived from it, exemplified by an instantiation and two implemented scenarios. Implementation details like support for certain standards for interoperability like IMS-LD, have for example been intentionally left out and will be expanded upon in future efforts. Other things that will be investigated further are the notions of linked data and how this impacts the exposure of the resources. This is to some extent further elaborated in [13]. The underlying ontologies in the future implementation also needs to be further investigated. Another important aspect that falls under future work is the fact that this architecture should let other stakeholders utilize the architecture and provide their own resources and components in a flexible fashion. The importance of a healthy Software Ecosystem in this domain has so far not been a research focus in the M-learning field. E-learning is a more mature field in this respect, but still there are a lot of challenges that are common for both fields [6].

REFERENCES

Providing Flexibility in Learning Activities Systems by Exploiting the Multiple Roles of Mobile Devices

Providing Flexibility in Learning Activities Systems by Exploiting the Multiple Roles of Mobile Devices

Didac Gil & Oskar Pettersson
Center for Learning and Knowledge Technologies (CeLeKT)
Linnaeus University, Växjö, Sweden
didac.gil-de-la-iglesia@lnu.se, oskar.pettersson@lnu.se

Abstract—The wide adoption of mobile and wireless technologies allow for users to have access to learning resources and generate digital content at any location and time. An immediate implication of this latest trend is the need to create and deploy learning activity systems that offer high level of flexibility. This flexibility can allow for the creation of adaptive activities, something highly required in dynamic learning environments such as those in the field of M-learning. Mobile devices can provide flexibility to activity systems by adapting their behavior on a context basis. To give an example of flexibility in the mobile device behavior, we present the FLexible Activity Client (FLAC). FLAC is a mobile application that allows the mobile device to switch from client-role to service-provider-role behavior. We describe our proposed architecture and its implementation and present the results of our initial tests.

Keywords: Learning Activity System, Software adaptability, mobile collaborative learning, mobile web servers

INTRODUCTION

The increasing adoption of mobile devices and applications combined with the wide availability of wireless technologies provide new possibilities for supporting learning across a variety of settings. These developments allow for users to have access to learning resources and generate digital content at any location and time. One of the implications of this latest trend is the desire to allow for reuse of the learning objects created in these emerging M-learning activities. Another consequence is the need to create and deploy learning activity systems that offer a high level of flexibility. This flexibility will allow for the creation of adaptive activities, something highly required in dynamic learning environments such as those in the field of M-learning.

While looking in more details at the notion of flexibility, it can be said that it is closely related to context. In many cases, flexibility is applied for dealing with adaptive content (e.g., delivering a particular digital object in connection to a location), while mobile devices have still well predefined roles in the flow of the learning activities. Inside Learning Activity Systems, hereafter LAS, we can find mobile devices playing the role of clients in the majority of the cases [1,2,3]. We can also find mobile devices used as master devices to control the learning activity flow or to provide learning objects within those activities [2]. Even though we can find different mobile device behaviours during a M-learning activity, these devices maintain their role during all the time this event takes place. We define the notion of “single-role mobile devices” as mobile devices with a fixed behaviour within a specific learning activity.

Mobile devices are tools that can be considered as being important components of a LAS. Thus, they need to operate in a proper manner with the rest of the system in order to achieve and fulfil the functional requirements that LAS has. The limitations of single-role mobile devices in terms of flexible behaviour can create some constrains related to the overall performance of the system. Therefore, there is a need to think about how to conceive and design a flexible architecture that may allow and support multiple roles in one single device. An adaptive multiple-role device inside a LAS could keep the malleability and flexibility of the whole system, thus opening locked doors to new activity designs.

Until now, the notion of context [4] has been used in the field of Technology-Enhanced Learning (TEL) to guide the adaptation of learning objects and to model the learning flow and the task rules. Apart from that, context can provide the required information to be used as an input for taking the decision about the role a mobile device could play at specific moment during a course of an activity.

In the following section, we present a brief overview of M-learning projects where single-role mobile devices were used and contextual computing techniques were applied to provide the system with certain kind of context adaptation. In sections III and IV, we elaborate on and describe the current problems associated with the static roles of mobile devices in LAS. By doing so, our intention is to provide a possible solution that can provide flexibility to the system. In the section V, we describe the results of testing our proposed architecture and its implementation through its use in the Geometry Mobile (GeM) trial, an activity in which we explore how teaching and learning about geometry can be supported using mobile technologies. Finally, we lay out our future efforts with regard to the development of the multi-role of mobile devices and its integration in LAS.
BACKGROUND

Many research efforts in the field of M-learning are exploring how to achieve flexibility and adaptation in the design and implementation of LAS. There are several goals associated with this task and they can be listed as follows: (1) improvement for the adaptation of the resources used in the learning activities, (2) enabling the share of those resources among learning activities and projects. Such system adaptation requires information from the context in which learning takes place through the understanding of the physical-, knowledge- and social-environment so that actions can be performed in accordance.

As being able to “sense” context is one of the foundations for the existence of adaptability in M-learning, we apply Kurti’s definition of context [4], where context is the amount of “information and content in use to support a specific activity (being individual or collaborative) in a particular physical environment at a specific time”. Using this definition, context provides objects with a number of features classifying them in 4 dimensions, (1) the activity, (2) the person, (3) the physical environment, and (4) time.

Next, we will present three mobile learning projects in which context has been used to enable the adaptation of different modules inside M-learning systems.

Collpad [1] is a project for supporting learning in collaboration through the use of PDA and netbooks. It presents a system where collaborative learning and the sharing learning-objects produced by the learners are the main goals. The Collpad system allows learners to: (1) create answers to predefined questions, (2) share the answers between the rest of the learners in the group to start productive learning discussion, (3) reflect about the correctness of the answers and (4) choose the most appropriate one. In Collpad, we detect two roles assigned to the mobile devices. First, a master device that controls the activity flow and generates the activity questions and, second, slave devices that allow the user to collaborate to create answers (create a potential answer for the question received, to get a set of possible answers and to choose the correct answer to the question). In the system architecture, master and slaves can connect due to the existence of a local wireless network, where the master device behaves as a central answer repository to share the students’ responses. In Collpad, there is a combination of mobile devices playing complementary roles, but still single-role per device; slave (client) mobile devices for the students, and a master (server) mobile device for the teacher.

In the Collage project [3], a learning platform has been developed that uses mobile and positioning technologies to engage learners in explorative learning activities conducted outdoors. As described by [3], the Carnuntum scenario described a learning activity in the field of archaeology in which the physical location of the students is used to offer which questions and activities should be used. Moreover, contextual conditions are used by the client mobile device to decide the language used during the activity. “The 13-year old pupils played the game in German, the 17-year old students had to choose the game in English as additional challenge” [3]. In Collage, a number of contextual conditions have been used to determine where actions should be taken, and how the complexity of the activities should be set.

Pinetree [5] is a system designed and developed to organize and retrieve user generated content by using semantic technologies. Pinetree refers to resources’ context to give them meaningful information. To illustrate the concept, with Pinetree it is possible to include metadata to a picture to identify its author, to provide information about the task where the picture was taken and to attach the image geo-location, as some examples. This meaningful information is required to be able to retrieve learning objects, not only by their specifications (e.g. size, amount of colours and MB), but also by their meanings in their creation (e.g. conversations recorded in a Mathematics learning activity). Using Pinetree, one user is able to retrieve objects based on a set of criteria variables to be included in learning activities. As an example, using Pinetree it is possible to obtain images and videos created by people living in Sweden and created inside the scope of Nature Science.

All projects presented above have in common the use of context to adapt and reuse objects. Sharplest al., [6] presents a mobile learning model (adapted from Engeström) where the main components are: (1) tools, (2) objects and (3) subjects (4) control, (5) context and (6) communication. One could claim that a particular learning setting containing all these components will be able provide the flexibility needed in a learning environment.

![Figure 1: Task model for mobile learning (Sharplest al. [6])](image)

Current context-aware applications in the field of M-learning have been constructed having in mind its connection with (1) the objects used in the learning activities, (2) their subjects, also called learners and technology users, and (3) the control components in the learning activity system. Firstly, with regard to objects used in the learning activities, context is used to reuse objects generated in previous works into new activities (object selection). Querying for objects that satisfy a set of contextual conditions, as described in Pinetree, can be an example of such a reuse. Furthermore, context
helps to determine the proper visualization to apply on objects used in the activity (object representation). Secondly, with regard to the subjects intervening in the learning process, context is used to ask students to switch roles in the learning activities, allowing more advanced learners to explore and solve more complex tasks. Thirdly, with regard to the control components in the learning systems, context can enable tasks to be done at the proper location and to conduct activities that are adaptive to the situation, as some examples. Trying to relate these ideas to Sharple’s model, we can see that current projects have considered the links between Context with Control, Subject, Object and, indirectly, the Communication, to make those last to become context-adaptive. However, analyzing the model, one can see that the list of components influenced by the context is also including the “technological tools”. We suggest then that context should also have an influence towards some of the tools used in the learning systems, namely; the mobile devices.

**CURRENT PROBLEM**

One could consider that, within M-learning projects, context is being used to communicate to the user how to behave in the activity, what kind of task to perform and how to present the information for a number of particular activities. However, this modification of behaviour that we can identify in the objects, users and tasks, has not been seen in the mobile devices per se, which stills continue to be a “single-role” across the different activities.

While for Flanagan [7], context awareness for mobile learning allows the device to sense and adapt itself to the changing environment as well as the preferences and needs of the learner; we have not seen how those features have been implemented in current M-learning projects. Following Flanagan’s lines of thinking [7], mobile devices should be able to adapt their behaviour during activities too in order to provide the desired flexibility in M-learning.

To instantiate and illustrate this problem with one example, we will present a common scenario. A mobile device is running as a client as part of a learning activity and it is used to create content and access resources during the activity performance. The resources created during the activity are not reachable for the rest of participants in the activity. Therefore, this information cannot be shared until an “indoors” activity is performed, or after the information is centralized (asynchronous information sharing) through storing the digital content in a shared repository. In the design of mobile learning activities, mobile devices were conceived to have merely a client role. Therefore, they were not prepared to behave as a content service provider. This limitation brings the constrain that emerging objects will not be able to be reused in all their potential, as we will better present in the following section.

Summarizing, M-learning activities requires from learners’ collaboration across multiple different settings in order to obtain the best outcomes and results from each environment. However, from a technical perspective this has not been fully accomplished yet, as the mobile devices used in the activities are tied to a single role.

**ADDING FLEXIBILITY TO MOBILE DEVICES: MOBILE WEB SERVERS**

A fully adaptive learning activity system should contain mobile devices ready to change their role with the idea to get their behaviour aligned to the environment and learning goals on focus.

A richer learning activity system will allow mobile devices to switch roles during the activity performance, sometimes acting as mere clients in the system, others times acting as service providers and in others situation as content repositories, just to mention some possible roles. It is obvious that offering mobile behaviour adaptation requires from the mobile device to understand the environment where it is surrounded. As already mentioned earlier, this environmental information can be acquired using the multiple dimensions of context described by Kurti [4].

Allowing a mobile learning tool to share resources in a collaborative task is one of our main concerns. Those resources are sometimes freshly created media objects. However, after their creation, they become static. As static objects, they could be shared by the use of a central repository. On the other hand, resources could also be dynamic sensor values that are only meaningful when they are created on demand to have time-related accuracy. For those last scenarios with dynamic objects, a central repository is not a suitable solution for the intercommunication.

In opposition to a centralized repository solution, mobile devices should switch role and behave as mobile resource providers, and enhance the content-sharing offering real-time and on-demand responses. A possible answer to this question are Mobile Web Servers.

Nokia Research Center has created an application that converts a mobile device into a web server. This application got the name “Mobile Web Server” (MWS) [8]. Using this technology, we have created a client that allows the mobile device to behave as a client in the learning activity and to behave as a content provider to offer internal resources whenever the collaboration in the activity demands it. This switching of roles in the mobile device provides more flexibility to it and removes some boundaries and limitations as seen in the previous architectures described in earlier sections. As a result, this approach provides more flexibility to those learning systems in where mobile technologies are used.

In figure 2, we can see two mobile devices that illustrate our client solution. While one of them is playing a client role (phone on the right), the second phone (left) has switched behaviour into a service
The developed architecture was completed with the integration of the Learning Activity Server, where the logic of the activity is controlled, and the activity logs are stored [9]. The LAS provides to the FLAC with the information required to carry the tasks in the learning activities and to determine the switching role of the mobile devices. The proposed architecture has been tested in experimental settings during a trial conducted with several students. We will present these efforts in the next section.

**GeM Experiment**

In the spring 2009, the flexible learning activity system was tested in a Geometry and Mathematics activity. This trial consisted of three activities, two of them being conducted indoors and another one outdoors. A group of 12-14 years old students participated in the GeM trial. In the outdoors activities the students were divided in groups of three people, where they were expected to collaborate in order to carry out the tasks designed in the activity. Each team was provided with a minimum of two mobile devices prepared to perform the outdoors activities that would be done in the second activity. In the first activity, the participants of the trial were introduced to the technology and to the activity goals during a training process of 45 minutes. The content of the activities were not presented in this stage. The students just were mentioned that Geometry and Mathematics would be the topics of study and that tasks’ descriptions would be presented during the performance of the outdoors activities.

In the second activity, where the outdoors tasks were executed, the activities to perform in the mobile device screen were presented to the students. Within the tasks to perform, the students were asked to calculate physical objects’ perimeter, area and volume, where a distance measurement was always needed as a parameter of the calculation formula. Two mobile devices with built-in GPS were running the FLAC application and they acted as the tools used to perform such calculation.

In this task, the activity engine is the actor responsible to supply the mobile devices with information about how to perform the task. One of the mobile phones was designed as a client device, being the one to provide the question-answer interface, descriptions and hints. The second mobile device was designed to have the role of a service provider. (To clarify this idea, one mobile device has been symbolized as a client, while the second one as a mobile server).

The client device was forced to access the second mobile resources to get its GPS location. Such collaboration was done through the use of HTTP RESTful requests. The figure 4 describes a simplification of the sequence of actions done between the mobile devices in order to promote collaboration between them. This instantiation used the activity

![Figure 2: Activity context-based role architecture](image)

Using mobile devices as Web Service Providers, resources in mobile devices can be accessed by a combination of their Name Domain and a RESTful API call or a URI direction. A 3G, GPRS or WLAN link can provide the connectivity needed for the server to be accessed by content consumers.

Pettersson and Gil [9] present a conceptual architecture where Logic, Agent and Resources are combined in an Activity Learning System in order to providing alternative dimensions to the flexibility of learning activity systems. According to their proposed architecture, FLAC allows a mobile device to play an agent role and as a resource source guided by the instructions described in the context.

![Figure 3: Conceptual Architecture for a flexible Learning Activity System](image)

1 www.mymobilesite.net
context-based role architecture presented in figure 2.

Figure 4: Communication between Actors in a distance calculation

The mobile device with the role of a service provider would then get its GPS position and offer it to the first mobile device, the client, as the answer to the request. With both GPS positions, it was possible to calculate the distance between the members in the group, and allow further calculations such as areas and volumes.

The reader should notice that this client-server roles differentiation could have been performed equally no matter which phone was the master and which the slave, as both devices are multi-role capable.

In sum, these initial tests showed that a flexible mobile device that is capable to switch roles (during the learning activity, with the activity goals and context in mind) made it possible to create new scenarios where information was shared on demand in a real-time basis.

DISCUSSION AND FUTURE WORK

In this paper, we explore and discuss the notion of adaptive mobile devices in a flexible Learning Activity System, which Flanagan [7] pointed out as a requirement in M-learning. We have presented a new approach, which allows mobile devices to switch from client roles to service provider roles. Additionally, we have concretized and given an example in which GPS data sets were the contents to share by students. We believe that a mobile switching role will contribute to the creation of new scenarios where the learning tool provides the desired flexibility.

In our opinion, an important future research direction for M-learning is on the self-adaptive LAS. Mobile devices, as being important components of such system, should not only provide context information but also take part of this behaviour adaptation in order to achieve the given objectives. As a proof of concept, we have used mobile web servers to provide access of demand to resources located in the mobile device, making web services technologies a stable basis for a scalability and access cross-platform solution.

In our future efforts, we will focus on the expansion of mobile roles, and the realization/creation of definitions on mobile device behaviour such as content repositories, connectivity providers as some examples.

Another focus of attention is the increase of complexity in the mobile switching logic. In our test, the LAS provided the context used to switch the mobile device role. Concretely, we used the activity-dimension of the context. In the future, we require consenting these decisions based on all the dimensions of context, such as environment and personal/interpersonal features [4]. We hope that the results from this study will motivate researchers to consider and include the notion of flexibility in the mobile device behaviour, as being a central component of the LAS.

REFERENCES

Publication III

Mobile Virtual Devices for Collaborative M-Learning

Mobile Virtual Devices for Collaborative M-Learning

Didac Gil de la Iglesia*, Jesper Andersson, Marcelo Milrad
Linnaeus University, Sweden
didac-gil.de.la.iglesia@lnu.se*

Abstract: The increasing use of mobile devices to support collaborative activities creates a need for developing new methods and perspectives to facilitate information sharing. In this paper, we present an approach for information sharing in mobile collaborative settings through the use of Mobile Virtual Devices (MVD). MVD emerges as a new conceptualization of an organization of mobile devices that supports collaborative tasks. The use of MVD allows designers and users to interact with and through mobile devices in novel ways, considering the aggregation of mobile devices as a single entity. The notion of MVD has been conceptualized on the idea of multirole devices, using components to provide and consume resources.

Keywords: mobile device organization, shared resources, collaborative learning

1. Introduction

The increasing use of mobile technologies in our daily life and improved mobile device performance allow for a wide range of solutions that support various learning activities [2]. The increased adoption of mobile learning (M-learning) solutions allows for learning activities to be performed in indoors and outdoors settings in ways that were not possible before. This provides provisioning for design and implementation of learning activities in a wide range of authentic environments that go beyond the traditional classroom. Providing mobile support in these changing settings means that mobile applications must be adaptive to the environment, thus facilitating the flow of information, provision of services and collaboration in new complex situations.

Mobile devices have several limitations in comparison to regular desktop computers. For example, Ally [2] mentions display limitation as a central issue for mobile learning applications and Jones & Marsden [9] point out the unavoidable constraint of the size of the viewable screen. Beside the screen issue, the diversity of mobile devices is large, for instance, regarding available resources (e.g., camera, GPS, connectivity features, etc). Some projects that utilize mobile technologies address this diversity by combining different devices and their resources in order to provide extended functionalities to the users [3]. For instance, Ballard [3] mentions that sharing information can be performed by the use of Pico Nets, Home Servers, and Shared Displays. Sharing data sets belonging to a first device with a second one in order to support mobile collaboration is one possible solution.

In this paper we introduce the concept of MVD and describe the potential benefits of this approach in the field of mobile learning. A MVD may be considered as an alternative approach to mobile sharing, where aggregated devices constitute a MVD, which allows for resource sharing within this aggregation. Our approach may be contrasted to the more traditional approaches where mobile devices, acting as separate units, collaborate and
share resources (described in section II). In section III, we present the MVD approach in more detail, including the main notions behind the concept. Some benefits of using the MVD approach are illustrated in two scenarios in Section IV. Finally we conclude and present future avenues of research in Section V.

2. Related Efforts

Recent efforts in the field of M-Learning have explored potential benefits of information and resource sharing in a variety of settings. Several projects have investigated how data between mobile devices can be exchanged in order to accomplish different types of learning tasks and where the display is shared as a resource as well [1, 3, 4, 11].

One of the goals of the Mobile Notes project is to “support classroom discussions with mobile devices and electronic whiteboards” [5]. Mobile Notes aims to bring technology advances to the classroom by providing PDAs and one Shared Space in a digital whiteboard. Sharing information between students is supported by a centralized database. A centralized solution that allows for asynchronous information exchange is convenient in some scenarios. However, sometimes data is highly volatile, making direct communications in a synchronous manner more suitable [7]. Another effort in this direction is Collpad [1], where a central repository is located in the teacher’s mobile device, and responses from students are sent to this central node using WiFi communications. Even if this approach is closer to a mobile-to-mobile information exchange, it is still dependent on WiFi communications, making it difficult to deploy in outdoors settings. Moreover, resource sharing is limited to student response exchange while support for more complex resource sharing is lacking. Camaleon-RT [4] illustrates how device interface sharing let users interact seamlessly to a multi-display device. Pick-and-Drop is another example of a system that provides functionality to pick and drop objects between devices [11].

The solutions described above are particular for sharing specific types of information and resources and were conceived for environments with unrestricted communication channels. An outdoors-learning environment that employs a GSM network is something completely different [12]. Moreover, our approach should not be limited to one-project goals, such as display or GPS sharing. We envision a solution which is scalable and extendible with respect to future requirements. In the following section, we introduce the MVD and its goal to bring multiple roles to mobile devices [7], in order to address all possible situations for information and resources sharing present in M-learning activities.

3. The Proposed Approach

Many M-Learning projects rely on the use of several mobile devices utilizing their capabilities in terms of connectivity, package of sensors [12] and activity. As each M-Learning activity has different functional requirements, different mobile devices with distinct features may be included in the activity toolset in order to fulfill requirements for a particular experiment. When mobile devices are considered as individual entities, it can be costly to find a device that addresses all requirements specific to a learning activity. With the MVD we advocate another approach, where we consider organizations of mobile devices as single entities that provides the support required by the learning activity. This approach will not only allow for the combination of features provided by resources in one of the mobile devices, but also to combine resources in different mobile devices. MVDs
make it possible to comply with functional requirements through the use of several simpler, cooperating, devices. Moreover, it makes possible to create unique features through the combinations of resources from a MVD, e.g. data manipulations of GPS coordinates generated by the devices that are part of the MVD can be used to perform specific calculations, or quality enhancing features such as improving the accessibility of a device using ambient networking strategies.

Sharing resources between mobile devices presents a number of challenges that need to be addressed; accessibility and communication between devices, organization mechanisms for coordination in the MVD, and mechanisms for service provisioning, discovery and consumption. Moreover, a MVD communication language is required for members in the MVD to define, form, and coordinate according to specific activity requirements. Fig. 1 presents a component diagram for the devices that comprise a MVD. Device resources should be considered as service instantiations in the figure. We describe the characteristics and dependencies of the components present in our approach below.

The components in Fig. 1 provide a minimal set of functionality for resource and service sharing. Fig. 1 specifically illustrates resource sharing through services between two mobile devices in a MVD. All devices in a MVD require the 6 components depicted in Fig. 1 and described below. In one specific scenario, Device 1 retrieves the activity script from a Learning Activity Server (LAS) [7], where activity flows are located. An Activity Interpreter component (C6) analyzes the script and identifies the resources required to perform the activity. The second step to be performed is to identify the location of the required resources, if present in the MVD. Two components are involved in the lookup process; the Service Discovery (C3) and the Service Publication (C1) components. C1 publishes the services available in a mobile device, together with service description and service consumption information. This allows the C3 to identify a proper service required in the activity and to understand how it should be consumed. Consequently, the Service Consumption component (C4) can make use of the services provided by Service Provision (C2). All devices in the MVD must share knowledge of the existing organization, provided by the Orchestration component (C5), defining the role to play for each one of the devices in the MVD. A more detailed, technical, description of the MVD and its components is presented in [8].

Communication between devices in the MVD will address the channel limitations in outdoors settings. A previous solution presented in [7], permits communication over WiFi, GPRS and 3G channels, covering most of the situations for indoors and outdoors settings. In the illustrative case, the activity script is offered by the LAS, but it could also be stored locally in a MVD participant, as an available service.
4. Use Cases

In this section we present two educational scenarios where the notion of MVD has been applied in order to illustrate the potential benefits of this approach. The first case is related to the MULLE (Math edUcation and pLayful LEarning) [6] project, an effort carried out in Stockholm in November 2009 exploring how to support learning with mobile devices in the field of Mathematics. The second instance described in 4.2 is an evolution of case 1.

4.1 MULLE (Sharing GPS and Display resources)

An initial implementation of the proposed MVD approach was tested in the MULLE project with two groups of young learners [6]. From a mobile infrastructure viewpoint, each group was given two mobile devices that formed the MVD that combined two screens and two GPS modules. One of the mobile devices in the MVD communicated with the LAS to receive the activities and to submit the students’ answers. This is presented in Fig. 2 below with the “Perform Learning Task” use case. Fig. 2 also depicts how components from in Fig. 1 are involved. This use case requires from the Activity Interpreter component (C6) to retrieve and interpret the task scripts from a LAS. In this case, two devices formed the MVD. This configuration enables students to calculate distances using two GPS receivers, one per mobile device. Moreover, the fact that the students had two mobile devices allowed for the application to split the visual information along the two mobile phone displays and let the students visualize more information simultaneously. It should be noticed that calculating the distances, as illustrated in Fig. 2, requires discovery of GPS modules (C3), a device to publish and offer the GPS service (C1, C2) and the consumption of this service (C4).

4.2 MULLE v.2 (dynamism in student group creation)

A new use case was added to the preceding MULLE system. To decrease the centralized control of how pupils solve the tasks, they are allowed to create teams on the fly, instead of having them defined prior to the activity. This introduces new dynamics in the composition of students’ groups, therefore devices in the MVD, which requires the identification of the new existing services. In Fig. 2, this is captured in the “Join Group” use case that requires Orchestration (C5) and publication of the offered services to the MVD (C1). Based on the two cases described above, we identify two main benefits in the use of MVDs in mobile collaborative scenarios. (1) Resources can be shared between devices by consuming them through the implemented interfaces. In comparison to previous solutions, MVD can be expanded to allow not only display sharing, but new
resources as well. (2) The possibility of dynamic creation of teams, thanks to the facilities provided by the orchestration and service publication components.

5. Discussion and Future Efforts

In this paper we have introduced the notion of Mobile Virtual Devices (MVD) and its use in the field of M-Learning. A MVD clusters devices and tools in a learning activity into a single entity and promotes the use of their full capabilities in order to offer new ways for user interaction and resource utilization. The implementation of MVD generates a set of challenges, such as connectivity issues, dynamic configurations, and a MVD communication language. Our current results indicate how some of these challenges could be tackled and how scalability from the technological point of view can be achieved by extending the offered services. We believe that our approach opens up a new door to M-Learning activity designers that deals with settings and activities where multiple devices can be inter connected conceptually and share a MVD space, thus creating new ways of interaction and, in turn, facilitate the activity design phase.

The potential benefits identified in the scenarios encourage us to continue exploring how to utilize MVDs in mobile collaborative environments. Future research efforts include analysis and design of more generic cases which requires a deeper understanding of the roles of mobile devices in relation to the learning activity system, the learners, and the activities along other components in the software ecosystem [10].

References

Publication IV

Enhancing Mobile Learning Activities by the Use of Mobile Virtual Devices - Some Design and Implementation Issues

Abstract—The use of multiple mobile devices is increasing in mobile learning, bringing a need for collaboration and resource sharing among participating pupils. This paper presents an approach that addresses information and resource sharing for mobile devices in indoors and outdoors settings. Our solution consists of aggregated mobile devices, forming organizations. These Mobile Virtual Devices (MVDs) provide a new mechanism that facilitates design of mobile learning activities offering a virtual complex device that combines the features of several mobile devices.

Keywords-component; mobile organization, shared resources, collaborative learning

I. INTRODUCTION

Adoption of modern technology in education suffers from problems related to the increasing difficulty to stay in pace with technology developments. This triggers costly adaptations and improvements of existing learning activities to make use of the new and improved features that modern mobile devices offer [2]. One advantage with mobile devices in education is that teachers may situate learning activities in real settings. The mobile learning field (M-Learning) creates learning scenarios in indoors and outdoors settings, bringing the class to authentic environments. A second benefit from using mobile technologies is related to the large set of resources these devices offer, “Mobile devices are the most common example of ubiquitous clients and the tendency is for them to become as a pack of sensor device rather than mobile phones” [15]. In real settings it is key to facilitate transparent and ubiquitous access to information, where the learning activities take place, and where the activity results are created. Information created as a result of learning activities is known as Emerging Learning Objects (ELOs). ELOs are one of the most important additions that M-learning brings to the education arena. Thus, a requirement for most M-Learning projects is sharing information between involved pupils and to maintain and drive an activity flow, and foster collaboration between the pupils and teachers participating in the activity.

Even if mobile devices offer several improvements and new possibilities in comparison to desktop solutions, it is also a fact that they lack important features found in traditional desktop computers. One example of this is the limited resolutions that the mobile devices screens offer [2][12]. Smaller displays restrict the amount of information that can be presented to pupils and the interaction types that pupils may engage in with the device. As a consequence, application design for mobile devices differs from desktop application design [5].

The specific features offered by specific mobile devices in projects also influence activity design. There is a large diversity of devices with different resources. Therefore it is common practice to include a variety of devices that together provide the features needed in a learning activity. Mobile devices with different resources can create different ELOs. This again, demonstrates the need for sharing between devices in order to combine the ELOs created in different devices. Another perspective on sharing between devices is resource sharing, where the resources required for ELO creation are shared and one device may utilize a resource available in another.

We propose Mobile Virtual Devices (MVDs) as a principal mechanism for sharing information and resources for ELO creation. A MVD is an organization created by aggregation of several mobile devices as into one logical entity, which share resources. Resource sharing inside an MVD can be used to support new ways of interactions in M-Learning activities.

The use of MVD provides several benefits for M-Learning activity designers, as it opens up for new ways of device-to-device interaction and thus, pupil to pupil, pupil to teacher, or teacher to teacher collaboration.

The main contribution of this paper is the discussion on architecture and mechanisms for the design and deployment of the MVD and services it provides.

The paper is organized as follows: section II presents work in the M-learning field concerning object sharing between multiple devices. In sections III we summarize some of the critical concerns identified in the M-Learning field. Section IV, discusses and describes our approach for the MVD, including its architecture, the required components and their interfaces. In Section V, we discuss some of the benefit we experienced from using MVDs in M-Learning settings. Section VI presents a scenario where the MVD approach was used to illustrate how the MVD
components were implemented. Section VII concludes the paper and suggests future work.

II. RELATED WORK
In the introduction we discussed information sharing for mobile devices in the M-Learning field. This issue has been addressed in several previous projects, including the human-computer interaction perspective to address the issue of mobile devices’ screen sizes [1][5][6][14][15].

Mobile Notes [3] is one example where technology is used in school settings to enhance pupils learning experience. The project uses PDA devices for the pupils and a digital whiteboard where a Shared Space is offered to the pupils for interaction and data exchange. Information in the project is shared through the use of a central database. This solution provides an asynchronous mechanism for information sharing. A centralized asynchronous solution is useful for exchanging information in several scenarios. Examples of centralized solutions, with the goal of sharing information include [1][3][5]. Here information is sent to a central node, and mobile clients poll the server periodically in order to detect new information to process or new tasks to perform. Centralized approaches may also be server based, for instance utilizing cloud-computing technologies, using them as a storage point, where information, pictures and data are stored. These solutions have a downside, that information is not shared immediately. Information is shared first after the creator has uploaded the object to the server and after the pull action is performed.

However, there are situations when information is required at the very moment it is created, in a synchronous manner. In some scenarios, when it is important to work with up to date data, ELOs may be created on demand by requests. In such scenarios a direct communication between devices is more suitable [11].

Collpad [1] is a project that makes use of mobile-to-mobile communication in an indoors-learning environment. The project uses PDAs for teachers and pupils. One pinpointed device is used as storage repository. In addition, it relies on WiFi connections for its communication channels, something that makes it difficult to deploy in an outdoor setting.

Another example in this direction is the Cloud to Device [4]. Its API allows pushing notifications to mobile devices in order to request the device to perform a task, such as loading the web browser to open a specified web page. However, this approach does not permit sending messages from the device back to the originating source. For example, this solution does not allow that an application requests a mobile device to submit a picture, as the technology is not prepared to include the picture file in a response.

From a resource sharing perspective, we can find projects with the interface distribution as a goal. This should be seen as a specific type of share of resources between devices that are working in the same activity. One case is the Camalcon-RT solution, from Couraz [5]. In their system, a user has the possibility to combine several device interfaces and to interact seamlessly with one multi-display device. The solution proposed is a layered architecture that relies on LAN as the communication channel.

Another effort in this direction is the system described by Rekimoto [14]. They describe a system users may pick up and drop objects between displays, like if they were physical objects. In Pick-and-Drop, objects are transferred between devices making use of digital pens, which are tracked to move objects. The communications are provided through a network layer, and a central system requires that the actions performed by the digital pens are recorded.

All these solutions were deployed in scenarios were network communications were not an issue. However, in outdoors settings, the communications are normally limited to the GSM network or 3G. In these settings, establishing communications when a mobile device is the destination has some limitations, as stated in [15].

The second factor we want to address here is the scalability issue. The projects discussed so far were single goal projects, such as display sharing [5][14] and exchanging text answers[1]. PaCom [15] allows exchange of information between devices by the combination of services they offer individually and facilitates “ad-hoc combinations of services as well as adaptations to changes in services” [15]. This solution provides a mechanism that may be adapted to M-Learning objectives, by combining resources that the user can find in pervasive environments. Our approach is aligned with this solution, while different in some aspects. We envision a solution that treats resources located in several mobile devices as if they were in one single device that brings multiple roles to mobile devices [11], offering a mechanism for service composition and sharing.

III. IMPLEMENTATION CONCERNS
The analysis of M-Learning projects presented above and our own experience from several activities generated a set of requirements for mobile device frameworks aimed at supporting M-Learning activities. We summarize these concerns below.

A. Single Device Resource Shortage
When considering mobile devices in isolation, activity designers may only make use of the features that specific devices can provide. This limitation forces designers to include complex devices that fulfill all resources required by the activities’ functional requirements. From another point of view, certain M-learning scenarios will not be feasible, since it will be close to impossible to find a single device that satisfies all resources requirements. One example is the difficulty to offer an activity where area calculation is the learning focus. No mobile device on the market provides functionality that can pick up 4 coordinates that constitutes the corners of a trapezoid figure. Most devices are GPS-enabled, but limited to acquire one coordinate at the time. However, this scenario would be possible if there exist one mobile device with 4 GPS modules.

The first concern is to provide a mechanism to consider a group of mobile devices as a single instance. This new
instance will be considered as a distributed device with multiple resources. Direct advantages are not only the possibility to simplify deployment of complex learning scenarios creation, also enabling new scenario types with fully equipped cooperating devices.

B. Accessibility and Communication

Mobile device collaboration requires a communication channel. In outdoors settings, GSM networks may be the only choice available, but these networks come with some accessibility limitations [17]. For security reasons, GSM providers do not allow communications not initiated on the mobile device. As a consequence, it is not possible to access resources located on a mobile device, unless it is the mobile device that initiates the transfer. This would not make possible to download a picture from a mobile device when desired, on demand. It would be necessary to wait until the mobile device starts the delivery. It could be partially addressed by a central messaging system, where requests could be stored and devices could poll and retrieve them periodically. Still this approach does not allow for synchronous communication.

It is an absolute requirement to provide a communication channel that is reliable in most of the environments where M-Learning activities can be performed, including outdoors environments using GSM networks.

Resolving this concern enables direct synchronous communications in multiple settings, both indoors and outdoors, and provides the required device-to-device connection for grouping multiple devices.

C. Common Activity Description Language

Activity scripts define the events that pupils perform in the activity, the flow and pupil roles, and other activity aspects. There is no standardized activity script definition language available. Most available languages are domain specific. This can be an issue in a mobile collaborative environment, not reaching an understanding between devices as a consequence of not sharing the same language. It can be the case that one device requires results from a service available in another phone, but does not know how it should be requested.

It is important that a common activity description language is shared along all the devices in the system, supporting unambiguous exchange of information and services.

D. External Interfaces

Some mobile solutions have been created with one single functional goal in mind, and have not taken future extensions into consideration. As a consequence, such solutions have not provided a mechanism for interaction and consumption of the services, such as an API. This often leads to that successful solutions generated in one project cannot be ported, combined, or reused in another project.

A complete mobile framework must provide for interaction with external actors that its services may use and thereby extend themselves.

The success of framework APIs, fostering reuse and flexibility M-learning frameworks provide external interfaces to allow external parties to extend and tailor its behaviours.

E. Mobility and Flexibility

Finally, M-Learning projects have in their initial stage, in the design phase, a closed definition of the possible collaborations what will be performed during the activity. Groups of pupils are predefined and cannot be modified during the execution, following the indoors-structured setting. However, the dynamism present in outdoors settings tends to be much higher than that. The mobility of devices introduces an openness in the field, it is possible that one device in a working group becomes inaccessible, maybe because of the low connectivity in the area, a low battery level, or because the pupil has moved too far from the group.

Mobility and flexibility must be better addressed in M-Learning projects. It must consider the dynamism and low reliability in the accessibility of the mobile devices. Moreover, considering them, it is possible to increase the potential of the mobile device aggregation, as their features can be increased dynamically on the field by adding extra devices. In addition, resolving this concern, pupils will be offered with more flexibility in their learning activities, letting them decide their own problem-solving approach.

IV. The Selected Approach

Based on the aforementioned requirements, we derived a framework architecture, which supports activity designers in scenario creation and pupils designing their individual problem solution processes.

When mobile devices are used as support in M-learning activities, designers and developers will assign all devices a role to perform in that particular activity. The role concept is flexible and simplifies the aggregation and use of features offered internally in the activity, and not to ignore features and capabilities from any of the devices. Moreover, considering the mobile devices as single entities limits the ways they can interact, which usually leads to limited intra-device resource utilization. Our proposed solution combines resources in multiple devices to create a composite instance, the MVD.

Figure 1. Mobile Virtual Devices working in collaboration under the activity provided by a LAS
A MVD instance is a composite of multiple mobile devices. The composite architecture enables resource sharing in between MVD members, which extends and enhances services that the mobile device can offer. The MVD increases its service offerings by combining a larger number of resources. Moreover, in the Fig. 1, we illustrate a scenario with four mobile devices forming two interacting MVDs that communicate with the external LAS.

In the Figure, we also depict the relations to the concerns enumerated in section III. For example, we may identify two devices in the first MVD. These devices share resources using a wireless channel a common language. The MVD organization interacts with actors external to itself; the Learning Activity Server, where learning activity information is stored, and the second MVD (MVD2), to which it offers its services.

Moreover, it also illustrates the possibility of mobile devices to leave a MVD organization to join another MVD (E). Such changes will obviously alter the set of services offered by the MVDs.

Assembling the MVDs require that the participating members can share resources, manage intra-MVD communication, govern collaborations, and allow for dynamism within the organization. Below, we describe some prerequisites for and how to compose MVDs, i.e., which components are required in a device that is a MVD prospect member.

A. Service Provision and Consumption

The main purpose of a MVD is resource sharing. Sharing resources between two or more mobile devices means that for each resource at least one device provides the resource and at least one consumes the resource. On the surface, this is a straightforward, simple, procedure but in practice several factors must be considered. In each MVD participant, two components are instantiated that together are responsible for resource sharing. One component provides an access interface to the participant’s resources, while the second component consumes required resources available in other participants. It is important to underline that this provision-consumption pairing can be done locally in one device or between two devices in the same MVD.

This pairing is illustrated in the Fig. 2 by two interfaces. The provided interfaces are implemented by the Service Provision component (2). The rationale for this interface is to provide mechanisms for accessing resources located in a mobile device. The second interface, located in the Service Consumption component (4), provides access to external resources located in other MVD participants. This interface is the mirror image of the provider interface it connects to. The connection of two devices is illustrated in Figure 2 where Mobile Device 1 uses (4) to consume a service provided by the Mobile Device 2 in (2).

A first observation is that the participant resource pairing requires common language. This language must provide sufficient information about how resources may be consumed, syntactically and semantically, which will constitute a contract between the two parties. One approach to address this is to derive an ontology that defines service characteristics, descriptions, parameters and relevant values for the service consumption and shares this with MVD participants.

The second observation is concerned with how to set up sufficiently effective communication channels in outdoors settings, which is an absolute requirement for the MVD. In our projects, we have worked with and evaluated several approaches. One approach that we found promising is the Mobile Web Server (MWS, www.mymobilesite.net), a Nokia Development Team project, allows access to mobile devices over the Internet using the HTTP protocol. The web server solution ported to mobile devices allows ubiquitous access to cellular phones. MWS may provide access over several wireless communications channels, including GPRS, 3G and WLAN [15]. In FLAC [11], a precursor to the MVD project, the mobile-to-mobile direct communication was used to implement the service provider components. The MWS approach caters for the accessibility problem presented in III.B [17]. In FLAC, the provisioning component is placed in a Web-Server container. The provided interface in FLAC is a REST API accessed by HTTP calls. This simplifies the implementation of applications that access mobile device content over the Internet.

Fig. 3 below shows a deployment diagram where FLAC is used to allow resource access from one mobile device to
another. Here the provided interfaces in FLAC allow mobile phones to share screens, access media objects in the repository and share GPS sensor data.

B. Service Discovery and Publication

In a dynamic environment, resource sharing requires a mechanism to detect and track where resources are located. Some resource types, for instance a Bluetooth module, may possibly be found in none, one, or several (all) devices in a MVD. Components for service discovery and publication mechanisms assist service identification, i.e., finding services that provide resources needed for specific learning activities. The Service Publication (1) component provides an interface for publishing sharable MVD resource services offered in the mobile device. The Service Discovery (3) component closes the loop. The Service Discovery component requires assistance provided by Service Publication interfaces for locating required services.

In section IV.A, we identified the common language required for device pairing. We discussed previously how such a language was an absolute requirement for sharing services. The services discussed here also require that and may use the ontology structure we proposed above. The ontology is also provided by Service Publisher interface, visible to Service Discovery components. Note that service publisher and provider components are linked internally. This linkage allows publishers to identify existing services in the device, how these services may be consumed and information similar to a SLA (Service Level Agreement, which permits for the selection of the best service with respect to the current state and environment. This service information is also shared to service consumer components via the service discovery interface.

C. Orchestration

A MVD is a complex decentralized system with a high degree of dynamism. Therefore, service availability will vary over time due to a constantly changing environment. Managing this dynamic situation is the responsibility of an Orchestration (5) component. Its goal is to keep the MVD organization active during the learning activity. An orchestration component will decide the composite role of each device in the MVD based on the activities’ goals. The MVD realizes its orchestration behavior with two interface types (5). The first interface allows devices in the MVD to coordinate and organize each device’s roles. This interface type connects Orchestration components from all devices in a MVD. The second interface requires service consumption components to perform tasks required by activities. These invocations are controlled by an Activity Interpreter component that interfaces with the orchestration. This interface is detailed below.

D. Activity Interpreter

The Activity Interpreter is responsible for activity governance, i.e., how the organization of the mobile devices needs to fulfill the activity goals. The objective of this component is to retrieve a description of the activity and orchestrate how the participating devices will behave. Therefore, it requires an interface where an activity script can be retrieved. The Activity Interpreter, as the name suggests, interprets the content of the script, and will identify possible methods to achieve the activity goals. In the scope of this article, MVDs are presented to be used in M-Learning scenarios, although its use can be extrapolated to a broader area. Therefore, in the Fig. 2, the Learning Activity Server (LAS) is the external actor that offers the MVD with activities to perform. Both the LAS and the MVD require a shared ontology in order to share an understanding of how to perform the activities. An Activity Interpreter module receives the task description and infers the services required.

E. Service Extension

The scalability of the MVD concept must be addressed. The arrival of new features in a mobile device can lead to the creation of new services, offered and consumed. A GPS module, an RFID reader and a video projector may be shared in the MVD by creating required interfaces that enable their consumption, setting up an ontology that defines how to consume them, and publish the services in offerings.

In addition, we have to consider other changes than technological improvements that may result in the creation of new services. ELOs can be dressed up as and presented as new service offerings; videos recorded by the pupils, pictures, and the answers for the learning tasks can be shared.
implementing the interfaces to extend the list of services inside the MVD.

V. BENEFITS OF THE APPROACH

One of the overall goals for the M-learning community is to for a more flexible and less restricted learning process, both for pupils and teachers. However, most efforts have focused only on the pupil. Currently pupils may enhance some aspects of lectures in outdoors settings, while teachers have found their responsibilities increase. Now they need to know how to design, create, and deploy outdoors learning activities that utilize mobile technologies. This implies that they need to able to manage tasks that normally are the responsibility of software designers and developers, for instance, understanding how to use the mobile technology, learning programming languages to implement applications for the mobile devices, planning deployment on each of the mobile devices in the activity, etc. With our approach, using MVDs, we may automate a majority of these tasks, thus teachers are able to focus on their main concern, designing an activity that covers the learning outcome goals.

For pupils, we may provide a less constrained learning environment. For instance, they can decide to have lectures outdoors where their role is more active than in regular ex-cathedra teaching. However, pupils are still tied to roles specified and assigned to them by the M-learning activity. The second benefit from the use of MVDs is a possibility for pupils to setup their own working groups for the learning activities. The MVD concept allows them to create their own MVD organizations and to combine technologies available in their group to perform the required tasks.

Something we believe is the main benefit from using MVDs is how multiple MVD organizations simplify the process of achieving functional goals, utilizing hard and soft resources more efficiently, and finally us the inherent dynamism as a pedagogical tool. Examples of MVDs as pedagogical tools are GPS coordinates provided by the devices in the MVD can be used to perform distance calculations and coordination to take simultaneous pictures for an event from different points of view, to allow 3D models. In another example, which also uses distance calculations, new services can be created to locate the barycenter in a triangle created by three mobile devices positions.

VI. MVD IMPLEMENTATION

In the autumn 2009, a project in the field of Technology Enhanced Learning (TEL) was carried out in Stockholm [10]. MULLE, Math education and pLayful LEarning, attempted to provide an outdoors situated-learning activity to enhance the understanding of geometry measurements, more concretely, rectangle distance and area values. Design researchers, developers and pedagogic experts were involved in the design of Mulle. The project aimed to use mobile technologies to guide the pupils within the activities and to provide them with mechanisms to be able to realize distance measurements; “approaching mobile devices as tools means that they no only support the flow of the learning activity, but also can function as tools within the activity” [10].

Experience from previous M-learning projects told us that learners pay excessive attention to the mobile devices while “encouraging face-to-face collaboration and communication makes the learning activity less dependent on the affordances of the mobile devices” [10]. To fulfill this requirement, the designer team envisioned the use of more than one mobile device to bring activity information to pupils. With this approach, pupils would be forced to focus on multiple devices and devices would be considered part of the tools used in the activity, and not the focus of attention.

From the discussions in the design team in the MULLE trial, the following set of technical requirements were identified:

- Share locations between devices
- Access to content created in real-time
- Distribution of activity information between the devices used inside a group

The MVD could provide the features for distance calculations that were a functional requirement for the activity, while providing for dissemination of information on several screens by sharing this resource. From the technological viewpoint, each group was provided with two mobile devices. Those formed the group’s MVD, combining two screens to provide the information and two GPS modules to calculate distances.

A primary device was in charge of the orchestration inside the MVD, deploying the activity control and flow. The tasks performed by the MVD can be described as: (1) the Activity Interpreter requests task information (Scripting Language-based activity file) from the learning activity system; (2) presents the questions regarding the the task; and, (3) send the pupils’ answers back to the LAS for checking and for retrieving information for subsequent tasks. The primary device, in its manager, orchestrated the collaboration with the second device. In this particular case, the Orchestration component requested from the primary device to communicate with the LAS and distribute tasks performed by each one mobile device in the MVD. In addition, the Orchestration component requested the secondary device to display the video hints for the current task, while the remaining information (questions, text, and feedback) were displayed on the primary device. In the Fig. 4 we can observe the composition of two devices inside the MVD. The device on the left side, the manager device, orchestrates the tasks to be done in each one of the devices, and requests resources from the second device (right side) when needed, in order to get the second GPS value to calculate the distance, or to request the second device to offer a video to provide more information to the users about how to solve the questions for the current task in the activity.
As outlined in section IV, communication between devices in the MVD could be handled through “client to mobile-web-server” communication over the Internet, using 3G connections (Fig. 3). The Web container allows for the Service Provision and Consumption to pair the services. Using this flexible pairing, the second device identified videos pupils could request for each task. Inter-device communication was also required for the distance calculations in the task (Fig. 5).

Using MVDs in the learning activity provides for resource sharing without involving a central node. This covers for the on-demand location-sharing requirement identified in the design process.

Our final requirement was concerned with service extension and dynamism. In this system, Web Services on mobile devices provided an acceptable environment for the envisioned requirements, where a device can adapt to offers even they are highly disparate. One such example was display sharing in the MVS to allow an interface distribution.

Results from the experiment showed a satisfactory reliability on the 3G communications between devices. On the other hand, a delay of up to 6 seconds in the communications between mobile devices (Service Provision and Service Consumption components) was registered. This delay could be explained by latency in 3G communications and performance issues to dispatch web service requests and display results on mobile devices. Fortunately, this delay was not affecting the flow of the activity. Pupils did not check task hints before they finished reading the task description.

**VII. DISCUSSION AND FUTURE WORK**

In this paper we have presented an approach for a design problem recurring in many M-Learning applications. The problem appears when several mobile devices are used to support learning activities. The problem has its origin in designers underestimating the actual need or design suboptimal use of devices’ features. Designers should develop systems that maximize end user experience but when some of these resources, such as screens, are not used to their full potential, or in the worst case, not at all, it affects end user experience and overall outcome of the learning activity.

We have also exposed other recurring problems in M-learning outdoors activities. For instance, pupils are still requested to perform the activities in the roles devices assign to them. Instead, we believe that mobile devices should adapt to end user behaviour, and not the other way around.

A third problem category we have discussed is how the new technologies and trends impose new responsibilities for teachers. Besides the more traditional teacher role, they must now be able to design, implement, and execute mobile learning activities, often with little or no support from tools or prior formal training.

To address these problems, we introduced the MVD concept. The concept considers the aggregation of all devices and tools in a learning activity instance as one single entity. This promotes the use of the devices’ full capabilities for the benefit of end user behaviour and interactions. Moreover, it provides for combining device capabilities inside the MVD with the objective of creating new more complex services, services that could not be provided by the devices individually.

The concept of MVD introduces a set of challenges that need to be addressed, such as connectivity, a common communication language, orchestration, and service provision. Our approach presents how these challenges could
be managed and allows scalability from the technological point of view to extend the service offerings of an MVD. MVDs provide a new dimension for M-Learning activity designers where multiple devices can be conceptually interconnected and share a MVD space. This creates possibilities for new ways of user interaction and facilitates the activity design phase. However, MVDs are not limited to the M-Learning field can be used in other fields where mobile technologies are also in use.

The Mulle experiment, where MVDs were used in field trials, demonstrated promising results. Both from the technical point of view and the learning activity design perspective.

For future work we have identified a number of issues we will address. Current efforts are being invested on the migration of the MWS on Nokia devices to other platforms, switching the implementation to a JADE solution [7]. Java Agents will facilitate the creation of Orchestration mechanism, Service Discovery and Publication and the framework provides a robust communication layer for mobile and desktop devices. Moreover, initial tests have suggested improvements in the communication latency by the use of JADE in comparison to MWS.

Additional efforts related to MVDs include the analysis of a more holistic view with external actors and systems. Mobile devices, and MVD by default, are actors inside a learning software ecosystem. A better understanding of these ecosystems is a prerequisite for successful development and deployment of mobile learning activities [13]. The common language must improve together with the MVD’s components. A new version should allow for tasks to be divided and shared between devices in an MVD. At the moment, the language has been used for small toy activities and more complex structures must be explored.

Finally, the dynamism in the creation of MVD should be addressed in more detail. Allowing mobile devices to become members in the organization, to leave the organization and to manage the tasks performed by each member is key for the success of the MVD concept. In a distributed organization, such as the MVDs, when a member leaves the organization, resources disappear, then tasks need to be reassigned and new configuration need to be derived on the fly.

References


Towards a Decentralized and Self-Adaptive System for M-Learning Applications

Towards a Decentralized and Self-Adaptive System for M-Learning Applications

Didac Gil, Jesper Andersson, Marcelo Milrad, Håkan Sollervall
CeLeKT (Center for Learning and Knowledge Technologies)
Linnaeus University, Växjö, Sweden
{didac.gil-de-la-iglesia, jesper.andersson, marcelo.milrad, hakan.sollervall}@lnu.se

Abstract: Through the analysis of the different iterations of the Geometry Mobile (GEM) project, a mobile learning effort in the field of mathematics, we have identified a major architectural issue to be addressed in the design and implementation of m-learning applications. Due to the dynamic nature of the field many challenging requirements are continuously emerging. One of them relates to the possibility to support collaborative activities that demand sharing resources between students and their mobile devices in constantly changing conditions. These situations generate the need of using decentralized distributed architectures in which mobile devices can share resources to carry out the activity covering the concerns defined by the different stakeholders. This paper describes our current efforts connected to identifying a set of requirements for M-Learning activities. Thereafter, we elaborate on why a decentralized distributed system (DDS) can be used to provide a novel solution to tackle the mentioned above problems. Moreover, initial aspects related to the design of a DDS, including a self-adaptation mechanism are presented.

Keywords: mobile learning, self-adaptation, decentralized distributed system

I. INTRODUCTION

In education, Information and Communication Technologies (ICT) have been adopted to provide new teaching and learning approaches that can be integrated in traditional classroom settings. One of the trends is the introduction of mobile devices in school activities outside the boundaries of the classroom, known as M-learning. However, there are several limitations regarding the use of mobile devices in schools in comparison with desktop computers [5]. Limitations in mobile devices range from performance capabilities (connectivity, power, local storage) to display-size constrain, just to enumerate some of them. The combination of devices is a possible solution to tackle some of these problems to eliminate the restrictions that each one of these devices poses, thus creating a mobile distributed architecture.

In this paper, we will present and discuss a set of requirements we have identified as a result of implementing a series of m-learning activities during the last two years. From these requirements, we will present the limitations of different software architectures to finalize in an architecture candidate that covers most of them.

The paper is structured as follows; section II introduces the different iterations of our mobile learning project called Geometry Mobile (GEM), an M-learning project in the field of mathematics [5,13]. The analysis conducted during the different phases of the project served to initiate a collection of software requirements that an m-learning application should address. The list of the identified requirements ends the section II. From these requirements, in section III we describe different architecture alternatives to identify the benefits and drawbacks that they present, to finally introduce the proposed candidate architecture for addressing the complex problems that emerge in connection to these M-learning activities. Section IV describes high-level characteristics this candidate must have to cover the requirements identified in section II. Finally, section V summarizes the conclusions of the paper.

II. IDENTIFICATION OF REQUIREMENTS

The requirements for the proposed architecture presented in this paper are the result of an evolutionary process from the lessons learnt during our trials and from a comparison with related efforts done on the field of M-learning. However, these areas of attention are not
necessary limited to the field of M-learning, but they can be extended to other domains where the notions of distribution and constantly changing settings are central.

Many M-learning projects involve the use of mobile technologies for outdoors settings. Some complex M-learning activities include the use of multiple mobile devices in the field in order to overcome the limitations of each separate device. Geometry Mobile belongs to this category. In the fall 2009, the first version of GEM used multiple mobile devices to work together in order to support geometry-oriented learning activities. Mobile devices were used to calculate distances and to measure geometrical dimensions including diameter, side length, etc. The distances were calculated by combining two or more GPS coordinates on the field [5]. A solution for this scenario demands devices to (1) share their location data from the GPS resources. This sharing requirement was soon complemented in the second iteration of GEM (GEM2) to (2) share the displays along the devices used in each group, to increase the amount of information that could be synchronous presented to the users in the different groups.

Generalizing, it is understood that mobile devices in a group should share not only the data they collect, but also the resources they hold.

To facilitate the share of resources, these can be presented as (3) services to share. This opens the possibility to create service composition from resources that are not placed in one device, but distributed along interconnected devices across the platform. In our particular case, we can offer a distance service that consists of the combination of several location services. Offering resource composition brings the possibility to increase existing capabilities for M-learning activities.

One step forward is to consider the association of devices as one single entity. This (4) device composition facilitates the design of M-learning activities, making the real implementation of the device capabilities transparent to the designer. However, to bring this transparency, it is necessary to provide the system with an (5) organizations management feature.

One organization implication is that devices require to provide (6) several behaviors depending on their role, a service consumer at some point, a service provider at other.

In the GEM3 iteration, we kept the focus on geometry but adapted the activities towards space orientation and geometrical shape constructions [13]. The design of the GEM3 activity was prepared to deal with three groups working one after each other. The hard weather conditions in this iteration affected the speed of the students, and demanded to rethink on the activity, to run it instead with the groups in parallel. However, the system was not designed to change the activity flow at runtime. In indoors activities activity flow changes can be reasonably done, but this becomes a challenge in outdoors M-learning activities. This brings the following requirement; the system should allow to some extent (7) changes in the activity flow during execution time.

It is known that the dynamism and flexibility of the environment is a decisive characteristic of outdoors activities. If environmental conditions are determinant factors in the activity, these will influence the devices behavior, and the related device organizations. The status of the hardware is dynamic as well. For example, the battery life of the mobile devices or its connectivity poses some limitation on the device availability. The potential issues arising as a consequence from these changing conditions need to be considered as well.

For research purposes, the software solution must log in the mobile devices relevant events. The information registered in these logs is key for evaluation of the system from the developer perspective, but also relevant for teachers to be able to carry out a personal follow-up on the students’ actions both individually and in groups. Due to its relevance, (8) log availability needs to be assured in spite of dynamic conditions issues.

Besides the mobile devices, carrying out the outdoors activity, there are other components involved in the design of the GEM experiments. One is the Activity Server [5], that contains the activity script and keeps track of the activity flow and students’ outcomes. But what would happen for instance: if the server became unavailable during the activity? The (9) availability of the activity script becomes a requirement for M-learning activities, so the activity can proceed.
Depending on the concerns to fulfill, different organizations may be demanded. A conclusion from these requirements is that (10) several organizations are required in different layers under different criteria, as illustrated in Figure 1.

![Layered view of devices organized based on different concern criteria.](image)

It should be noticed that this assumes that devices could belong to different organizations concurrently. Even though, organizations in one layer can impose implications that would affect the existence of organizations in other layers. Therefore, a requirement for organizations is to analyze and evaluate the (11) coexistence of concerns to create organizations. To finalize, we want to stress the need for a (12) system that is resilient to issues affecting availability. Due to the dynamism in the environment and system components, devices may become unavailable, and not satisfy the organization requirements. This dynamism forces a constant evaluation of the organizations, to assure that the requirements keep covered. The table below summarizes the requirements identified in the GEM iterations; however, they can be applied to other fields with distributed and dynamic environments.

<table>
<thead>
<tr>
<th>ID</th>
<th>Specific Req.</th>
<th>Req. Category</th>
<th>Identified in</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Location Shar.</td>
<td>Share Resources</td>
<td>GEM1</td>
</tr>
<tr>
<td>2</td>
<td>Display Sharing</td>
<td>Share Resources</td>
<td>GEM2</td>
</tr>
<tr>
<td>3</td>
<td>Service Shar.</td>
<td>Share Services</td>
<td>GEM2</td>
</tr>
<tr>
<td>4</td>
<td>Device Composition</td>
<td>Mobile Virtual Dev.</td>
<td>GEM2</td>
</tr>
<tr>
<td>5</td>
<td>Organization</td>
<td>Organization Comp.</td>
<td>GEM2</td>
</tr>
<tr>
<td>6</td>
<td>Dev. Behavior</td>
<td>Organization Comp.</td>
<td>GEM3</td>
</tr>
<tr>
<td>7</td>
<td>Dynamic activity flow</td>
<td>Self-adaptation</td>
<td>GEM3</td>
</tr>
<tr>
<td>8</td>
<td>Log availability</td>
<td>QoS (Availability)</td>
<td>GEM3</td>
</tr>
<tr>
<td>9</td>
<td>Activity Script availa.</td>
<td>QoS (Availability)</td>
<td>GEM3</td>
</tr>
<tr>
<td>10</td>
<td>Multi-layer organization</td>
<td>Organization Comp.</td>
<td>GEM3</td>
</tr>
<tr>
<td>11</td>
<td>Coexistence of concerns</td>
<td>Organization Comp.</td>
<td>GEM3</td>
</tr>
<tr>
<td>12</td>
<td>Resilience to changes</td>
<td>Self-adaptation</td>
<td>GEM3</td>
</tr>
</tbody>
</table>

Table 1. Emerging requirements during the GEM iterations

III. Architectures in Mobile Learning

Our scenarios present a set of resources that are distributed across different devices and need to be shared. This concept is one of the foundations of distributed systems. "The sharing of resources is a main motivation for constructing distributed systems" [3]. Some examples of distributed systems are [4,12], which interconnect desktop devices and servers to provide QoS, performance, availability, etc. making use of the CORBA or RMI standards. Distributed systems are extending beyond the traditional forms and expanding also to the mobile field [9,10,11]. The combination of distributed systems and mobile devices allows the latest to access more efficient computational power and other resources placed on servers. The use of distributed systems to support learning is aligned with one of the shifts mentioned by Goodyear [7], claiming that one of the major challenges for educational research communities consists of addressing the spread of learning activities on several distributed contexts.

Before analyzing the benefits and drawbacks of different distributed architectures, it is relevant mentioning that due to the characteristics of centralized systems (CS), these become not suitable for collaborative activities, as they do not support the interaction among multiple peers. The main issues here are that data is generated on one device and required on another device (req. 1,2,3), collaboration between students is required and the mobile device limitations need to be overcome.

A. Centralized Distributed Systems

The use of centralized distributed systems (CDS) has enabled the possibility of distributed education [2,8,10]. Cogburn [2] conducted a study on the status of the Globally Distributed Collaborative Learning during the last 10 years to identify an appropriate technology integration approach for supporting different kind of learning activities. He determined that a collection of web-based tools would cover the need for supporting distributed collaborative learning, such as Wikis and Chats [8]. This solution covers the requirements identified as (1,2,3,8,9). Considering the Internet as a central node for communications, requirements (4,5,6,10,11) cannot be considered on the device side and need to be implemented on
the server side. This leads to severe limitations for some M-learning activities where peer-2-peer approaches would be needed. Concerning the system being adaptive to changes (7,12), these systems have not paid enough of attention. Studies like the one carried out by [1] consider the use of a client-server architecture. In these, the server will contain mechanisms to provide adaptation to the students, based on their knowledge, actions performed and the device in use; but those adaptations are always towards content adaptation. However, no adaptation has been considered to address issues happening on the field. Moreover, having a server as a central point of connection introduces one level of bottleneck [10].

B. Decentralized Distributed Systems

The decentralized distributed system (DDS) or fully decentralized system addresses this issue of existing bottlenecks [10].

To overcome the limitation of bottlenecks in a DDS, the services that compose the system are distributed or replicated across multiple locations, facilitating the scalability of these. Therefore, by replication a DDS can reduce the risk of failures. In case a service is not working properly, a mechanism should be on place to substitute it and maintain the full system working correctly [15].

The current use of cloud computing techniques is one case of DDS. The cloud can provide mobile devices with the learning resources needed to carry out the activity. In DDS, the mobile device can play a more active role, enabling a peer-2-peer communication between mobile devices. This would allow the req. (4,5,6) by constructing organizations with multiple mobile devices. Yet, the mobile device still remains as an entity being susceptible to failures and replication of devices does not present as a suitable solution.

C. DDS + Self-Adaptation Mechanisms

The dynamism of the environment can provoke failures to the devices and the overall system. Adding a self-adaptation mechanism can be an approach to address these failures. As the authors mention in [14], “Most distributed applications are brittle; they work in a limited environment and cannot adapt to changes in this environment”. The authors add a self-adaptive mechanism to their system to address levels of issues that are classified into predefined QoS regions, defining behaviors to correct each of the set regions. This solution addresses req #12. However, the self-exclusiveness of the regions restrains the possibility of the coexistence of concerns (11), as presented in the Figure 1. Table 2 below describes the deficiencies in connection to each of the different architectures described in this section.

<table>
<thead>
<tr>
<th>ID</th>
<th>Specific Req.</th>
<th>CS</th>
<th>CDS</th>
<th>DDS</th>
<th>DDS+ SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Location Shar.</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Display Sharing</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Service Shar.</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Device Composition</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>5</td>
<td>Organization</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>6</td>
<td>Dev. Behavior</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>7</td>
<td>Dynamic activity flow</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>8</td>
<td>Log availability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Activity Script availability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>Multi-layer organization</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>11</td>
<td>Coexistence of organizations</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Not addressed</td>
</tr>
<tr>
<td>12</td>
<td>Resilience to unexpected changes</td>
<td>No</td>
<td>Very low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 2. Coverage of requirements by existing system architectures.

IV. DECENTRALIZED AND DISTRIBUTED SELF-ADAPTIVE ORGANIZATIONS

Multiple iterative versions of distributed mobile applications with a focus on M-learning have been created, to cover the requirements introduced in section II. In our latest design, the use of the Mobile Virtual Device (MVD) concept, presented in [5], offers sharing of resources and the organization management. The MVD is a conceptual organization where several devices work in collaboration sharing their resources to achieve a set of goals that are determined by a group of concerns. One of the characteristics of the MVD is that depending on the organization to be formed, the devices in the MVD will have different behaviors to play. Technical information about its development can be found in [6]. The design has considered a DDS implementation using a multi agent system for the creation of organizations and service provision and consumption. With a decentralized architecture, it is possible to liberate a central node (potential bottleneck) from dealing with concerns that can...
instead be addressed in each MVD, such as information availability for each student group [15].

Furthermore, our activities involve several stakeholders, having their own activity goals and concerns that may not be compatible. This requires a mechanism that evaluates the coexistence of concerns, to avoid potential conflicts. The mechanism will decide either that (a) all the concerns can be completely satisfied by a MVD organization or if (b) there is a need for dealing with concern trade-offs to satisfy the concerns to some extend so the activity can still be performed.

In our solution, a concern interpreter has been placed to perform this issue. The interpreter will periodically, based on changes of context, evaluate the viability of a MVD organization that fulfills the set of defined concerns. Based on the decision taken by the interpreter, the members in the MVD will be notified to compose the new MVD organizations. Figure 2 illustrates the components in the system and their interconnections.

Figure 2. Self-adaptation of MVD organizations through the use of a concern interpreter

V. CONCLUSIONS AND FUTURE EFFORTS

We have presented a list of requirements originated from a series of m-learning activities we have been conducting in the last 2 years. We have elaborated on the possible system architectures to conclude that the study of a decentralized and distributed self-adaptive architecture would bring the capabilities needed for our current and near future M-learning applications.

The design, deployment and implementation of a self-adaptive system for supporting M-learning activities pose some challenges that need to be studied in detail. We have presented a generic overview on a novel solution that offers a self-adaptive architecture to address changes in the environment and in the activity goals and concerns, characteristics that are necessary to tackle the problems associated with our M-learning projects. The technical aspects of this solution can be read on [6]. This solution makes use of a concern interpreter to offer a mechanism to define multiple concerns that the organizations must cover. This is a quite multifaceted setting which becomes quite complex in mobile distributed systems, as the environment and the situations where the activities take place are highly dynamic.

REFERENCES
