

# Personalized and context sensitive foreign language training supported by mobile devices

**Jan-Kristian Markiewicz**

Master of Science in Computer Science

Submission date: July 2006

Supervisor: Monica Divitini, IDI

Co-supervisor: Sobah Abbas Petersen, IDI  
Sondre Skaug Bjørnebekk, Inspera



# Problem Description

Inspira delivers content management solutions where the same content can be distributed in different formats or channels. These systems can among other things be used for production of digital teaching materials. This project is supposed to target central issues with personalized teaching on mobile devices. The project is restricted to the training of words from foreign languages.

It is important that the user gets personalized lists of word to practice on. The list will be built as the user reads and works with the course material. For this project we assume that this list is created manually, even though this will be done by means of text analysis at a later point. The system will control the connection between teaching resources and the relevant word list (the word list to a certain text is defined as a teaching unit) and offer automatically generated tasks where the user is supposed to insert the correct words based on context.

In this Masters thesis the student is expected to evaluate different development platforms for mobile devices, choose a method to personalize repetition practice based on an expected rate of memory persistence that allows the use of mobile devices and develop a prototype of such a system.

A task list for the project:

1. Research the possible technical platforms that can be used to develop applications for mobile devices.
2. Evaluate possible algorithms for personalizing the foreign language training.
3. Implement a prototype of a personalized repetition practice application for a mobile device based on the chosen algorithm.
4. Evaluate how scalable the prototype is, including a suggestion of a method for evaluating the effect of personalization.

A depth analysis of the language model that could be used to automatically generate lists of words for repetition is out of scope of this project. Only the following repetition tasks are expected to be considered: translate chosen words in a text, simple word translation (list of words).

The server part and database modeling (Oracle) used for the prototype will be implemented using the Inspira Platform (J2EE).

Assignment given: 20. January 2006  
Supervisor: Monica Divitini, IDI



## Preface

This report is a result of my Master's thesis project titled *Personalized and context sensitive foreign language training supported by mobile devices*. The project was supplied by Inspera AS, and was originally focused on how to best use learning content from existing content management systems on mobile devices. My advisors and I quickly agreed that it would be very interesting to focus more on the possibilities mobile computing could provide to foreign language learning. Inspera were also included in this process and they did not have any objections. The shifted focus of this project, and my redefined research questions, are well documented in the first three subchapters of chapter 1.

I would like to thank my supervisors Sobah Abbas Petersen, Sondre Skaug Bjørnebekk and Monica Divitini for all their helpful advice and support during the work on this project. A special thanks to Sobah who I worked closest with while doing my research in Trondheim. I would also like to thank my family and friends for all their support and for always believing in me. Also a super thanks to my wife Ingunn for promising to spend her life with me.



Jan-Kristian Markiewicz – July 21st 2006, Tønsberg



## **Abstract**

In this research project I try to answer how context sensitive mobile devices can be used to complement traditional foreign language learning in a classroom. My approach to answering this question is based on the goal of creating a prototype application called the “*Personalized And context sensitive foreign Language training system for real Life situAtions using mobile devices*” (PALLAS). To be able to develop PALLAS satisfactory I start by performing a prestudy to create a theoretical ground for my ideas. This allows me to come up with ideas of how PALLAS could be used. To systemize my process I create a scenario-based development method. I start by writing user scenarios of the PALLAS system in use. This is the part where new ideas are included into PALLAS. Then I analyze the user scenarios and create formalized models of how PALLAS is supposed to work. The models are in turn used to design a system architecture, which finally is used to implement the PALLAS prototype. I finish my research by comparing the finished PALLAS prototype to the original user scenarios. It turns out that the process has resulted in an architecture that can almost completely satisfy the features found in the user scenarios. With this solid architecture in place the next natural step for PALLAS will be testing in real-life scenarios with students and teachers using the system.

# Content

<b><u>1</u></b>	<b><u>INTRODUCTION</u></b>	<b><u>4</u></b>
1.1	BACKGROUND	4
1.2	PROBLEMS TO BE ADDRESSED	4
1.3	METHOD	5
1.3.1	THE SCENARIO-BASED DEVELOPMENT METHOD	6
1.4	REPORT OUTLINE	7
<b><u>2</u></b>	<b><u>PRESTUDY</u></b>	<b><u>9</u></b>
2.1	APPROACHES TO LANGUAGE TEACHING	9
2.1.1	WHERE DOES PALLAS FIT IN?	10
2.2	GUIDELINES FOR DESIGNING LANGUAGE TEACHING COMPUTER PROGRAMS	11
2.3	LANGUAGE TEACHING REQUIREMENTS FOR PALLAS	13
2.4	POSSIBILITIES AND IMPLICATIONS OF USING MOBILE DEVICES	14
2.5	EXISTING MOBILE FOREIGN LANGUAGE LEARNING APPLICATION	16
2.6	THE USE OF CONTEXT	18
2.6.1	TYPES OF CONTEXT	18
2.6.2	PHASES OF CONTEXT	19
2.7	USE OF PERSONALIZATION AND ADAPTIVITY	20
2.8	SUMMARY	23
<b><u>3</u></b>	<b><u>USER SCENARIOS</u></b>	<b><u>24</u></b>
3.1	INTRODUCTION	24
3.2	USER SCENARIO 1 – LANGUAGE STUDENT IN HER HOME COUNTRY	24
3.3	USER SCENARIO 2 – LANGUAGE STUDENT IN A FOREIGN COUNTRY	25
3.4	SUMMARY	26
<b><u>4</u></b>	<b><u>SCENARIO ANALYSIS</u></b>	<b><u>27</u></b>
4.1	INTRODUCTION	27
4.2	DEFINING THE USERS	27
4.3	ANALYSIS OF CONTEXT NEEDS	28
4.4	USE CASES	30
4.4.1	USE CASE 1 – STUDENT USING PALLAS	31
4.4.2	USE CASE 2 – TEACHER USING PALLAS	32
4.4.3	USE CASE 3 –SERVICE PROVIDERS USING PALLAS	33
4.5	CONTENT MODULES	34
4.6	FUNCTIONALITY CHECK LIST	37
4.7	SUMMARY	38
<b><u>5</u></b>	<b><u>SYSTEM ARCHITECTURE</u></b>	<b><u>39</u></b>
5.1	INTRODUCTION	39
5.2	SYSTEM OVERVIEW	39
5.3	THE PALLAS SERVER	41
5.4	TEACHER AND ADMIN CLIENTS	43



5.5	MOBILE SMART CLIENT .....	44
5.6	POSSIBLE MOBILE DEVICE DEVELOPMENT PLATFORMS .....	46
5.7	SUMMARY .....	49
<b>6</b>	<b><u>THE PALLAS PROTOTYPE .....</u></b>	<b>50</b>
6.1	INTRODUCTION.....	50
6.2	TECHNOLOGY CHOICES .....	50
6.3	TEACHER CLIENT.....	52
6.4	MOBILE CLIENT FOR STUDENTS.....	55
6.5	LIMITATIONS OF THE PROTOTYPE .....	57
6.6	SUMMARY .....	58
<b>7</b>	<b><u>EVALUATION AND CONCLUSIONS .....</u></b>	<b>59</b>
7.1	EVALUATION OF THE PALLAS PROTOTYPE .....	59
7.2	CONCLUSION .....	59
<b>8</b>	<b><u>REFERENCES .....</u></b>	<b>60</b>

## Figures

Figure 1: Overview of how my method will help me answer my research questions ...	5
Figure 2: The four-step scenario-based approach used for creating the PALLAS prototype .....	7
Figure 3: Predicted, dynamic and static context over time .....	20
Figure 4: Dynamic parameters are marked with a circle of arrows. The .....	22
Figure 5: Use Case diagram showing system operations that are relevant when the student uses the PALLAS client .....	31
Figure 6: Use Case diagram showing how teachers interact with the PALLAS system .....	33
Figure 7: Use Case diagram of the Service Provider's interactions with PALLAS ....	34
Figure 8: PALLAS' different content modules and their relationships .....	36
Figure 9: System overview of the PALLAS architecture .....	40
Figure 10: Overview of the PALLAS server architecture .....	41
Figure 11: Four-tiered client server application with synchronization .....	43
Figure 12: Mobile Smart Client application architecture .....	45
Figure 13: Overview of the technology used in the PALLAS prototype .....	51

## Tables

Table 1: Summary of how PALLAS addresses requirement and issues of language learning .....	14
Table 2: Hardware issues for mobile devices .....	15
Table 3: Summary of the key differences between LOCH and PALLAS .....	17
Table 4: Summary of what kind of context information the language learning system uses .....	30
Table 5: Summary of some of the advantages and disadvantages of the mobile development platforms .....	48

# 1 Introduction

## 1.1 Background

The digital revolution is affecting almost all parts of our society. We read newspapers on the internet, write reports in word processors and go to the bank through a web browser. It has never been this easy to quickly share content, and we have never had access to such enormous amounts of information as we do now. With these possibilities we unfortunately also produce problems. Too much content can be hard to manage and synchronize in an orderly fashion. To address this problem several companies have started to sell products for content management. The Norwegian company Inpera [1] delivers such systems. These systems make it possible to organize information and knowledge, and to transfer it to different channels or media from the same source. Teaching materials can also be organized in such systems, and Inpera works among others with publishers of teaching materials for foreign language learning. Currently the material is mainly available by using personal computers, but making the content available for mobile computing devices opens the possibility for many new interesting scenarios. Today, a cellphone is an every day item and Personal Digital Assistants (PDA) are also common. This project seeks to answer how these mobile devices, together with a content management system, can be used to provide personalized and context sensitive foreign language training.

## 1.2 Problems to be addressed

This research project's main goal is to develop a prototype of a mobile application for foreign language learning. This prototype can then be used in future research in the field of mobile language learning. When creating an application like this it is important to keep in mind that learning is a highly subjective process. Because learners will vary both in how they learn and what they want and need to learn, we can not say that there is one best way of learning [2]. Ideally a mobile language learning application would therefore adapt to the user so that it would provide the best possible learning experience. To create such an application is not a simple task. There are many factors one needs to take into consideration, both pedagogical and technical difficulties needs to be overcome. To make the research problem more manageable I have formulated a main problem which I in turn have reduced to smaller research question.

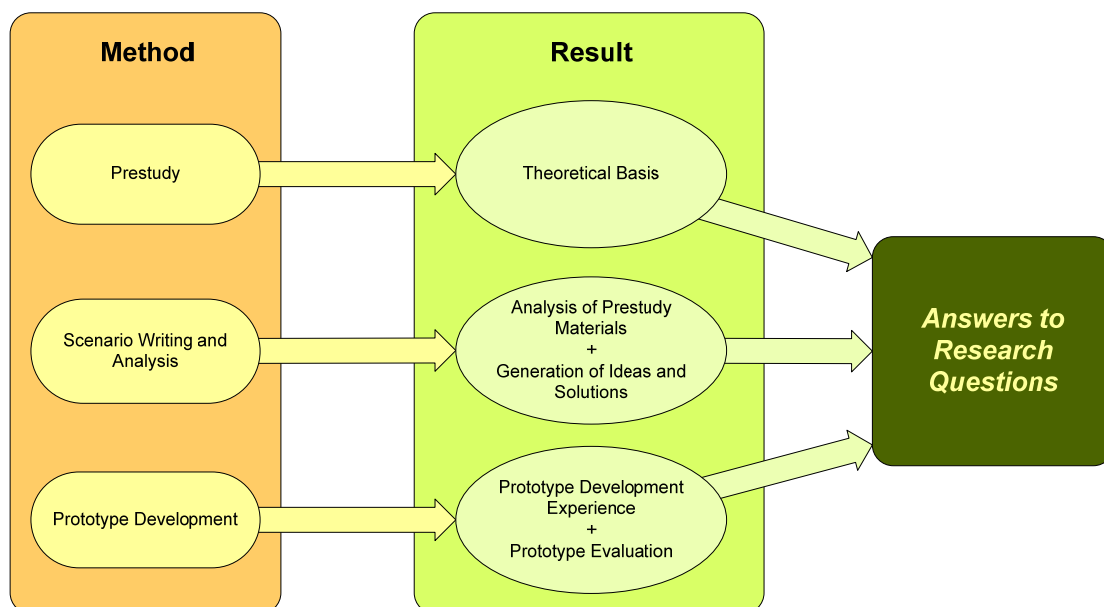
The main research question of this project is: *How can context sensitive mobile devices be used to complement traditional foreign language learning in a classroom?* Which I have further divided into the following research questions:

1. What approaches do we have to language teaching?
  - a. How does a mobile application fit into this?
2. What kind of possibilities do mobile devices provide in terms of language learning?
  - a. Are there any guidelines for language teaching software that should be followed to avoid common pitfalls?

3. What kind of implications do we need to take into account when using mobile devices?
4. How can we take advantage of context to enhance the language learning experience?
  - a. What type of context can be used?
  - b. How can we model context for use in mobile language learning system?
5. How can personalization and adaptivity be used to enhance language learning experience?
6. What mobile development platforms are suitable for development of a mobile language learning application?
7. How should a system for personalized and context sensitive foreign language learning be designed?

### 1.3 Method

The goal of this research project is to answer the questions in section 1.2 through the development of a working prototype of a mobile foreign language learning application. The prototype is called the “*Personalized And context sensitive foreign Language training system for real Life situAtions using mobile devices*” (PALLAS). It is referred to as PALLAS in the rest of this report. Pallas is also the name of a Titan in Greek mythology [3]. He was referred to as the god of wisdom. This is a good fit as the PALLAS system is intended to provide students wisdom in the form of language knowledge. Other projects, such as the Cyberguide Case Study [4], demonstrate how building a working prototype can be used to better understand how context-aware computing can add value to the field of interest.



**Figure 1:** Overview of how my method will help me answer my research questions

Figure 1 shows an overview of how I have planned this project. I will start my research by doing a prestudy. First I will research the field of language teaching, and

then use this to try to figure out the pedagogic requirements for the language learning system. I will then look at the possibilities and implications of using mobile devices. Further I will describe how context-awareness and personalization fit into the picture of language teaching. This prestudy will give me a theoretical basis for the next phases of the research project.

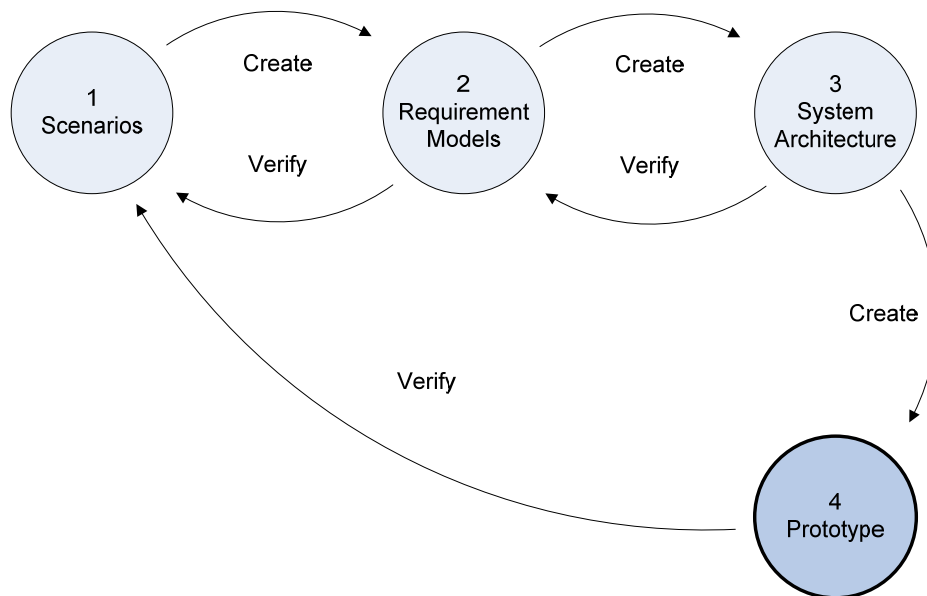
When I am finished with my prestudy I will start to design PALLAS. I will use a scenario-based approach [5] to help me identify the user groups and system features and requirements. The scenarios will be based on my analysis of the prestudy material and my own ideas and suggestions.

The final phase of my research project will be the development of a PALLAS prototype. By implementing a prototype I will learn how well the proposed system architecture works, and I will also gain hands-on experience on how well my ideas on context-sensitivity and adaptability fit the realm of mobile language learning. I will test and evaluate the prototype based on the feature requirements I compose during the scenario analysis. This will give an indication on how well the prototype can meet the requirements I have found to be necessary for a mobile foreign language learning application to be successful.

The combination of the results from the three phases should allow me to be able to answer all my research questions.

### **1.3.1 The Scenario-based Development Method**

As previously mentioned, I am going to use a scenario-based method to develop the PALLAS prototype. My approach is a four-step process. First I will write textual scenarios of how the system is supposed to work. These are easy to understand for people who have domain knowledge but lack extensive technical knowledge. Feedback can easily be incorporated at this stage. The next step is to transform scenarios into generalized models which contain the system's requirements. For this purpose I will utilize use case diagrams. To verify that the use case diagrams cover all the requirements I will match them up to the scenarios. I will also create a checklist of all the features. This will be used later to verify that the prototype works according to the specifications. In the third step of the process I will create the system architecture of PALLAS. The use case diagrams will help me identify the necessary system components and also allow me to verify that all requirements are covered. Scenario generation and usage combined with rapid prototyping has been showed to have a symbiotic effect [6]. The final stage of the scenario-based process will therefore be the development of a prototype of PALLAS. The scenarios I create do not only function as design inspirations, but also as material to test the prototype design [7]. I can thus use the scenarios to verify that the prototype satisfies the needs of the PALLAS system. This will be done by using the feature checklist created in step two. The whole development process is illustrated in Figure 2.



**Figure 2:** The four-step scenario-based approach used for creating the PALLAS prototype

After I have finished the development of the PALLAS prototype I will perform an evaluation of the PALLAS architecture. Experiences from the implementation and test-runs of the system will be presented.

Conducting a full scale test of the prototype on a group of students is out of the scope of this project and subject to future research. The prototype will however be of such quality that it can be used in a large scale test.

### **1.4 Report Outline**

This report can be roughly divided into five phases: Prestudy, idea generation, analysis, implementation and evaluation.

I will start with a prestudy covered by chapter 2. It will cover theory about foreign language learning, guidelines for language teaching programs, possibilities and limitations of mobile devices and theory about context and adaptivity. This prestudy will give me a solid foundation to start working on my own solution, PALLAS.

Chapter 3 is the central part of the idea generation phase. In this chapter I will present user scenarios of how I think PALLAS ideally should work based on the knowledge from I gained through the prestudy.

With all the ideas from chapter 3 loose it is important to create some sort of formalized knowledge. This is what chapter 4 is used for. In this chapter I will create formal requirements models that can describe what PALLAS should be able to do. To do this I will among other things utilize use case diagrams and a functionality check list.

Now that we know what PALLAS should be able to do, we need to figure out how such a system should be developed. This is the implementation phase. It starts with chapter 5 where I suggest a system architecture for the complete PALLAS system. In chapter 6 I describe the prototype I have developed based on the architecture in chapter 6.

Finally, in chapter 7, I perform the last phase – a quick evaluation of the PALLAS prototype.

## 2 Prestudy

### 2.1 Approaches to Language Teaching

Teaching someone a foreign language is not an easy task. Because there are many different ways to describe languages there are also many variations in the approaches to teach them [2]. There are a number of clearly separable approaches that can be identified. The following is a list of the most usual ones:

- **Grammar-translation method:** This method focuses on the learning of vocabulary that is relevant to the texts being studied, and the language rules necessary to translate the text from the target language to the native language. This works well in a classroom format, and because the learners are expected to work independently it also lends itself well to technological implementation. The weaknesses of this method are its narrow focus on reading and translation of texts, and also the lack of dealing with spoken language and other forms of communication.
- **Oral approaches and situational language teaching:** This approach focuses on spontaneous use of oral language and absence of translation. The learning is conducted only in the target language, and the teacher is the principal source of information instead of textbooks and texts. This makes the success of the method highly dependant on the teacher's responsiveness and abilities. It is also difficult to figure out how such an approach can be adapted to the limitations of technology. One of the strengths of the method is that there has been put a lot of scientific effort into identifying items to include into the syllabus to make it truly relevant and motivating for the learners.
- **Audio-lingual approaches:** This approach is highly structural and focuses on the differences between the native and target languages. This requires that the learner learns a lot about languages, rather than the language itself. Because this approach emphasizes formal drilling and memory it has been widely used with technology such as tape recorders and the language laboratory. Through the language laboratory learners get to focus both on speaking and listening as well as reading and writing. However, this method can seem both tedious and inefficient. It does not seek to motivate the learner because the interest for learning is not an important aspect in this method. Repetition alone is intended to be enough to induce learning. When creating material for technology-based methods it can be very easy to follow the audio-lingual approach because gap-fill and drill tasks are easy to program. Tasks of this sort can be useful to a limited degree, but in excess they can be counter-productive.
- **Communicative language learning:** Not surprisingly, this method seeks to introduce communicative competence. This is usually attempted by carrying out tasks or solving problems in pairs or groups using the target language. The communicative approaches focus on meaning, while teaching of structure is downplayed. The importance of memory is also minimized in favor of internalization of language rules which may not be explicitly explained. The source of motivation for this approach is thought to be the naturalistic use of language for communication. Because the method can be applied to almost any theme or context it gives a lot of flexibility in the selection of genuinely interesting material for the learners. All the interactive foreign language use



required by communicative language learning also allows great fluency gains. However, because this method has been seen as a rationale for never teaching explicit structure, learners have been found to be notoriously weak in this area. The approach requires a lot of meaningful exposure for learning to take place. Technology-based materials could therefore be helpful to create the needed volume of meaningful exposure, and also to practice task and problem-solving activities.

It is important to recognize that these methods are not exclusive. Teachers will often be inspired by more than one method in their teaching. Additionally there will also be a variation in the learners. They can vary both in obvious attributes such as age and gender, but also in preferred learning style and learning preferences. Statistical analysis on Carroll and Sapon's language learning model [2] has resulted in the clear identification of two styles of learning: the analytic learners and the memory-based learners. Later a third group also emerged; people who are good at both [8]. If we relate these groups to the language teaching approaches, we can assume that analytic learners probably would get best results with the grammar-translation method and memorizers with the audio-lingual approach.

Gardner [9] has another way of describing the characteristics of learners. His hypothesis presents seven types of intelligences which the learners might have different amounts of. The intelligence types are:

- Plays with words (verbal-linguistic)
- Plays with questions (logical-mathematical)
- Plays with pictures (visual-spatial)
- Plays with music (music-rhythmic)
- Plays with moving (body-kinesthetic)
- Plays with socializing (interpersonal)
- Plays alone (intrapersonal)

To maximize learning in a group one should make sure that all these intelligence types are addressed. This allows learners of any intelligence type to find material and exercises that retain their interest.

Because of all the variation in learners we can not say that there is one best way to learn a foreign language. The best practice in teaching may vary from learner to learner. In an ideal world, teachers would therefore analyze learners' individual needs and circumstances and adapt their teaching style based on that.

### **2.1.1 Where does PALLAS fit in?**

PALLAS is a system targeted for use on mobile devices, and will not be used in a typical classroom setting. The mobility allows for real-life scenarios which simply can not be fully achieved in a classroom. PALLAS combines elements from the different teaching approaches mentioned in the previous subchapter, and will hopefully gain some of their strengths while covering over some of their weaknesses. This is intended to allow PALLAS to combine explicit learning of vocabulary and language rules with unconscious skill development in the fluent application of the language. Some of the features inspired by the teaching approaches are:

- **Grammar-translation method:** The textbook publishers can connect their content management system to PALLAS. This gives user access to their textbook material, including glossaries to each text. This material is well suited for independent work, and allows the user to learn wherever they are. The issues concerning lack of dealing with spoken language and communication is dealt with by not having this feature as the main learning method.
- **Oral approaches and situational language teaching:** The students are given tasks they have to perform in the real world. These tasks have to be performed by using the target language. If the tasks are well thought out, relevant and interesting, it will help to motivate the students. Unfortunately it is not possible for all the students to have a teacher by their side to help them at all times. PALLAS tries to compensate for this by offering some help related to the current task while it is being performed.
- **Audio-lingual approaches:** Gap-fill and drill tasks will be available for students who prefer these kinds of exercises. However, they are not a central feature of the PALLAS system.
- **Communicative language learning:** This approach is covered by the same features as the *oral approaches and situational language teaching* method. It is perfect for the real-life scenarios made possible by PALLAS' mobility and context sensitivity. This allows for a lot of meaningful exposure of the target language in use.

The PALLAS system does not automatically create the language learning content, and as such it is not responsible for the quality of the content. However, PALLAS needs to have the possibility to store and distribute content that is pedagogically sound. One of the ways to do this is to support content and exercises which cover all of Gardner's [9] intelligence types. This should make it possible to provide content that can fit all kinds of learners.

PALLAS is also adaptive, so it is able to accommodate the variations in the learner groups. Each student has their personal user profile that keeps track of their progress and preferences. Analysis of this profile can be used to find a student's preferred learning style and then adapt the application to better cater for that style.

## **2.2 Guidelines for Designing Language Teaching Computer Programs**

There has been conducted research on language teaching materials and software for many years. As a result several suggestions and checklists have been produced to help in the creation of language teaching software. Using a checklist in the design stage of software can allow the developers to avoid common mistakes and pitfalls. One of the most systematic and complete checklists [2] for evaluating language teaching computer software was written by Philip Hubbard. The following is a list of the main considerations from Hubbard's list [10] [11] that are relevant for PALLAS:

- **There should be a consistent approach and method leading to the materials:** Technology should not be used simply because it is possible. It should be used to aid the learning process. There is therefore a need for a clear

idea of how the languages are to be learned. This should be reflected through the materials and exercises.

- **There should be a clear set of objectives:** It should be clear what you are trying to teach, and how this is supposed to be achieved.
- **The materials must be usable:** It must be clear for both the learners and teachers how to use the system, and it should be quick and easy to use. Things to take into consideration include saving, switching between different tasks and help sections/instruction manuals.
- **There should be feedback to user responses:** It is hard to learn a language when you never know whether you are right or wrong. Software can be used instantaneously to grade, correct mistakes and provide feedback and explanations on exercises. It is important to take into consideration the effect of feedback, i.e. if negative feedback is presented harshly it can hurt the student's motivation.
- **The material should be appropriate for the age of the learners:** There is a wide spread in the different learners' interests. A lot of this is based on their age, i.e. educated, adult learners can interact on completely different topics than young learners in the early stages of their education. Materials must address the age level of learners precisely for learning to take place. Other factors like sex, native language and specific learning needs should also be taken into consideration.
- **The materials must be motivating:** There are many types of motivation, for example instrumental (learning a language because you need to use it), intrinsic (motivated because you enjoy it) and extrinsic (external rewards, i.e. good grades). It is not possible to simply go with one of these and hope that it works, because the type of motivation that works best is based on each individual. It is, however, certain that learners need to be motivated for learning to take place. Intrinsic motivation can for example be affected by good choice of topics and materials. Basing the materials on age could help with this. Extrinsic motivation can be affected by offering appropriate rewards for success in language learning.
- **The materials must fit the language level of the learner:** If the material presented to learners is too hard there is a big chance that they will not understand anything, and thus not learn anything. Successful teaching materials should therefore be pitched just above the learners' level of knowledge, and they also need to be comprehensible. Allowing the students to be able to solve exercises is also good for their self-confidence and motivation. If there is a big variation in the users' language level the software should be able to adapt to the different users. Pre-testing and progress testing of the learners can be used to find out where they stand.
- **Is the internal organization of the materials sufficient and appropriate:** Technological materials should be utilized so that they support meaningful repetition and recycling of content. The greater the repetition, with appropriate spacing, the greater the likelihood of an item being learned. The choice of activities and sequencing of content should also be adapted to the learners and their goals.
- **The materials should provide enough time for learning:** The time spent in meaningful language learning is usually the number one factor of success. Language learning systems should therefore provide hundreds of hours of activities for it to be able to make a serious impact on language learning.

### 2.3 Language Teaching Requirements for PALLAS

Research in educational sciences, and more specifically the field of language teaching, has resulted in many theories, classifications and best-practices. Some of these have been presented in the previous chapters. For PALLAS to be successful it is important to try to leverage the results from this research. Table 1 summarizes some suggestions on how PALLAS can try to solve the different requirements and issues previously discussed.

Requirement/Issue	PALLAS property or feature
<i>Variation in learners (preferred learning style, native language, age, interests etc.)</i>	<ul style="list-style-type: none"> <li>• PALLAS needs to be adaptive and adjustable according to each individual user’s preferred learning style, interests etc.</li> <li>• It should be possible to create materials and tasks that can target all the language teaching approaches and intelligence types listed in chapter 2.1.</li> </ul>
<i>The material must be motivating</i>	<ul style="list-style-type: none"> <li>• PALLAS does not create the content automatically, but needs to allow creation of new content. The possibility to use existing content by connecting to content management systems would be very useful.</li> <li>• Users’ interests, skill level and other parameters should be stored and updated in a personal user profile. This can then be used to serve personalized learning material that motivates by including content the student is interested in, and by having an appropriate difficulty level.</li> <li>• PALLAS allows real-life situation training, and will hopefully be motivating by guiding the students through use of the foreign language in the real world. Mobility and context-sensitivity is needed to support this.</li> </ul>
<i>Provide enough time for learning</i>	<ul style="list-style-type: none"> <li>• Allowing the PALLAS system to be accessed through mobile devices will allow language training to take place wherever, whenever. For classes using a language laboratory, Kreeft-Peyton [12] found that half of all class time was spent on getting to the laboratory, logging on and setting up. PALLAS’ mobility would allow quick access to the system and eliminate the dead time that often arises when switching from standard classroom teaching to using software aided teaching.</li> <li>• By reusing content from existing content management systems, PALLAS could offer enough materials for students to learn language concepts that need a lot of attention.</li> </ul>
<i>Feedback to user responses</i>	<ul style="list-style-type: none"> <li>• PALLAS should give immediate feedback after a user solves a task or exercises. A score should be provided and mistakes should be explained.</li> <li>• The teacher should be able to grade students’ work manually. When the grading is finished the results could immediately be made available to the students.</li> </ul>

<i>Usability</i>	<ul style="list-style-type: none"> <li>• The graphical user interface for all clients needs to be developed with human-computer interface techniques, such as metaphors [13], kept in mind.</li> <li>• Special considerations need to be made when design the mobile clients because of their limited output (small screen size) and input (no full-size keyboard) capabilities.</li> </ul>
<i>Consistent approach and clear set of objectives</i>	<ul style="list-style-type: none"> <li>• This is basically up to the creators of the language learning materials, but PALLAS needs to support creation and presentation of the materials' objectives and goals.</li> </ul>
<i>Easy to replace language content</i>	<ul style="list-style-type: none"> <li>• The system architecture should allow the language content to be replaced easily while retaining programming and ideas common. This can be used to provide language training for several different foreign languages.</li> </ul>
<i>Meaningful repetition</i>	<ul style="list-style-type: none"> <li>• The PALLAS profile should keep track of users' progress as they use the system. This can be utilized to provide meaningful repetition because the system knows what you need to practice on. This will of course require that enough materials are available.</li> </ul>
<i>Support the instructor's teaching in the classroom</i>	<ul style="list-style-type: none"> <li>• Teachers should be able to choose tasks and exercises to make available for their students.</li> <li>• If no relevant tasks/exercises are available the teacher should be able to create new ones that based on what they are doing in class.</li> </ul>
<i>Keeping track of students' scores</i>	<ul style="list-style-type: none"> <li>• The teacher should be able to use PALLAS to keep track of the students' progress. Scores and statistics on task and exercise solving should be made available to the teacher.</li> </ul>

**Table 1:** Summary of how PALLAS addresses requirement and issues of language learning

All the suggestions presented here have been used in the design of PALLAS. By addressing these issues at the design-stage we will hopefully avoid unnecessary problems when the prototype is already implemented.

## **2.4 Possibilities and Implications of Using Mobile Devices**

Mobility is one of the most important aspects of PALLAS. It creates a lot of possibilities that are not achievable when using more conventional language teaching software on regular PCs, such as language learning through real-life situations and system access independent of time and location. However, there are a lot of key issues with mobility that should be considered. Naismith et al. [14] discuss the following implications:

- **Context:** PALLAS needs context information to make the system personalized and to be able to provide real-life training scenarios. However, it is important to take the users' wish for anonymity and privacy into consideration. For example, sharing location information with your peers and teacher should definitely be a feature that requires the users' consent.
- **Mobility:** Being able to escape the classroom and still engage in language learning is great. Unfortunately, when the students are out of sight from the

teacher it can be difficult to monitor that they are in fact doing school work. When moving outside of the class room non-school based distractions are likely to occur.

- **Learning over time:** When solving tasks and exercises on the go it can be hard to keep track of everything you have done. It is therefore important to have effective tools for recording, organization and retrieval of the mobile learning experiences.
- **Ownership:** Students may want to own and control their personal technology. This can be a problem when they are forced to use the mobile devices for some purposes only.

In addition to these social/usage related problems, mobile devices also have several limitations in form of hardware if we compare them to full featured PCs. Many of the issues discussed by Forman et al. [15] are still valid today, but progress in mobile hardware is happening very fast. Table 2 summarizes these issues and presents some possible workarounds.

Issue	Problem	Workarounds
<b>Network connectivity</b>	<ul style="list-style-type: none"> <li>• Not always connected</li> <li>• Limited bandwidth</li> <li>• Expensive (i.e. pay pr. downloaded kb)</li> <li>• Uses battery power</li> </ul>	<ul style="list-style-type: none"> <li>• Cache data when a good connection is available</li> <li>• Use cached data when no/bad connection is available</li> </ul>
<b>Processing power</b>	<ul style="list-style-type: none"> <li>• Usually a lot slower compared to PCs</li> <li>• Heavy use of the CPU drains the battery</li> </ul>	<ul style="list-style-type: none"> <li>• Use centralized servers to do heavy calculations (requires network connectivity)</li> </ul>
<b>Battery life</b>	<ul style="list-style-type: none"> <li>• Limited battery capacity</li> </ul>	<ul style="list-style-type: none"> <li>• Minimize use of CPU and wireless network access</li> <li>• Add larger battery at cost of portability</li> </ul>
<b>Risks to data</b>	<ul style="list-style-type: none"> <li>• Bigger chance of physical damage when the unit is moved around</li> <li>• Unit can get stolen or lost</li> </ul>	<ul style="list-style-type: none"> <li>• Synchronize data (backup) regularly</li> <li>• Sensitive data can be encrypted</li> </ul>
<b>Storage capacity</b>	<ul style="list-style-type: none"> <li>• Usually small storage capacity</li> </ul>	<ul style="list-style-type: none"> <li>• If you have a lot of data: Store it on a centralized server (requires network connectivity)</li> </ul>
<b>User interface</b>	<ul style="list-style-type: none"> <li>• Limited input possibilities compared to PCs full keyboard and mouse</li> <li>• Limited output possibilities - usually a very small screen</li> <li>• Varying input/output properties between different units makes it hard to target many units with one user interface</li> </ul>	<ul style="list-style-type: none"> <li>• Customized user interface design for each (small) screen size/resolution</li> <li>• Minimize the need for text-input</li> <li>• Multimodality – more than one type of input/output, i.e. gestures, speech recognition and text-to-speech.</li> </ul>

**Table 2:** Hardware issues for mobile devices

As we can see from Table 2, the different workarounds often have a negative impact on one of the other issues. For example, if you have a slow processor you can use a centralized server to do the processing work. This requires a network connection and thus the issues related to network connectivity. When designing a system for mobile devices it is therefore important to be aware of these compromises. If you use a workaround you have to pay the cost somewhere else.

## **2.5 Existing Mobile Foreign Language Learning Application**

The field of mobile language learning application has existed for quite some time, and there has been developed a range of different applications. A lot of them are fairly specialized in certain fields of language learning, such as the *PhotoStudy* vocabulary study system [16] and *Mobile Adaptive CALL* which is designed to help Japanese English-speakers distinguish the English phonemic contrasts [17]. The system that has most in common with PALLAS is called LOCH [18] and was developed at the Tokushima University in Japan. LOCH (Language-learning Outside the Classroom with Handhelds) was created to help overseas students to learn Japanese while involved in real life situations. The teacher assigns field activities to the students. The students can then bring a PDA and go around town to fulfill the tasks and share their individual experiences. To help fulfill their tasks they can write down annotations, record questions, take pictures and communicate with their teacher through instant messaging or IP telephony. The teacher can monitor the students' position and guide them through their tasks by providing suggestions or hints. After all the students have finished their tasks, they meet together in the classroom and use a smart board to display and discuss how they solved their tasks.

LOCH and PALLAS have some similarities in their approach of using real-life situations as a means for foreign language learning. However, there are many ideas and features that separate the two. For starters, LOCH is meant to be used for day trips, where the students are sent out with the specific goal of solving the provided tasks. PALLAS is designed to be a companion in the students' everyday life. It can be used in the same way as LOCH if the teacher wants to, but PALLAS can also help you learn even when the teacher does not have any specific tasks for you. This is possible because PALLAS has more content than the tasks created by your teacher. PALLAS is integrated with textbook publishers' materials directly through their content management system, and can therefore provide all this information in a version that is adapted for the mobile device. Having all this content available makes it possible to ask PALLAS for information or an explanation based on a keyword or expression. PALLAS can then for example return content from a textbook that explains that subject. You can also send asynchronous questions to your teacher. On the other hand, when using LOCH you always have your teacher online. This allows synchronous communication with your teacher. But LOCH does not have any additional content than what is provided by the teacher through his answers or tasks. When it comes to the systems mobile clients there is a big difference. LOCH uses a thin client approach by using a web-based application on the PDA to access the system. This makes it easy to update the client because you only need to update the web server application. Additionally applications for instant messaging and IP telephony have to be installed. The biggest drawback of using a web-based application to access the system is that it requires a constant internet connection. For

PALLAS we have instead created a smart client application. This is a mix between a thin and fat client. The smart client is not completely reliant on an internet connection because it can cache data for later use. Only relatively small amounts of data are stored in the cache, so PALLAS does not require the huge amounts of storage space a fat client with all PALLAS-content would. Both LOCH and PALLAS' servers are based on open standards, such as SOAP (Simple Object Access Protocol) and Web Services, so it possible to create new clients that work differently than they currently do. Table 3 shows a summary of the differences discussed in this paragraph.

<b>Point of comparison</b>	<b><i>LOCH</i></b>	<b><i>PALLAS</i></b>
<b>Usage</b>	<ul style="list-style-type: none"> <li>• Day trips specifically for task and exercise solving</li> <li>• Present how tasks where solved on smart board in class</li> </ul>	<ul style="list-style-type: none"> <li>• Is used in the students everyday life</li> <li>• Keeps track of students progress over a longer period of time</li> <li>• Students can also be sent out to solve specific task (i.e. during class or on a field trip)</li> </ul>
<b>Content</b>	<ul style="list-style-type: none"> <li>• Tasks and exercises created by teacher</li> </ul>	<ul style="list-style-type: none"> <li>• Common repository of tasks and exercises available</li> <li>• Teachers can create new tasks or exercises</li> <li>• Integration with publishers' content management system provides text book materials and exercises</li> </ul>
<b>Queries/asking for help</b>	<ul style="list-style-type: none"> <li>• Teacher answers all questions</li> <li>• Teacher is always online</li> </ul>	<ul style="list-style-type: none"> <li>• The students can query the system for answers</li> <li>• Questions can be sent to the teacher, but instant reply is not guaranteed</li> </ul>
<b>Mobile client</b>	<ul style="list-style-type: none"> <li>• Uses web browser to access the system</li> <li>• Internet connection required at all times</li> <li>• Instant messaging and IP phone application installed</li> </ul>	<ul style="list-style-type: none"> <li>• Uses rich client application that needs to be installed</li> <li>• Smart client supports caching, i.e. does not need internet connection at all times</li> </ul>

**Table 3:** Summary of the key differences between LOCH and PALLAS

Even though there are many differences between the two systems there are still some interesting results from the research done on LOCH that should be considered in the development of PALLAS. On the positive side the users of LOCH found the PDA to be helpful when they were confronted with situations that aroused doubts or when they were uncertain about the goals of their tasks. All of the users wanted to use the system again. This reflects positively on the idea of language training through real-life situations supported by mobile devices. Among the negative feedback they received, where complaints about the text input methods of the PDAs and the difficulties of



using GPS. These issues need to be considered during the development of the PALLAS mobile client, and can hopefully be solved by making the client's user interface well adapted to the mobile devices' limitations.

## 2.6 The Use of Context

Context is a popular buzzword in the world of mobile computing and has been given a lot of attention lately. According to Modafferi et al. [19] context is “...*any information that can be used to characterize the situation of an entity.*”. And an entity is defined as a person, place or object that is relevant to the interaction between the user and application. This definition allows us to characterize anything from the location of a car to the mood of person as context, but only as long as it is relevant to the interaction between user and application. Further we can define context-aware systems as systems that use context to provide relevant information or services to the user [20]. So to make our application context-aware we need to provide relevant information or services to our users, and we do this by obtaining information about relevant objects. This how to do it, the more important question however, is why do we need our language learning application to be context-aware? The goal of the language learning system is to teach the students a foreign language by encouraging them use the language in real-life settings. This is made possible by using mobile devices that the students always can bring with them. If the mobile language learning application simply was an encapsulated system, it would in reality be nothing more than a portable version of existing PC language learning applications. By making our system context-aware we can make it react and adapt to the user's current situation (context), and thus be able to interact with the user based on his real world situation. This is crucial to differentiate our language learning system from the standard language learning software that already is available.

### 2.6.1 Types of Context

Context covers a wide specter of different information types. It is therefore useful to break context down into different categories, as done by Krogstie et al. [21] and Göker et al. [22]. They describe five types of context:

- **Spatio-temporal context** deals with time and aspects of spatial extent, i.e. time, location, speed, direction, place and shape.
- **Environment context** captures the entities that surround the user. This can be anything from things, services, weather, noise to information sources such as movies, text etc.
- **Personal context** consists of two subparts. The first is context related to a person's *physiology*, such as pulse, weight, hair color and blood pressure. And the second one is context related to person's *mental* state, i.e. mood, stress and expertise. It is noticeable that items in personal context category might differ greatly in how dynamic they are. A person's pulse can change by the second, but a person's height is a lot more static.
- **Task context** is related to what the person currently is doing. This can be described through goals, tasks, actions, activities or events. For example, a janitor sealing a heat pipe might perform this action as part of a goal to prevent

heat leakage. Information about the current task can then be used to provide the janitor with information related to the task, or to adjust system behavior.

- **Social context** deals with the social portion of the user's current context. It can contain information about any person the user has some kind of relation to, for example friends, enemies, co-workers or neighbors. When describing the social context it is natural to assign the user to a role, and additionally connect the role to a social arena. One example could be an anesthetist working in an operating room.
- **Information context** is simply the information space that is available at any given time. This can be either physical information providing objects, such a book, or virtual information providers, such as a webpage.

### 2.6.2 Phases of context

Schilit [23] describes the actions carried out by context-aware software as a context-aware cycle. The cycle is a three step process which includes:

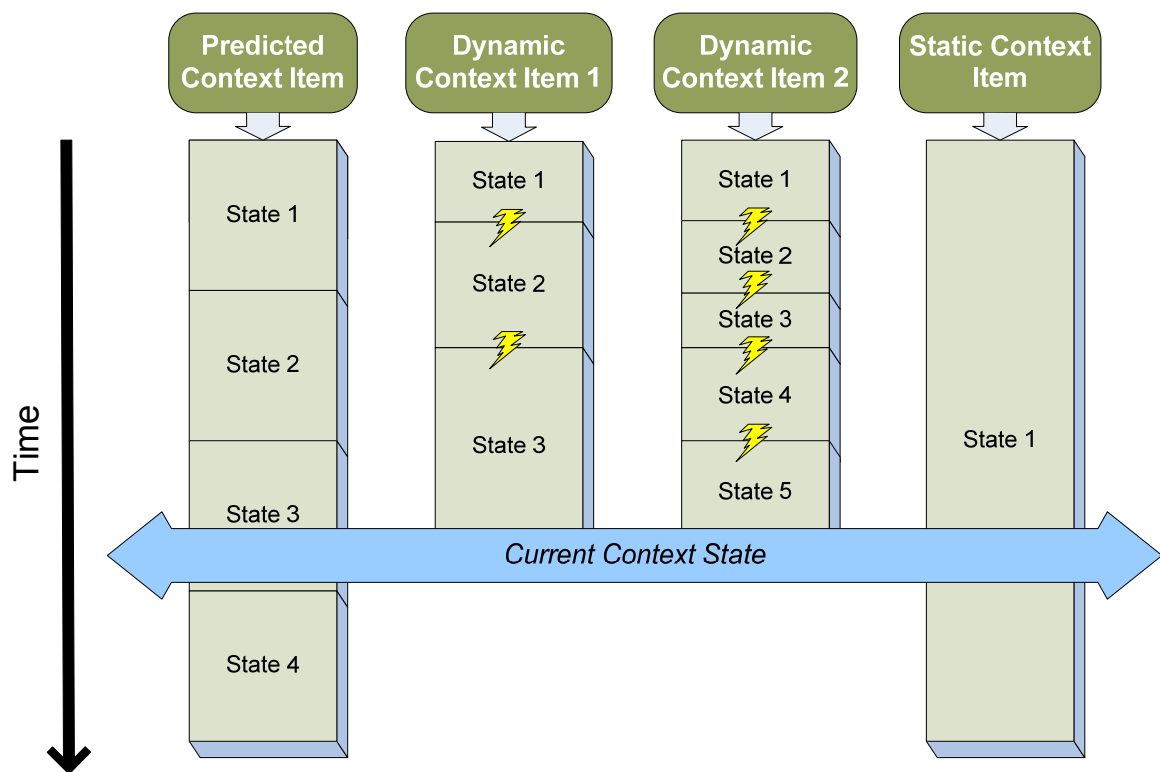
- **Discovery:** This is the first phase of the cycle. When the context changes the system needs to find out what resources that are available and what their characteristics and capabilities are.
- **Selection:** When the context resources have been found and updated, the system has to decide what context information to utilize.
- **Use:** With the essential context information selected the application can employ the new context resource. The fact that the resource is employed does not necessarily result in an application action. The application simply uses the resource to decide if an action is required, and possibly to decided what kind of action to perform.

PALLAS' context engine needs to be designed to support all three phases of the context-aware cycle to be able to react reliable according to context changes.

Context properties can be static, for example the users gender, but in most cases context properties are a dynamic [24] – the context changes as the user interacts with it. The change of context is actually what most context aware applications use to trigger their context events [25]. If the context is dynamic we can also keep track of the context history. Previous context values can then be used to make better decisions. For example can location history be used to make sure that PALLAS suggests that the user should visit a museum in the direction he is heading, rather than a museum back where he came from.

The topic of static and dynamic context properties is well covered in existing research documentation. However, the possibility of having predicted context information has not been given a lot of attention yet. One example where this could be useful is if you have a mobile device that synchronize with your computer occasionally. Weather context data, including the forecast for the next couple of days, can be downloaded via the internet each time you synchronize. This future context data can then be used without the need for any external measuring device, or even a current internet connection. Predicted context can also to a certain degree be used for location, for example if you write down the location of all your appointments in your digital calendar application. The weakness of predicted context is that you can never be

completely sure that the context information is true. The fact that your calendar appointment says you are supposed to be in a certain meeting room does not guarantee that you actually went to that meeting. There can also be different degrees of probability for the predicted context. The weather forecast for tomorrow will usually be more accurate than the one for next week. If an application is to use predicted context it should hence use some sort of probability algorithm that can weigh the significance of the predicted context data.



**Figure 3:** Predicted, dynamic and static context over time

Figure 3 illustrates the concepts of predicted, dynamic and static context. The predicted context item's state over time is already known. The dynamic context item state-changes are unpredictable. External events (shown as lightning bolts) cause the context to change, for example the location changes when the user moves. We do not know when such an event will occur, and we do not know its implication. Because of this we do not know the dynamic context items' state in the future. The static context item's state is constant during the whole time span. The blue vertical arrow shows how the complete context of that specific point in time includes the predicted context timeslot that is currently active, the current state of the dynamic context items and the constant state of the static context item.

## **2.7 Use of Personalization and Adaptivity**

As previously discussed there is a great variation in the students' characteristics and their preferred approach to learning a language. It is therefore impossible to say that one way of teaching is better than another – it is ultimately up to what style that suits and benefits each individual student. In typical classroom teaching the problem is that

one teacher can not adapt his teaching style to cover all the students' different needs. Usually he will have to decide on some sort of compromise that works well for most of them. However, if the teacher only had one student he could completely adapt to this student's needs. By using personal mobile devices for language learning we can try to mimic some of this one-on-one relationship. The teaching methods employed by the application can be personalized and adaptive to the student's preferences and skills and also to the surrounding environment. This will give each student a unique learning experience that is specifically adapted to suit his needs.

Adaptivity has previously been employed for non-mobile language teaching system. The *German Tutor* [26] is a web-based adaptive tutoring application. Their argument for making the application adaptive is to be able to serve the vast diversity of the users. The German Tutor has a student model that keeps track of the users' score for three categories corresponding to three learning levels: beginner, intermediate, and advanced. The system can create three categories of feedback and uses this to select an error response suited to the student's expertise. The central idea is that within a framework of guided discovery an expert requires less detailed feedback than a beginner learner. To be able to figure out what the student did wrong the system has an error checking engine that checks for the following types of errors: Punctuation errors, spelling errors, task errors, missing word errors, extra word errors, word order errors and grammar errors. The error evaluation is also used to categorize the students' skill level dynamically as they use the system.

*Task-based Adaptive learner Guidance On the WWW* (TANGOW) [27] is another adaptive web-based tutoring system. The creators of the system argue that adaptive teaching systems are necessary because not all students have the same skills for learning a concrete subject. Some of them may need more explanations than others, and they may have previous knowledge about the subject or not. In addition to these important differences their tutoring system also takes personal features, such as age, interests and preferences, into consideration when presenting the content. The TANGOW system is created to facilitate the construction of adaptive learning environments for the web. It guides the students during their learning process by registering their behavior and profile. The curriculum sequencing for each student is generated dynamically, so the same concepts may be taught in different ways. The generation of content depends on the student's profile and his actions while interacting with the course. TANGOW uses three different mechanisms to achieve adaptivity. The first one is task decomposition. Based on the student profile a task can be decomposed into subtasks (theory, exercises and examples) that are tailored specifically to each student. The second way of implementing adaptivity is by including rules that check for preconditions related to other task achievements. This can for example be used to make a task locked until you have completed a set of other tasks. This customizes a student's experience by presenting different tasks based on his previous actions. The third and final mechanism is the adaptability in the building of HTML pages. Different media elements appear on the web pages based on some of the settings in the students' profile, i.e. student language and age. Depending on the student's age the same concepts can also be explained in different ways by using words of different difficulty. Students can also choose their preferred learning strategy between theory and practical examples. This is used to decide what to present first when they start their learning sessions. The system is designed to allow new learning strategies to be added later, and it is possible to change the active strategy during the

learning session. This feature will hopefully allow improved guidance for the students.

To be able to make an application adaptive and personalized we need some knowledge about the user and his current surroundings. We simply need context information. Based on the current context and context changes we can adapt the appearance and content of our application. This will allow the application to be personalized to each learner's individual needs and preferences. The key context attributes we need can be divided into two different categories: profile parameters and environment parameters. The profile parameters are attributes that are stored in each user's profile, such as age, native language, interests etc. The environment parameters reflect the physical situation the user is in by providing information about location, weather, the computing device that is being used etc. These parameters can either be static, such as age and native language, or they can be dynamic, such as location and date and time. Some of the static parameters do however change, but they do it so seldom (i.e. age) that I have chosen to regard them as static. There is also variation in how dynamic the dynamic parameters are. Some are highly dynamic and change continuously, such as location, while others change less frequently, for example the user's skill level. The concept of profile and environment parameters is visualized in Figure 4. It shows how we can try to get complete personalization data by combining context information from the student's profile and from his environment.

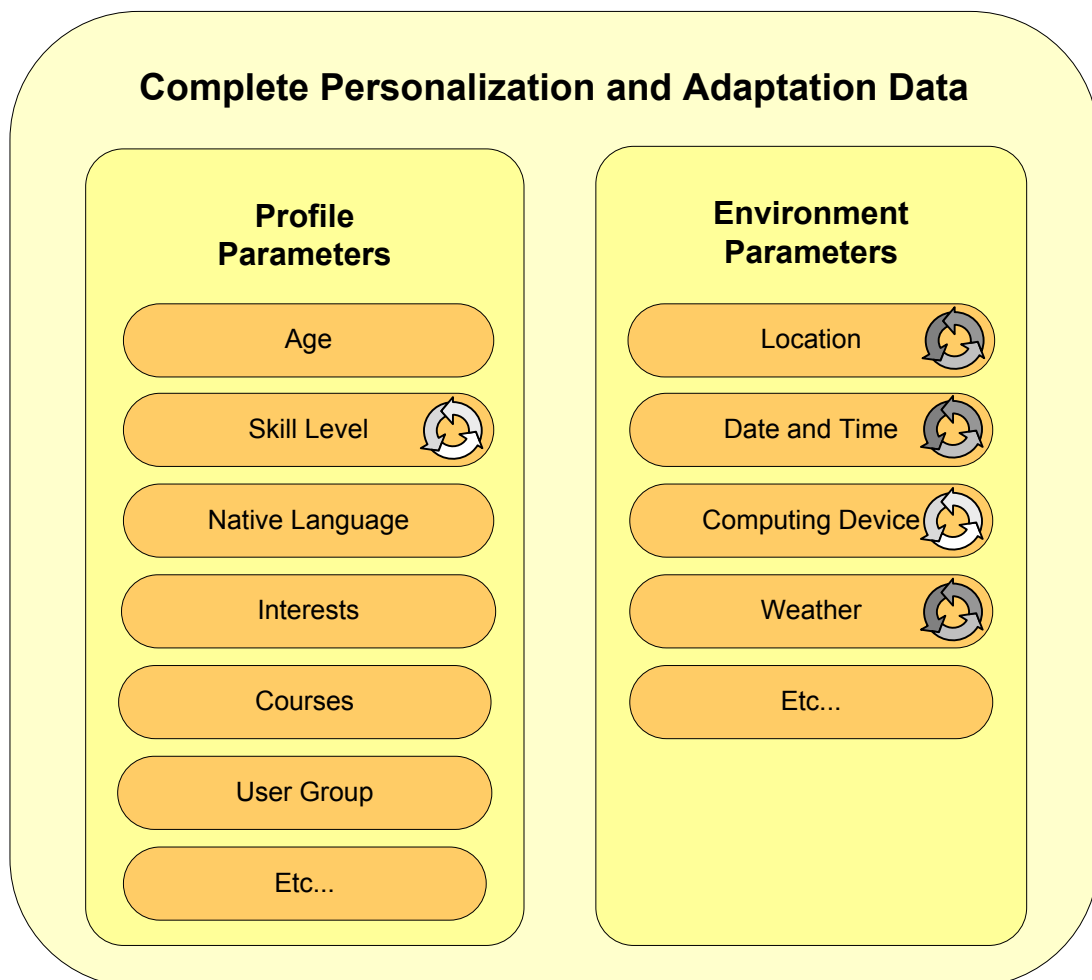


Figure 4: Dynamic parameters are marked with a circle of arrows. The

## **2.8 Summary**

In this chapter I have researched theory concerning language teaching and the use of mobile devices for this purpose. I started by addressing research question 1 by looking at different approaches to language teaching and different classifications of the learners intelligence types. I then tried to use this information to give a theoretical-based answer to research question 1.a. The answer to this research question will be further detailed as I develop a prototype of such a mobile application. Next I targeted research question 2 by exploring the new possibilities mobile devices can give to language learning. The main advantages seem to be the whenever, wherever access to learning materials and that it allows situation-based training in real-life scenarios. I also researched guidelines for language teaching software (research question 2.a). These guidelines, together with the derived language teaching requirements for PALLAS, will be used during the development of the PALLAS system and help us avoid unnecessary mistakes. The possibilities and implications of using mobile devices presented in chapter 2.4 (research question 3) provide useful information for the design of the PALLAS system architecture. The hardware issues presented in that subchapter will provide material for a lot of the design decisions that will have to be made.

The rest of this chapter dealt with the concepts of context and adaptivity. I started by looking at what features existing mobile language learning applications offered. This will be used as an inspiration and starting ground for coming up with ways to take advantage of context in the mobile language learning application (research questions 4 and 4.a). Then I researched the concept of context and different categorizations of context. The exploration of phases of context resulted in a model of context that will help me in the development of a context engine for PALLAS (research question 4.b). The model includes static, dynamic and predicted context items and describes how they change over time. With regards to the use of adaptivity, the main argument seems to be the variation in learners. By making PALLAS adaptive the client can be automatically personalized to each individual learner's needs and current situation (research question 5). But to be able to make the application adaptive we need some information about the user and his surroundings, i.e. context information. To make this context information easier to feed into a software application I have created a model consisting of profile parameters and environment parameters. The profile parameters contain information about the user, such as age and interests, and the environment parameters contain information about the external properties of the user's situation, such as location and weather. The combination of these parameter types should provide adequate context data to make PALLAS truly adaptive.

## **3 User Scenarios**

### **3.1 Introduction**

A user scenario is a great way to quickly explain how a feature or system would work in the real world. It is a narrative that describes a set of users, a work environment and a set of tasks that the users perform [28]. User scenarios are great when you want to show a concrete vision of how human activity interacts with the proposed technology. A good scenario should realistically describe human activity, but it also needs to be believable, relevant to business needs and simple. Simplicity is key – the scenario should be a quick way to understand the big picture of the system. Scenarios are particularly valuable at the beginning of a project to help share the vision and understand the needed functionality. All of this makes it reasonable to start the work on the system’s architecture by writing some user scenarios. The user scenarios can then be used to devise more precisely how the system is supposed to work, and thus the needed systems components. To cover the system’s functionality I have written two scenarios describing different usage situations. The first scenario shows how a foreign language student could use the system while still in her home country, while the second scenario shows a foreign language student that is traveling to a country to learn the local language.

### **3.2 User scenario 1 – Language Student in Her Home Country**

Nina is a student in Trondheim currently taking a beginner’s class in German. One day on her way home from school she passes by the downtown farmers market. It strikes her that she doesn’t know a lot of German names for vegetables, so she pulls out her Smartphone and starts the language learning application. She selects German and enters the keyword “grønnsaker”. Nina is presented with a short text in German describing different vegetables. The difficulty level of the text is based on Nina’s profile in the language learning application, which has been built automatically based on Nina’s previous interactions with it. After she has finished reading the short text Nina chooses to look at the glossary connected to the text. The glossary contains the German names and pictures of a bunch of different vegetables. When she is done reading through the glossary, Nina selects the “Glossary test” option. This starts a test based on the glossary words. Nina didn’t score to well, but she doesn’t have time to take the test over again right now. When Nina logs on to the web based language learning system on her desktop computer later that day, she is notified that she should practice more on the vegetable glossary. She clicks on the link to the vegetable glossary and practices some more before she tries to take the test again.

Nina enjoyed the German class so much that she decided to continue with an intermediate class the next semester. She has also started to watch German movies to learn spoken language better. One day while passing by the Nidarosdomen cathedral her location aware Smartphone started to beep. Nina took it out of her pocket and saw a notification from the language learning application telling her that Nidarosdomen has a guided tour in German starting in 10 minutes. Nina quickly joined the tour which also allowed her to come in contact with a lot of German tourists she could

practice her German on. At anytime that Nina had problems understanding a German word she simply queried the language learning application's built-in dictionary.

### **3.3 User Scenario 2 – Language Student in a Foreign Country**

Fredrik studies French at the university in Trondheim. His teacher, Mr. Belmas, firmly believes that to learn the language they have to use it in a real-life setting. That is why they are going on an excursion to Cannes in France.

On his previous trips to Cannes Mr. Belmas would give the students a paper-based list of historical places to visit. The list included some information about each site and some tasks to solve. This year however, the students will be using PDAs with a new language learning application on their excursion. Before the trip Mr. Belmas uses an administrative application to create the content for the language learning application. This allows him to add customized content, like historical information, explanations of special words, quizzes and tasks. He can also select to include ready-made content from the text book company's content management system. Mr. Belmas uses this to include relevant chapters from the text book his class is using, and he also includes a Norwegian-French, French-Norwegian dictionary. Finally Mr. Belmas context-links some of the content he has put together. In the administration application he simply adds regions to a map of Cannes, selects the content he wants to be linked to that specific region and the time span that the content should be available. The day before the excursion to Cannes he synchronizes all the PDAs with his customized content and packs them in his bag.

When the class arrives in Cannes, Mr. Belmas hands out one PDA to each two-person group. The students then proceed by entering their username and password to start using the PDA application with their user profile activated. Fredrik is sharing his PDA with Nils. After they have logged on they are presented with a list of tasks they are supposed to complete today. The tasks include visiting three historical sites or museums, buy fruit at Marche Forville (market) and go to a restaurant and order dinner in French. They start off by looking up Marche Forville on the PDA and get presented with a short text in French explaining what it is and also a map of where it is located. Because the PDA is connected to a GPS-unit they can also see an indicator of their current location on the map. On their way to the market the PDA starts to beep and shows a notification telling them that they are next to Musee de la Castre, and that museum currently is open. They read the short museum description on the PDA and decided to visit it. To have this visit count as a museum visit they have to fulfill a list of tasks presented on the PDA, i.e. find specific items in the museum. When they find the items they take pictures of themselves in front of them using the PDA's built-in camera to prove that they found them.

Later the same day Fredrik and Nils return to their hotel to rest and prepare for dinner. While waiting for Fredrik to get ready Nils looks up the word "restaurant" on the PDA. The application shows that the word produces results from the text book, dictionary and quizzes. Nils selects the link to the text book chapter and is presented with information about French restaurant culture and a dialog where a person orders food at a restaurant. The text book chapter also has a link to a quiz that is connected to this chapter, but because Fredrik is ready to go Nils does not have time to take it. At the restaurant they order the food in French as planned – Nils even gets to use some of



the new phrases he just learnt. By looking up the word “escargot” on the PDA’s dictionary Fredrik also avoids ordering snails for dinner. To finally get the “order dinner in French”-task completed they have to complete a quiz on PDA. When their score is satisfactory the task is marked as completed.

After a busy and interesting week in Cannes the class excursion has come to an end. Before leaving the hotel for the airport, Mr. Belmas collects the PDAs from all the students. Later when the class arrives at the airport Fredrik wants to buy an ice cream. Unfortunately he does not remember the French word for ice cream and he has already handed in the PDA to Mr. Belmas. Fortunately Fredrik has his cell phone with him. He sends an SMS containing “ice cream” to a number provided by the language learning application developers. The server that receives the SMS recognizes the sender’s cell phone number and automatically matches the word lookup to Fredrik’s profile. A reply customized for cell phone messages is sent back to Fredrik. It includes the word ice cream in French, and because Fredrik’s profile says that he is a beginner it also includes a short explanation of how one would order an ice cream.

When Mr. Belmas gets back home he re-synchronizes the PDA’s with the language learning server. Using the administrative application he can see how all the students did at the different tasks. The students’ can access their results through a special web page, and they are also presented with suggestions on areas they should improve.

### **3.4 Summary**

This chapter deals with the first step of the scenario-based approach for creating the PALLAS prototype (see Figure 2), which is the writing of user scenarios. The two user scenarios I have written will be analyzed in the next chapter. This analysis will provide feature and functionality requirements for the PALLAS system. The creative process of creating ideas and functionality to put in the user scenarios addressed the research questions dealing with new ideas for use of context and adaptivity, such as questions 4, 4.a and 5.

## **4 Scenario Analysis**

### **4.1 Introduction**

I have now created user scenarios that explain possible situations of the PALLAS system in use. The next step in my scenario-based approach is to transform these user scenarios into formalized systems requirements. These system requirements should cover a more general use than only the specific situation in the user scenarios. To document these requirements I will start by defining the users of the system. Next I will analyze the context needs of PALLAS. To capture the functionality and use of the system I will utilize use cases. These diagrams should clearly convey the possibilities and limitations of PALLAS, and can be used to make sure that I have covered all the needed functionality. Finally I will write a functionality check list. This checklist should cover all the functionality needed to perform the tasks described in the user scenarios. The checklist will later be used to verify that the prototype meets the needs of the software envisioned in the user scenarios.

### **4.2 Defining the Users**

Based on the user scenarios we can define four main user groups; the language students, the teacher, the content providers and finally the service provider.

The language students are the main users - the other users primarily use the system to be able to provide services for the language students. The students mainly use the language learning system through mobile devices so that they can query it for information whenever they want and wherever they are. This is meant to enable them to use the foreign language in more real-life situations. The language learners can also access the system by using a normal web browser, i.e. at home or at school. This way they can get access to all the learning materials without even having a mobile device. Each language student has a personal profile that keeps track of their settings, skill level, quiz results etc.

The second user group is the teachers group. They act as administrators for their courses and can select which users to include in their group. The teachers can select what content to make available for the language students in their courses and add customized content. They can also keep track of the students' progress and use of the system, i.e. through quiz results etc. All this is done through an administration interface to the language learning system.

The content providers use the system to share content such as text book material, quizzes etc. The language learning system can hook up to the publishing houses' existing content management system, and thus act as an extension for content delivery.

Finally someone has to grant the teachers and content providers access to the language learning system. This is where the service provider comes into the pictures. The service provider is responsible for running the system, which includes setting up and running the central server, grant teachers and content providers access to it and

customize communication modules to the content providers' content management systems.

Besides the user groups previously mentioned there are several other end users who might find such a system useful. Tourists could use it as a guide and translator and history students could for example use it for field trips. This also adds new potential user groups in form of content providers, such as tourist offices and museums. All of this is out the scope of the research project, so I will only focus on the use of the system for foreign language learning purposes. The system architecture does however allow this kind of use without any significant changes besides providing different content.

### **4.3 Analysis of Context Needs**

There are many types of context which could be interesting to use in the language learning system, but unfortunately there are technological boundaries we need to take into consideration. It would for example be very nice if the application could monitor the user's mood and then adjust the difficulty of the exercises, so that they build up the user's self esteem when he is feeling down, and challenges him when he is feeling confident. With today's technology this would be very hard to achieve, especially in an unobtrusive way. Having all kinds of monitors connected to your body would definitely make the user experience less pleasurable. However, there are also a lot of ways to harvest context information that are fully possible with today's technology.

By analyzing my user scenarios I have come up with several different concrete uses of context for PALLAS. My findings are here presented based on their context category (as explained in chapter 2.6.1):

- **Spacio-temporal context:** Date and time of day can for example be used to determine whether to launch alerts or not – an alert about an interesting museum should only be launched if the museum is open. Location information should also be used to trigger alerts about nearby points of interest. The user's location can be automatically updated by connecting the mobile device to a GPS (Global Positioning System) receiver or by using GSM (cellphone) positioning.
- **Environment context:** The client application should adapt to type of device the user is utilizing. If the user for example has an advanced PDA with a decent amount of storage space, the application can use caching of data to minimize or eliminate network traffic. The client application's user interface should also customize to the different devices' input and output possibilities. Large and colorful graphs showing a student's progress makes sense on PC, but showing the test scores in numbers would be more suitable for a cellphone screen. The mobile client application could additionally be adaptable to its current internet connection type. If the client uses WiFi, speed and cost are not any issue, but if the client is out of WiFi-access range and uses GPRS (cellphone) it should try to minimize the use of internet access. This is because GPRS access is relatively slow and you usually have to pay for the amount of data that is transferred, while WiFi access on the other hand is a lot faster and either free or has a fixed price. Weather information is another kind of environment context that can be useful for PALLAS. It can be used to make

better decisions when giving students suggestions on things to do. If it is raining it can for example suggest going to a museum instead of outdoor sightseeing. Providing more relevant suggestions will hopefully improve the user experience. Weather information could be retrieved through hardware monitoring devices connected to the mobile unit, but it would be costly and add bulk to the unit. It is much easier to simply download current weather data from the internet as this approach does not need any additional hardware. The location information that PALLAS already has can be used to get weather information for the user's current location. There are many free services, such as [www.weather.com](http://www.weather.com), that offer updated weather information and even weather forecasts. The amount of data that is needed to transfer this information is minimal. Forecast data can be used if the user does not have an internet connection at some point. However, it is important to be careful when using weather forecasts because they are only predictions about what might happen. Giving a completely wrong suggestion can make the user stop trusting PALLAS.

- **Personal context:** The language learning system can tap into the user's mental context. This could be done by letting each user have their own personal user profile. The profile should be created the first time the user logs onto the system by having him answer a questionnaire. This gives the system some insight about the user's interests, knowledge level and other properties. The profile would then be automatically updated as the user interacts with the system. If the user does well on an exercise the profile will be updated to reflect the user's improved knowledge level. This allows the application to dynamically adapt to the user's skills. The profile could also keep track of what kind of learning style the user prefers.
- **Task context:** The client application for mobile devices can adapt to the tasks the user is performing. The user would have to manually tell the application that he is performing a task. One example of this is when a student is on an assignment, such as "visit museum X", he can then tell the application that he has arrived at the assignment location. If he is required to document the visit the application will adapt and make the "take picture"-feature easily available.
- **Social context:** The user profiles can be used to support different user roles. The features available to a user when utilizing the system should differ based on what user group he belongs to. What user group he belongs to is decided based on his login information. The possibility to share context information (location, profile information etc.) between users could provide many interesting scenarios. This could for example be used to add a social dimension by providing information about where your peers are and what they are doing. Unfortunately this is out of the scope of this project. The system architecture is however designed to allow the addition of such features in future work. The sharing of such information could impose on the users' privacy, so it is important that they need to approve to having their information shared. They should also be able to turn off sharing at any given time.

There are many interesting uses of context information for PALLAS. All of the suggestions given here are chosen because they add useful information while still being highly feasible with today's technology. Standard cellphones or PDAs with a

GPS module should be able to handle the collection and use of all these context types. A summary of the suggested context use in PALLAS is given in Table 4.

<b>Context Type</b>	<b>Context information</b>
<i>Spacio-temporal</i>	<ul style="list-style-type: none"> <li>• Date and time of day</li> <li>• User location</li> </ul>
<i>Environmental</i>	<ul style="list-style-type: none"> <li>• What device is being used</li> <li>• Type of internet connection</li> <li>• Weather</li> </ul>
<i>Personal</i>	<ul style="list-style-type: none"> <li>• User's static properties (age, native language, interests, courses etc.)</li> <li>• User's knowledge level</li> <li>• User's preferred learning style</li> </ul>
<i>Task</i>	<ul style="list-style-type: none"> <li>• What task is being performed</li> </ul>
<i>Social</i>	<ul style="list-style-type: none"> <li>• User group (role)</li> <li>• <i>Future</i>: Peers' location and task information</li> </ul>

**Table 4:** Summary of what kind of context information the language learning system uses

The use of context is a central issue for the mobile PALLAS clients. There is therefore a need for a context engine that can act as central repository for context data and context changes. Context dependant features could then simply hook up to this context engine and subscribe to be notified of changes. For some of the context properties it might be interesting to keep track of previous values. Previous location data can for example be used to figure out where the user has already been. The context engine should therefore support storage and retrieval of context history. In addition to history it should support future context. This will be needed for context properties with predicted values, such as weather information. Because the predications are not completely certain there should be a probability algorithm that calculates how reliable the data is. For weather information it will make sure that forecasts that are made about the very near future are given more weight than long-term forecast.

#### **4.4 Use Cases**

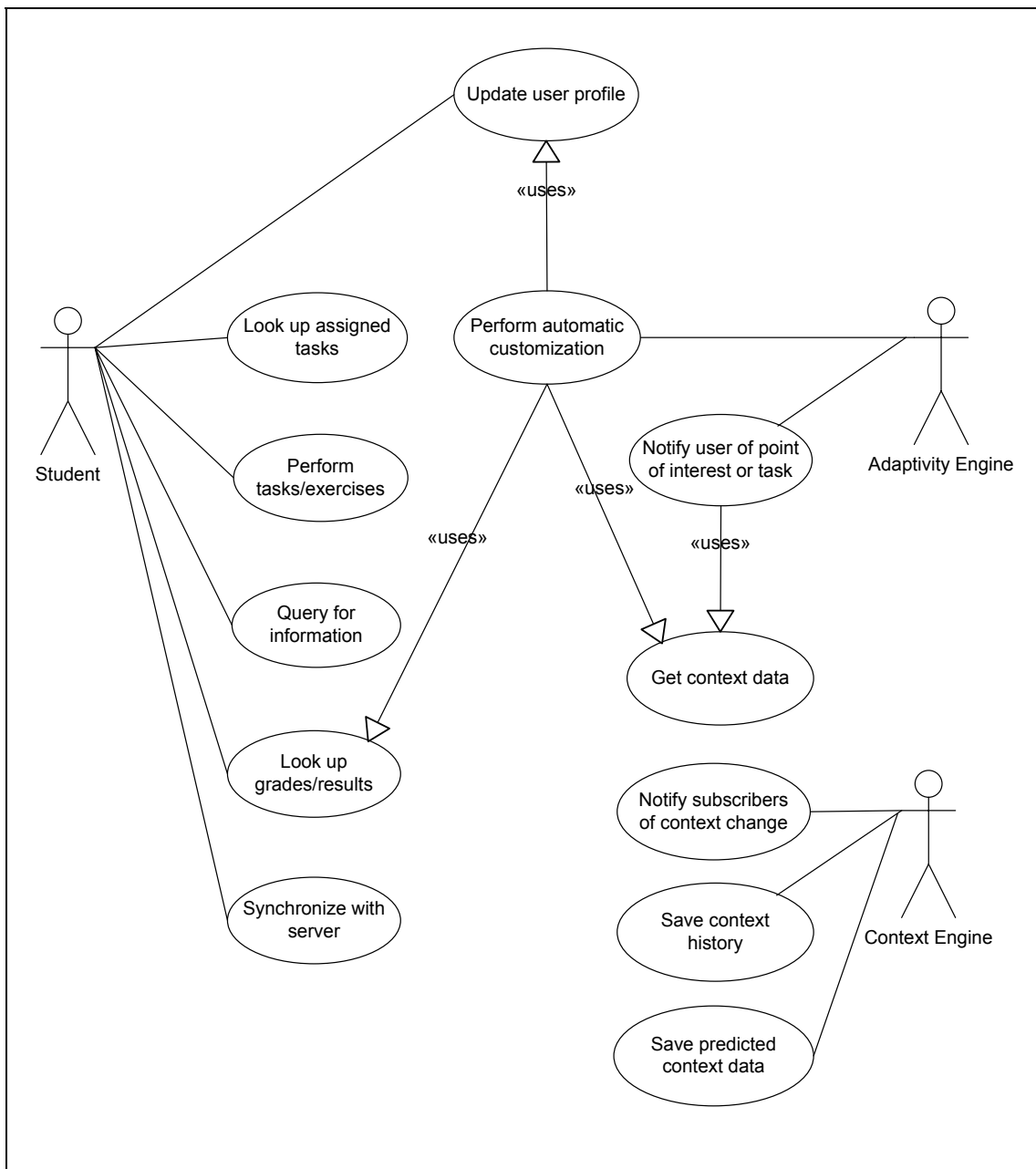
Use case diagrams are a way to represent an external view of a system [29]. They can be used to show the interaction between the system and actors (the users) [30]. A user scenario only shows one possible outcome of system interaction, a use case on the other hand is a combination of scenarios and should include all the alternative paths. This does not mean that you should include everything; it suffices to include the things that really help to clarify the interactions. I have chosen to include use case diagrams because they are a great way to show how the users and system interact, and this makes them useful for documenting the requirements of the system architecture.

My analysis of the user scenarios have resulted in three use case diagrams. The first one covers how PALLAS works when the students use the system, the second one shows how teachers interact with PALLAS and the last one shows how the service

providers use PALLAS. The content providers are not included in the use case diagrams because they do not interact directly with the system.

#### 4.4.1 Use Case 1 – Student using PALLAS

The use case diagram in Figure 5 shows how the student interacts with the PALLAS system. In addition to the Student actor, which is based on one of my defined users, I have included two actors who come into play when the student uses PALLAS: the Adaptivity Engine and the Context Engine. These are both system components used to make PALLAS adaptive to the user and the context.

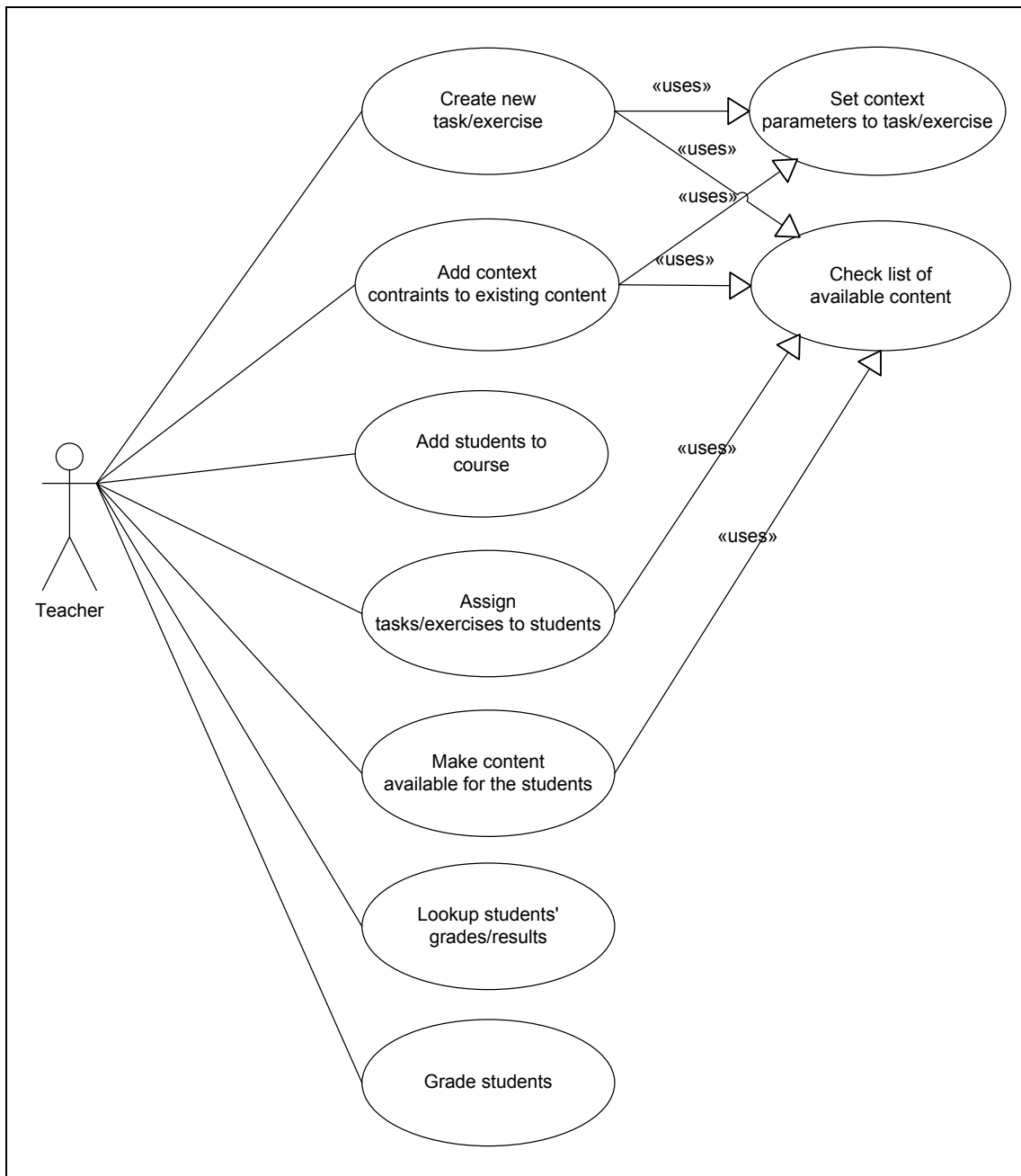


**Figure 5:** Use Case diagram showing system operations that are relevant when the student uses the PALLAS client

The Context Engine keeps track of previous and current context data and when there is a context change he checks if he has any subscribers interested in that context change. If he does he will notify them of the changes. The Adaptivity Engine is one example of a service that will subscribe to context changes. If the changes activate a contextualized trigger the Adaptivity Engine will notify the user of the point of interest or task. When checking a context trigger it might also be necessary to look up more context data than for just the context type that changed. The Adaptivity Engine can also perform automatic customizations based on context data and the student's results. The changes are stored in the student's profile for later use. The Student can also update his profile manually, for example to change what his interests are. In the student's everyday use of the PALLAS client he will do things like look up assigned tasks, perform tasks and exercises, query for information, look up grades and results and synchronize the client with the PALLAS server.

#### **4.4.2 Use Case 2 – Teacher using PALLAS**

Figure 6 is a use case diagram of the teachers' use of PALLAS. The teachers mainly use PALLAS to facilitate the students. They can create new tasks and exercises for the students, and add context parameters to trigger the tasks or exercise. The teachers can also link to content made available from the publishers' content management system from the new tasks or exercises. Content imported from the publishers may not have any context triggers, or it may simply lack some triggers that are important for the teacher's intended use of the content. The teacher can overcome this by adding context constraints to content manually. Some of the more administrative tasks the teachers can perform are adding students to their course (so that they can use PALLAS), assign tasks and exercises to students and select what content that should be available to the different students. When the students start solving exercises and tasks their scores are stored automatically. The teacher can look up these scores and see how their students are doing. This can be used to see if they have a negative or positive trend in their learning experience. The teachers can also manually grade each student.



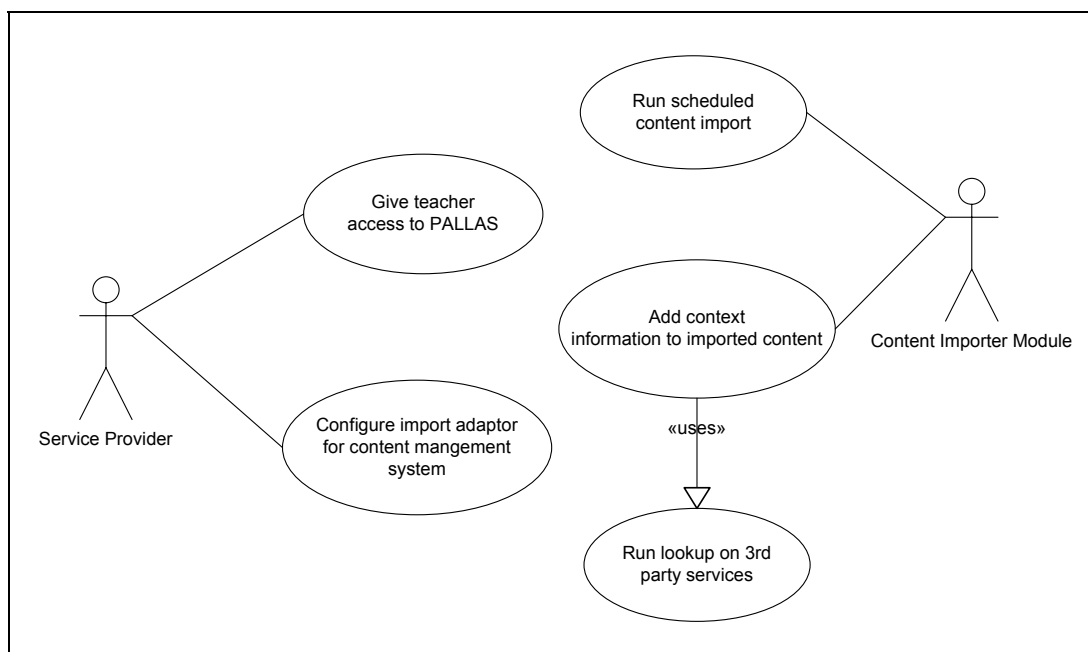
**Figure 6:** Use Case diagram showing how teachers interact with the PALLAS system

#### 4.4.3 Use Case 3 –Service providers using PALLAS

The PALLAS service provider is responsible for hosting and running the PALLAS server. When interacting with the PALLAS system he performs two main operations. The first one is to give teachers access to PALLAS so that they can use the system, and the second one is to configure the adapters that makes it possible to import content from different content management system into PALLAS. Each publishing house might have a different content management system, so the service provider needs to configure an import adaptor for all of the different content management systems' interfaces. In Figure 7 there is also another actor; the Content Importer Module. This module takes care of importing content from the publishing houses'



content management systems to the PALLAS server. It does this by running scheduled (set by the service provider) content import. Each configuration set by the service provider will be used when connecting to the different content management systems.



**Figure 7:** Use Case diagram of the Service Provider’s interactions with PALLAS

When the content has been downloaded from the publishers it is converted into a suitable format for storage on the PALLAS server. The Content Importer Module can further try to add context information to the imported content. To be able to do this the module can utilize different third party services, for example can Microsoft MapPoint be used to map keywords to a location. The teacher can also add context information to the content later manually (as shown in Figure 6).

## 4.5 Content Modules

A central part of PALLAS’ functionality is to provide students with context sensitive and customized language learning content. To be able to do this we need two things. The first one is obviously to keep track of the students’ previous and current context. This information is necessary for PALLAS to be able to react to the students’ situation in a reasonable manner. But to be able to react we need something that tells us when to react and how to react. This is our second need – contextualized language learning content. The content can either be created by the teachers or it can be copied from the publishing houses’ content management systems (CMS). The language learning content from the publishing houses will however usually not contain any contextualized information. The PALLAS’ content module should therefore allow the addition of context triggers to ready-made content. Some context triggers can be added automatically by doing lookups on third party services based on keywords from the content. But to get highly customized context trigger it is recommended that teachers add them manually to the content they want to use. The added context trigger data can be made available to all teachers, so it only has to be added once.

In my prestudy I have argued for that there is not one single best way of learning a language. Different students will have different objectives and preferences in their learning. Because PALLAS is supposed to be adaptable to each student it needs to support different teaching methods. The approaches to language teaching covered in chapter 2.1 (*Approaches to Language Teaching*) will therefore need to be supported as explained in chapter 2.1.1 (*Where does PALLAS fit in?*). The PALLAS content system should also follow the guidelines I presented in chapter 2.2 (*Guidelines for Designing Language Teaching Computer Programs*) extensively.

Based on these requirements for the context sensitive and customized language learning content in PALLAS I have created several content modules. The modules and the relationship between them should allow PALLAS to provide the functionality presented in the prestudy and user scenarios. Figure 8 shows all the modules and their relationships. The language learning content is stored in modules on the lower part of the figure. The Multiple Choice Quiz and Word Fill-in modules are used to store these kinds of exercises. The Text module is used to store formal texts, such as chapters from text books etc. Each text can have a link to a corresponding glossary training exercise that contains words from the text in question. The Glossary Training module stores the different glossary tests and it uses the Dictionary module to lookup the correct translation of words. The Dictionary module is also used to perform normal dictionary lookups in PALLAS. These exercises can be accessed directly in PALLAS if the student wants to perform a certain type of exercise, but they can also be accessed through tasks. The Task module is used to create tasks that the students can perform. These tasks can range from real-life situation tasks, such as “buy a stamp at the post office”, to homework type tasks which include reading certain texts and learn the corresponding glossary. This is possible by linking the task to different content. A task can include multiple choice quizzes, word fill-in exercises, reading of texts and glossary training. The task can also link to content that is not part of the task but still might be useful for solving it or it might be relevant in some other way. A task can link to texts, words in the dictionary and point of interest information. The Point of Interest Info module is used to store information about places, such as tourist attractions, buildings, restaurants etc. This can be useful for field trips or excursions. The teacher can for example create a task that tells the students to visit three of Rome’s churches. This task can then be linked to point of interest information for some of Rome’s churches. The task can also link to a relevant text and maybe some useful words from the dictionary.

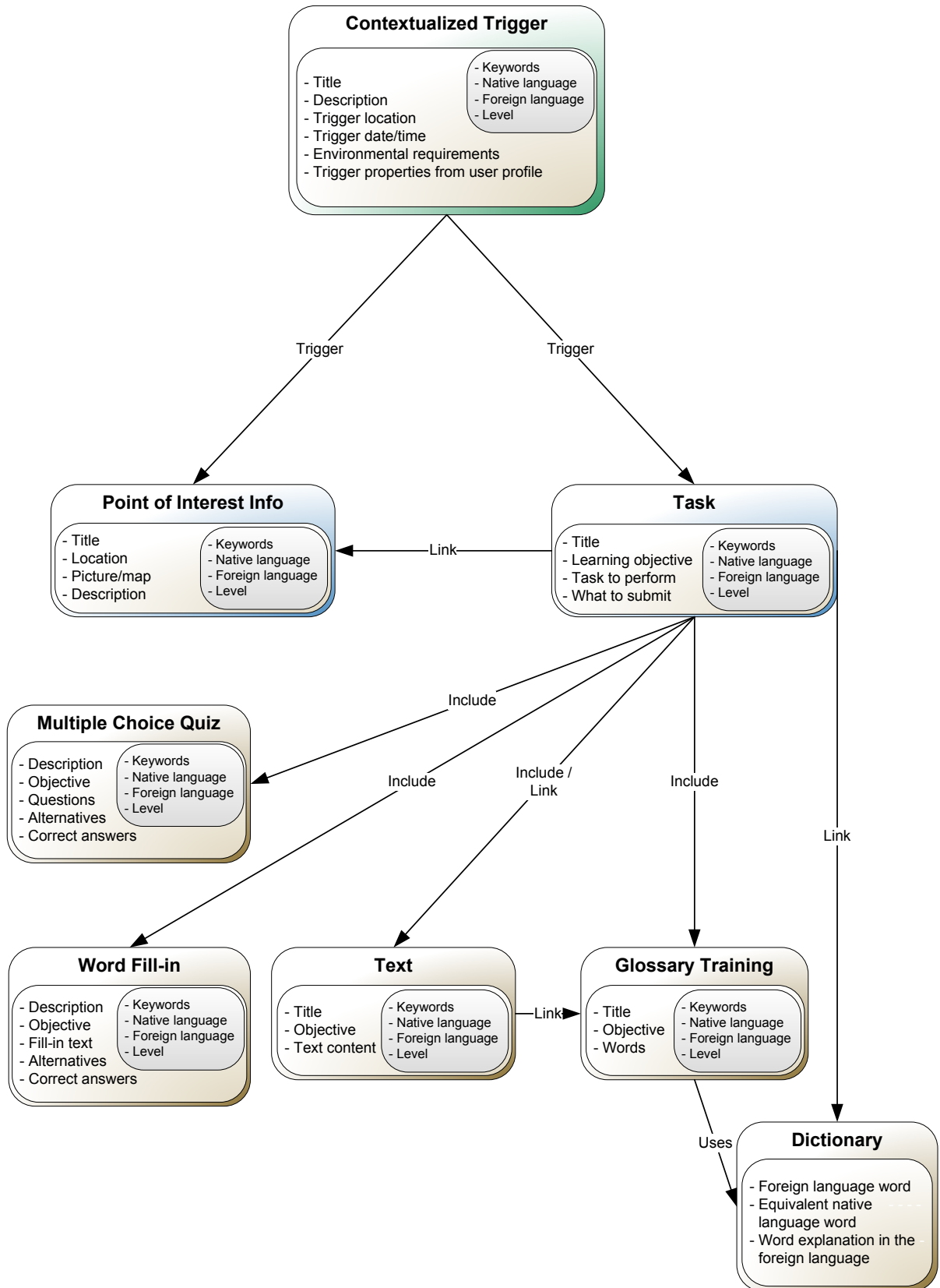


Figure 8: PALLAS' different content modules and their relationships

The last module is called Contextualized Trigger. This module is responsible making PALLAS react on context changes. The triggers are activated based on the user's location, data and time of day, environmental properties and properties from the user's profile. The trigger can force two different actions. The first is to show point of interest information. This is for example useful to make students aware of nearby locations that might be worth visiting. The second action a trigger can cause is to display a task. This can for example be used to make a task pop-up when the student arrives to one of Rome's churches. Each church can have their own trigger linked to different tasks. There can even be several triggers for the same church (location), but they can differentiate on the student's profile. This way advanced students can get different tasks than beginners when they reach the same location. All the language learning content is made available to the contextualized triggers because they can link to tasks. Context information is however not only used for the contextualized triggers. It can also be used by each content module to make the content adapted to the user. The dictionary can for example use context information about the user's knowledge. If the student is advanced he can be presented with a word explanation in the foreign language when he looks up a word (with the option to also lookup the translation of the word). A beginner on the other hand would get the translation from the foreign language to his native language up front.

The language learning content can also be looked up by entering a simple query. This can for example be useful in real-life scenarios, such as when you are eating out at a restaurant and you want to order using the foreign language. By doing a query on i.e. "restaurant order" PALLAS will give a list of all content relevant for that situation, whether it be a chapter from a textbook explaining how to order, or a glossary training test with restaurant related words. In Figure 5 you will notice that most of the content modules have a set of common attributes (the inner box with a light grey background). The first attribute, Keywords, makes it possible to do accurate query lookups. The rest of the attributes are needed to make the content adaptive to the current user. They make it possible to check what content is applicable to by comparing the attributes to information in the user's profile. Another important attribute that most of the content modules have is the Objective attribute. As explained in the prestudy it is important for language learning content to be clear on what it is trying to teach, and how that is supposed to be achieved. This attribute is meant allow this information to be stored. The rest of the content modules' attributes are mostly self-explanatory. They are needed to store all the data required to make things like the different exercises work. In some cases this will have a double effect and also make sure we are following the guidelines in chapter 2.2. The correct answers for a multiple choice quiz are for example both needed for PALLAS to be able analyze how well the student did (make sure the quiz works) as well as to be able to give feedback to user responses (one of the guidelines).

#### **4.6 Functionality Check List**

This scenario analysis process is a step in the direction of the goal of creating a working PALLAS prototype. To make sure that the prototype is working satisfyingly I want to test it against the functionality proposed in the user scenarios. To make this process easier I have created a functionality check list. The complete list covers all the functionality I think is required for the prototype to work satisfyingly according to the user scenarios. The check list is hence also an analysis of the user scenarios from

chapter 3. However, because I am developing a prototype and not the full system I will of course not require absolutely all of the functionality to be included (you can read more about the limitations of the prototype in chapter 6.5). With the functionality check list I can compare the prototype to the user scenarios by simply starting at the top of the list and work my way through the complete list by checking if each feature is supported or not. The higher the percentage of functionality tests that pass, the more complete the prototype is, and thus more ready for testing with real teachers and students.

The complete functionality check list can be found in Appendix A. It is divided into two main sections, one for each of the main users portrayed in the user scenarios: Students and teachers. Each section includes a table with a row for each functionality that needs to be tested. The table is also divided into three columns: Functionality from scenario, Availability in the PALLAS prototype and finally a column to mark if the feature passed the test or not. The functionality check list will be used in the evaluation of the PALLAS prototype in chapter 7.

#### **4.7 Summary**

In this chapter I have performed the second step of my scenario-based approach and created the requirement models for the PALLAS prototype. I started by defining the users of the PALLAS system based on the user scenarios I wrote in chapter 3. Next I performed an analysis of the system's context needs, which included a summary of all context types that are useful and possible to implement in the prototype (research question 4.a). With all this information at hand I could start creating some generalized models of how the PALLAS system should work. I did this by constructing use case diagrams that showed all the important system functionality and the actors of the system. The PALLAS system needs to handle a lot of language learning content. The content needs to be adaptable to the users and their current context. To make this possible I created a set of content modules and established links and relationships between them. This should make it possible to provide the functionality presented in the user scenarios. Finally I created a functionality check list that later will be used to verify that the prototype works according to the PALLAS specifications. In comparison with Figure 1, this and the previous chapter have dealt with the second part of my research method – the scenario writing and analysis has provided me with an analysis of the prestudy materials, which I further used to generate new ideas and solutions for a mobile context-sensitive foreign language learning application.

## **5 System Architecture**

### **5.1 Introduction**

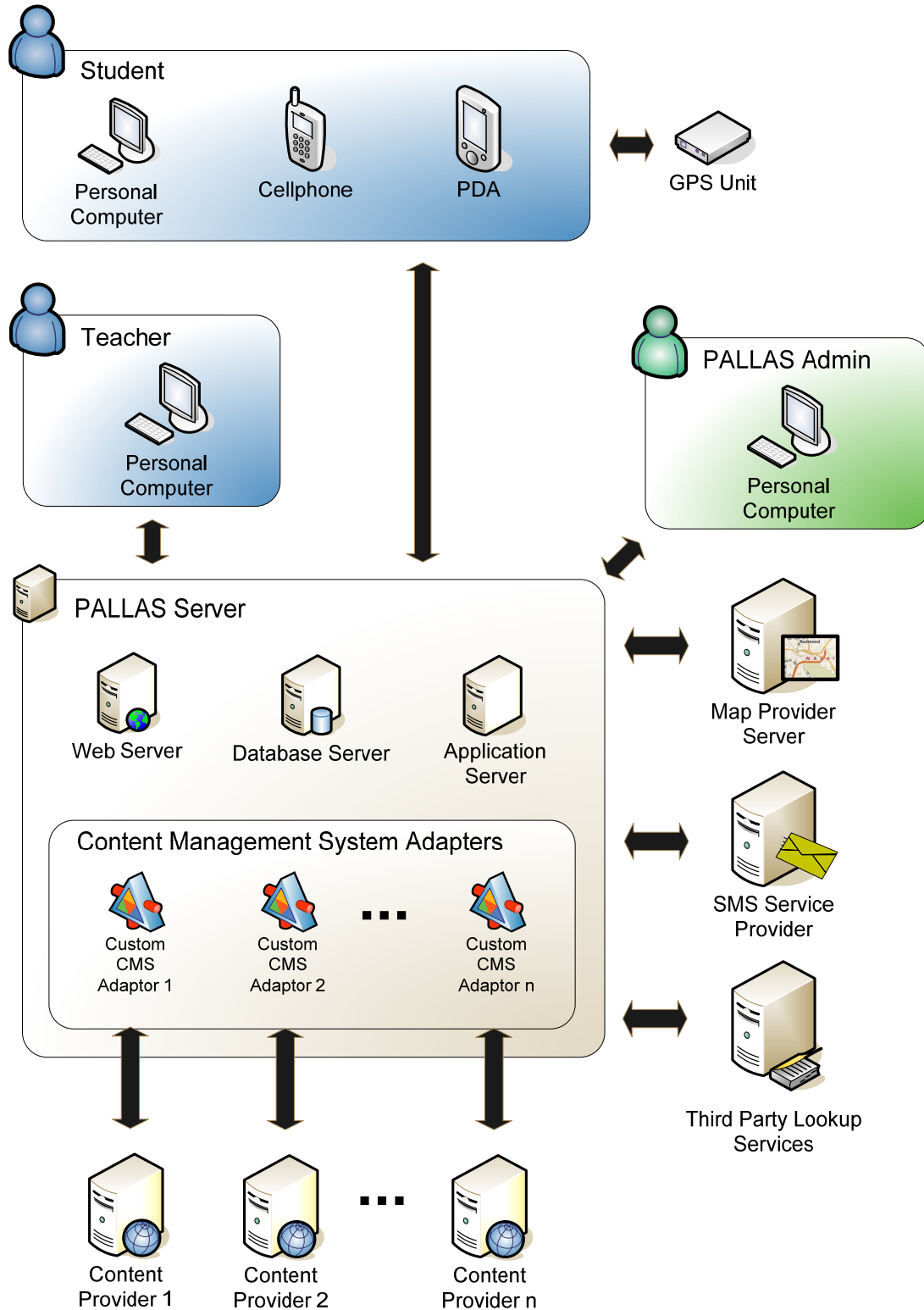
The product of the previous chapter's scenario analysis was generalized system models of how PALLAS should work. With these models ready we now have a solid foundation to use for the development of the PALLAS system architecture. In this chapter I will suggest a platform independent system architecture for PALLAS. At the end of the chapter I will also present some possible development platforms that can be used to implement the PALLAS mobile client.

### **5.2 System Overview**

The PALLAS system is based on a central server architecture. As shown in Figure 9 the PALLAS Server is the main hub of communication, content storage and content distribution. The main server components are the Web Server, which can host web pages and web services, the Database Server, for data storage, and the Application Server which is used to run scheduled tasks such as importing content from the publishing houses' content management systems. The server uses a set of Content Management System Adapters to be able to interface with the different content providers. Each content provider can have their own customized interface adapter. New adapters can be added to allow interaction with new content providers. The PALLAS server also uses a range of different third party service providers. The Map Provider gives access to map data which is useful for PALLAS' context needs. Maps are among other things used to make it easy to add location context to content. The reason why the Map Provider interfaces through the server instead of directly to the different clients is that it makes the use of maps transparent to the client application. This makes it possible to for example change to a different provider without changing any code in the client applications – they will still use the same PALLAS Server interface to get maps. The SMS Service Provider is used to make it possible for the server to send SMS responses to users. Finally, the server uses different Third Party Lookup Services to automatically add context information to content. Some third party services that can be used are for example Google Search [31] (i.e. to find the address of a point of interest) and the MapPoint Web Service [32] (i.e. to get the location of that address).

The PALLAS system administrators are responsible for performing administrative tasks which include giving teachers access to PALLAS and configuring the server's content import module settings. This functionality can both be provided through a web site on the PALLAS server and a client application that uses the PALLAS Web Services. All the system administrators need to perform their tasks is a normal PC with a web browser or the client application. The teacher can also access the PALLAS server in the same way, but with a web site/client application specifically developed to support their tasks. Students have a range of different ways to use PALLAS. They can use a standard PC to log onto a student portal web site on the PALLAS server or they can use mobile devices such cellphones and PDAs. The mobile devices can use PALLAS in a number of ways: They can run a custom client application that provides all functionality (including full context-sensitivity), they can use a mobile web

browser as a thin client or they can do queries by SMS. If the student is using a mobile device with the PALLAS mobile client application installed they can use a GPS unit to automatically provide location information.



**Figure 9:** System overview of the PALLAS architecture

### 5.3 The PALLAS Server

The PALLAS Server is a central hub that offers a range of services needed for PALLAS to work. As Figure 10 shows there are three ways for the PALLAS users to interact with the server: They can use a web browser that accesses PALLAS web pages, they can use a client application that uses the PALLAS Web Service Interface or they can send an SMS from their cellphone.

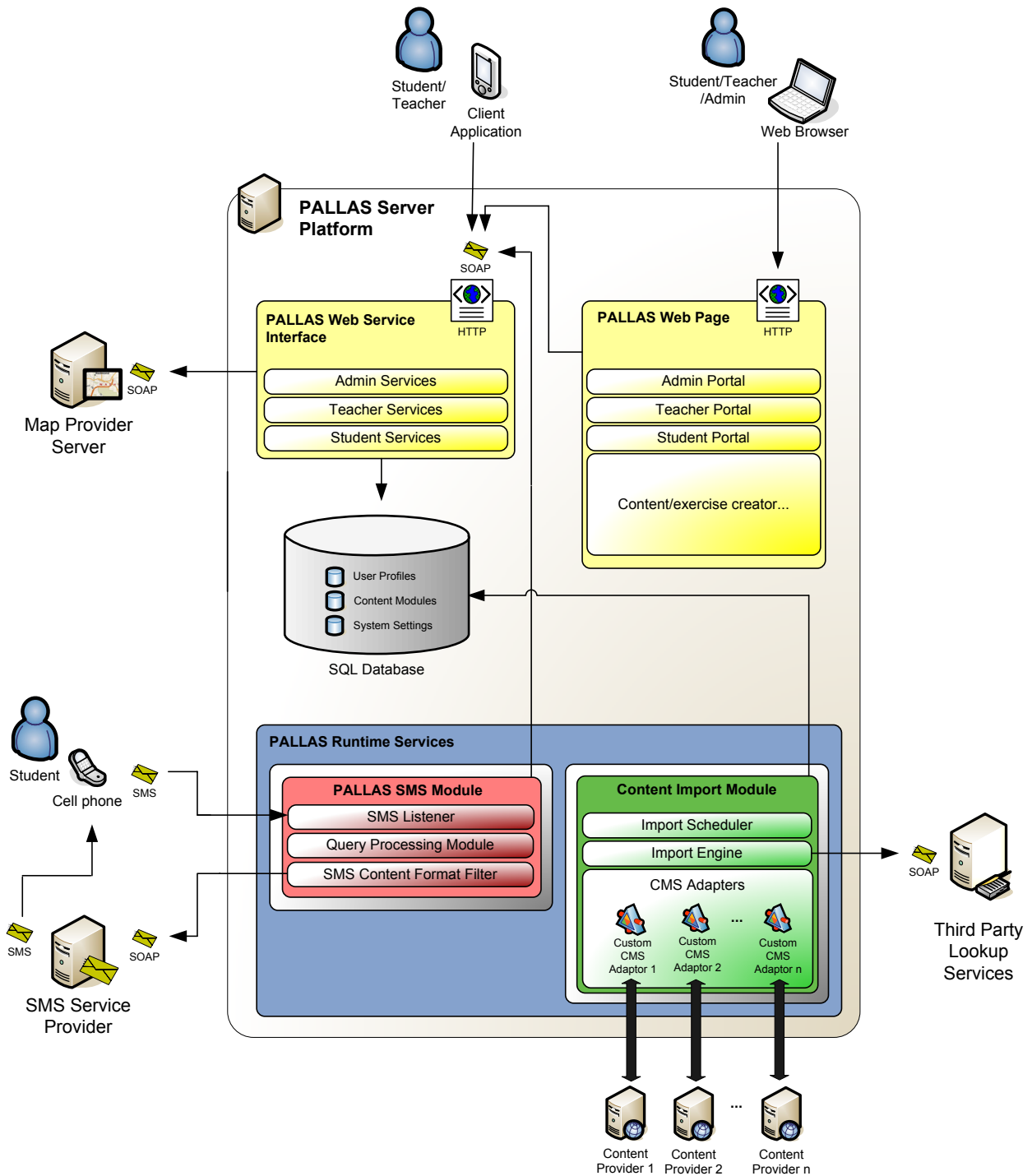


Figure 10: Overview of the PALLAS server architecture



The PALLAS Web Page module offers portals for system administrators, teachers and students. These portals use the PALLAS Web Service interface for to perform core PALLAS tasks. In that sense the portals work pretty much the same way as the client applications, but the advantage of the PALLAS Web Page is that you do not need to install any extra software on your computer to use it. The PALLAS Web Service Interface (PWSI) offers a consistent interface that can be used to perform most of PALLAS' functionality. These services are exposed through the industry standard Simple Object Access Protocol (SOAP) over HTTP, and are hosted on the PALLAS Server's web server. SOAP over HTTP uses port 80, so if the user's computing device can browse web pages the client application should not have any trouble with firewalls when using the PALLAS Web Services. The PALLAS Web Services are divided into three modules based on the target users and their tasks: Admin Services, Teacher Services and Student Services. To be able to provide map data the PWSI uses web services provided by a map provider. There are many map providers available. Because the client applications use the PWSI for all their PALLAS services it is transparent for them which map provider that is used.

The PALLAS Web Services need access that to provide their functionality. This is done by connecting to the SQL Database on the PALLAS Server. The SQL Database is where all the PALLAS data is stored. It contains modules for User Profiles, Content Modules and System Settings. The User Profiles contain all the important information about the users, which includes data that is needed for adaptivity. If a student is progressing and his knowledge level has increased that information will be stored in the User Profiles. The Content Modules are used to store all the content that is either created by the teachers or imported from publishing houses. The specific content modules and their relationship have been covered in chapter 4.5. The System Settings stores the configurations made by the PALLAS administrators. This includes settings for the Content Import Module.

The Content Import Module is part of the PALLAS Runtime Services. It is used to import content from different publishing houses' content management systems and into PALLAS' content storage. The Import Scheduler is controlled by the configurations set by the PALLAS Administrator, and will based on that start importing content by using the Import Engine. Each content provider might have different interfaces for downloading content. This is why the Import Engine uses the CMS Adapters layer when downloading content. In the CMS Adapters layer each content provider can have their own customized CMS Adaptor. The adaptors will convert the content to PALLAS' content module format. The transformation is transparent for the Import Engine. Each time the Import Engine is run it will get the new data from all the content providers through the CMS Adaptors layer. Based on the System Settings it can then try to automatically add context information to context by utilizing Third Party Lookup Services. Finally the new content will be stored in PALLAS SQL Database.

The PALLAS Runtime Services also contain the PALLAS SMS Module. This module is used allow students query PALLAS for information through SMS messages on their cellphones. The SMS Listener checks if a new SMS has been received. If it has, the message will be forwarded to the Query Processing Module which will load the users profile, based on their phone number, and perform their query using the PALLAS Web Service Interface. The profile adapted query reply will then be

forwarded to the SMS Content Format Filter. This filter will convert the reply to a format that is suitable for SMS messages and then use an SMS Service Provider Web Service to send the reply back to the student.

#### 5.4 Teacher and Admin Clients

The PALLAS system administrator and the teachers can access PALLAS through the Admin Portal and Teacher Portal. However, because core functionality of PALLAS is in the PALLAS Web Service Interface it is easy to create custom client applications. The main advantage of creating these applications is that it is possible to create much richer user interfaces than you could on a web page. You could also provide caching features to allow offline use, and then perform synchronization with the PALLAS Server when an internet connection is available again. This is for example useful for teachers with laptops that are not always connected to the internet. Cache and synchronization would allow them to create the content whenever and wherever they want. Besides this feature the client application would act as a thin client and mostly rely on the services of the PWSI. The implementation of such an application could be based on a three-tier application pattern [33], but with an added synchronization-tier between the logic-tier and presentation-tier. As shown in Figure 11 the presentation tier and synchronization tier would be on the client-side application. The logic tier (PWSI) and data (SQL Database) tier reside on the server. When in use the presentation tier will use the synchronization tier for performing tasks. The synchronization tier will take care of the dirty work, like figuring out if an internet connection is available or not, and if it is not it will use its own logic tier and data tier to save content for later synchronization.

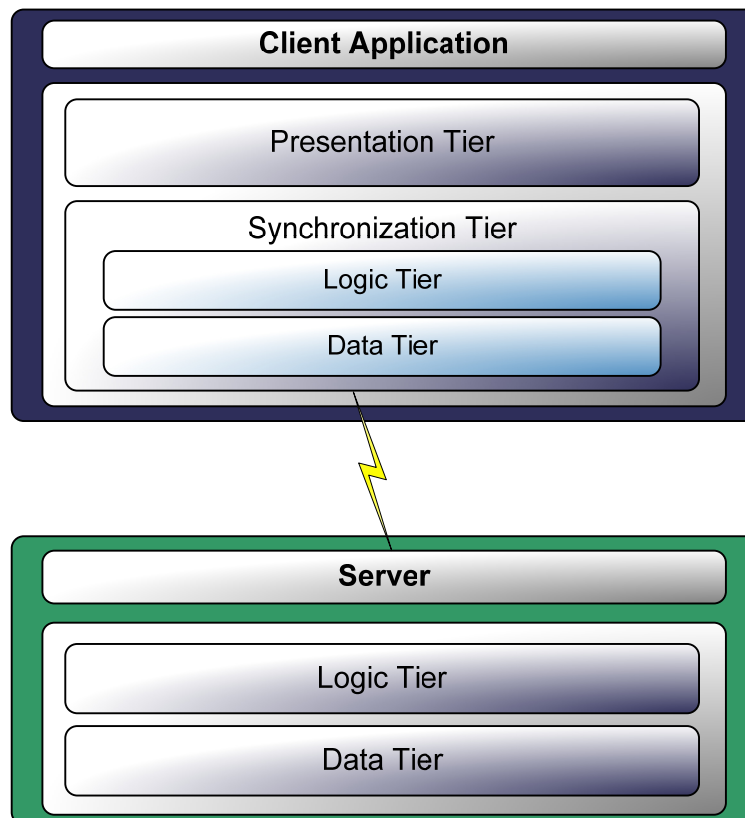


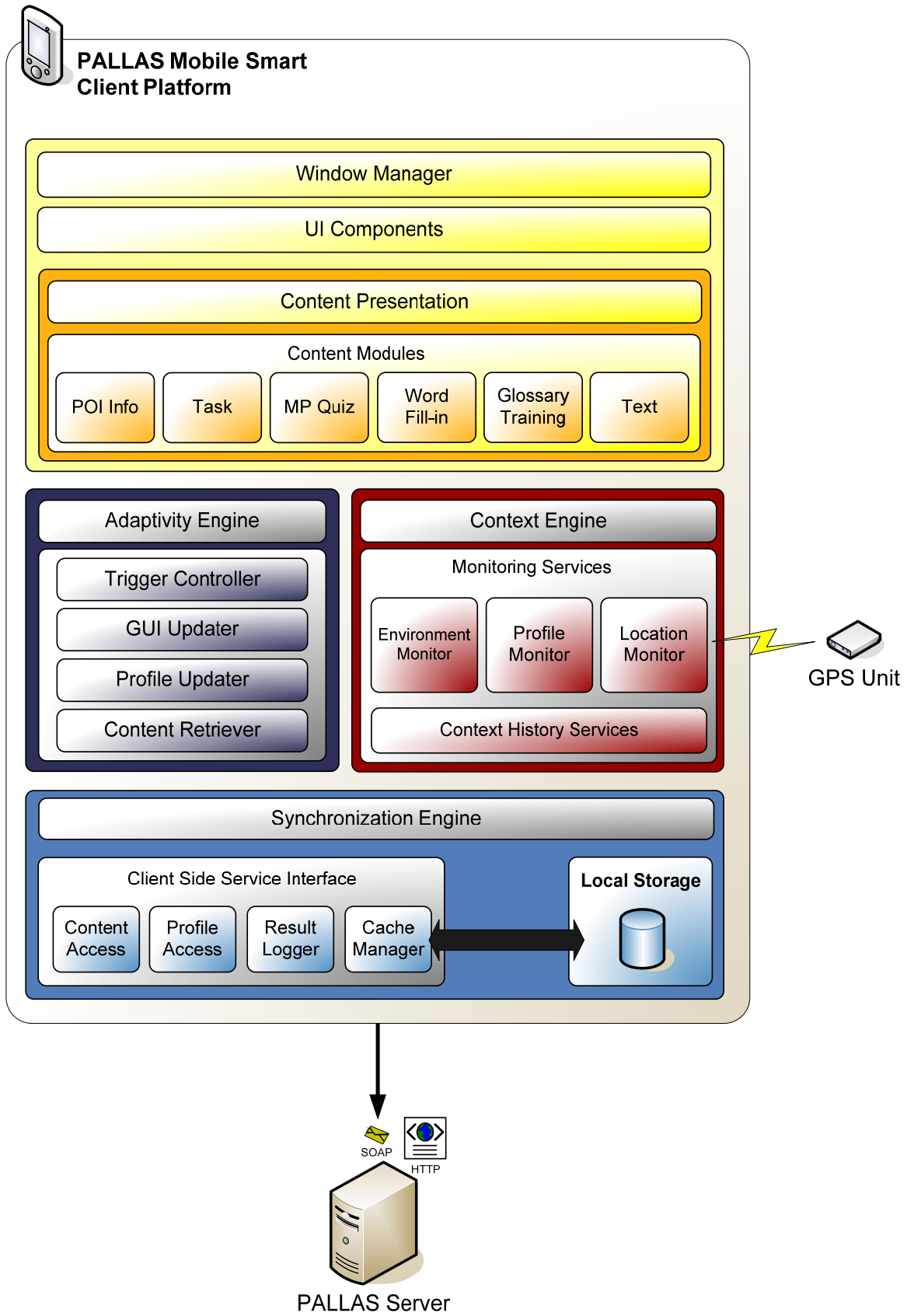
Figure 11: Four-tiered client server application with synchronization

## **5.5 Mobile Smart Client**

The Mobile Smart Client platform needs to support a lot of features. All the functionality from Use Case 1 (chapter 4.4.1) should be available. Quickly summarized this includes lookup and display of content, context-sensitivity, automatic adaptivity, caching and synchronization of data. The architecture shown in Figure 12 is designed to handle all of this. The yellow box represents the presentation tier. Window handling can often be harder on mobile devices than on regular PCs because of the limited screen space and resolution. The Window Manager is used to simplify this by handling all the application windows. A stack can be used to push windows on when a new window is shown. When the new window is closed the Window Manager will pop the previous window from the stack and display it again. The Window manager can additionally be used to store different window layouts that are targeted at different screen sizes and resolutions. The presentation tier also has a Content Presentation block. This used to display and allow interaction with all of the different content PALLAS provides. The Content Presentation block has a content module for each of the different types of content that can be displayed.

The Context Engine and Adaptivity Engine are needed to make the mobile PALLAS client context-sensitive and adaptive to the user and environment. The Context Engine runs monitoring services that keep track of a range of different context data. It also has the Context History Services to allow storage and retrieval of previous and future context data. The Context Engine also provides events that other modules can subscribe to so that they can get notified when a specific context type has changed. The Synchronization Engine uses for example this to get notified if an internet connection is available or not. The Adaptivity Engine also subscribes to a lot of the events of the Context Engine. This is needed so that the Trigger Controller, GUI (Graphical User Interface) Updater and Profile Updater automatically can react to context changes. The presentation tier should always use the Content Retriever module when looking up content. The Adaptivity Engine will then automatically make sure that the context information needed is sent with the content request.

All PALLAS content is initially stored on the PALLAS Server. On PALLAS Mobile Smart Client Platform all communication with the PALLAS Server is done through the Synchronization Engine. This engine works as a transparent layer to the PALLAS content. If an internet connection is available it will get content directly from the server, if not it will use the locally cache data. Data that needs to be sent to the server, such as test results, can also be cached locally and sent to the server upon the next synchronization. The Synchronization Engine uses the PALLAS Web Service Interface when communicating with the server.



**Figure 12:** Mobile Smart Client application architecture

The smart client architecture described here seems to be a good solution in terms of the hardware limitations discussed in chapter 2.4. Network connectivity issues are handled by caching data you download when you have a connection, and use the cached data when you do not. If a query is made and you have poor or expensive connection (i.e. GPRS where you have to pay for the data you transfer) then the client will present you with the locally stored content first. If that content is unsatisfactory new content can be downloaded from the server. This minimizes the use of data transfer. Less data transfer is also good for the mobile device's battery life because wireless communication is very power intensive. The drawback compared to always getting content from the server is that the cached content might not contain the latest changes that are made on centrally stored content. The cached data would however be updated upon the next synchronization. This method will also allow much better updating of content than we would have if we stored all the content locally on the mobile client. Caching of data will also have minimal storage requirements compared to complete local storage. When creating the graphical user interface for the mobile client it is important to take the limitations of mobile devices' screens into consideration. The Window Manager can handle windows that are targeted at different screen sizes and resolutions, and the Adaptivity Engine will make sure that the best match for the current device is used. This does not mean that the design of the GUI needs less attention, but it makes it easier to provide the best possible GUI to as many devices as possible.

## **5.6 Possible Mobile Device Development Platforms**

There are an endless number of different mobile devices available on the market today. To support development for these devices there are also a many different development platforms. I have researched five of the most relevant mobile development platforms to be able to decide on the best platform to use for developing the PALLAS prototype.

Sun Microsystems' *Java Platform, Micro Edition* [34] is a collection of Java Application Programming Interfaces (API) targeted for development on resource constrained devices. The JME architecture consists of three modular and scalable layers: Java Virtual Machine (JVM), Configurations and Profiles [35]. The JVM layer implements a virtual machine that is customized for each device's host operating system and needs to be able to support a specific JME configuration. The configuration defines a minimum set of Java class libraries and JVM features that are available. The configuration tells the developer what minimum set of Java platform features he can assume are available for use on the devices with the selected configuration. The profile layer defines the minimum set of APIs on a particular group of devices. These APIs are developed on the underlying configuration. The profiles are designed to provide device portability. If you have written an application for one profile it should be able to run on any device that supports this profile. Small, mobile, such as cellphones, usually support the Connected, Limited Device Configuration. The only profile currently available for this configuration is the Mobile Information Device Profile (MIDP). Applications written for profile are usually called MIDlets [36]. The applications have to be written in the Java language. There are many different JME development toolkits that can be used to ease the development process. Cellphone manufacturers, like Siemens and Nokia, even provide their own free development toolkits [37].

The *.NET Compact Framework* (CF) [38] is a subset of the full .NET Framework developed by Microsoft. The .NET CF was designed to run on a variety of different operating systems so the architecture was specifically developed to support cross-OS functionality [39]. The Platform Adaptation Layer (PAL) is designed to provide an abstraction layer between the host operating system's specific APIs and the .NET CF's requirements. .NET CF support for new devices could therefore be added by creating a new PAL that matches the Common Language Runtime's (CLR) requirements and the new host operating system's capabilities. The CLR lies on top of the PAL and can run a processor-independent operation language emitted from the compiler that is called the Microsoft Intermediate Language (MSIL). The just-in-time compiler (JIT) converts this MSIL to the specific processor's machine code. Each processor type needs thus a runtime implementation that can JIT compile to its machine code. On top of the CLR is the programming infrastructure which includes base class libraries that supply building block functionality for all applications. .NET CF also includes high-level functionality built on top of that, such as XML Web Service support, a graphics and Windows Forms library and the data-access technology ADO.NET. All these building blocks make the .NET CF very suitable for rapid application development. Microsoft's Visual Studio has full support for development for the .NET CF, including a what-you-see-is-what-you-get (WYSIWYG) window GUI designer. In theory developers can program in any language they want when creating applications for the .NET CF, as long as there is a compiler that can convert the code to MSIL. At the moment only two programming languages are supported: C# and Visual Basic .NET.

Adobe offers a lightweight version of their Macromedia Flash Player that is optimized for small mobile devices. It is called Macromedia Flash Lite [40]. Flash Lite 2.0 is based on Flash Player 7.0 and it supports ActionScript 2.0 for programming. It is designed for use on devices such as cellphones and has API features to make it easy to do just that, i.e. device capabilities lookup and info, dynamic xml support and persistent data storage. To be able run Flash Lite applications you need the Flash Lite player on your device. Some phones have the player preinstalled. If your device does not already have the player installed then you can buy it from Adobe – provided that they support your device.

If you need a complete platform-independent development platform you could develop your application as web pages. Wireless Markup Language (WML) and XHTML are well suited for implementing web pages that target devices with small screens [41]. The web pages can for example be developed by using server-side scripting languages such as JSP, PHP, ASP.NET etc. However, using this web page approach would effectively make the mobile device a thin client. It would also be hard, if not impossible, to implement a lot of the functionality of the PALLAS Mobile Smart Device Platform. Mobile web pages could instead be used as a supplement to the PALLAS Mobile Smart Client. This would allow PALLAS to be accessed from even very primitive cellphones that can not run the smart client application.

The final development platform I have researched is called Crossfire and it is developed by a company called AppForge [42]. Crossfire consists of a plug-in that allows development in the Visual Studio IDE, a Crossfire Client which is a runtime environment that allows Crossfire applications to run on a device and finally the

Crossfire Component SDK that allows third parties extend the functionality of Crossfire so that it can meet their needs. The Crossfire Client is available for a great number of mobile devices, for example the BlackBerry, Palm OS, Windows Mobile OS and several Sony Ericsson and Nokia units. This will in turn mean that a Crossfire application can run on all of these devices, but you will need to by the Crossfire Client first. The capabilities of all these devices are very varying though, so you still might have to perform some customizations. As mentioned, you can use Microsoft's Visual Studio IDE to develop Crossfire applications. The programming languages you can use are C#, Visual Basic .NET and Visual Basic 6.0. The code will be compiled into a Crossfire application that can be run on any Crossfire Client.

<b>Platform</b>	<b>Target devices</b>	<b>Pros</b>	<b>Cons</b>
<i>Java ME</i>	<ul style="list-style-type: none"> <li>Cell phones/PDAs with Java ME installed</li> </ul>	<ul style="list-style-type: none"> <li>Runtime included on many cellphones</li> <li>Familiar for Java developers</li> </ul>	<ul style="list-style-type: none"> <li>Hard to create nice and functional user interfaces</li> </ul>
<i>.NET Compact Framework</i>	<ul style="list-style-type: none"> <li>Cell phones/PDAs with Windows Mobile OS</li> </ul>	<ul style="list-style-type: none"> <li>Visual Studio is great for rapid application development</li> <li>Great support for Web Services</li> <li>Extensive class library with ADO.NET for data access</li> <li>The runtime component is free</li> <li>Familiar for .NET developers</li> </ul>	<ul style="list-style-type: none"> <li>Runtime is only available for Windows Mobile devices</li> </ul>
<i>Adobe Flash Lite</i>	<ul style="list-style-type: none"> <li>Cellphones and PDAs that can run the Flash Lite Player</li> </ul>	<ul style="list-style-type: none"> <li>Good cross-platform support</li> <li>Familiar for Flash developers</li> </ul>	<ul style="list-style-type: none"> <li>Need to buy Flash Lite Player if not already installed</li> </ul>
<i>WML/XHTML/HTML</i>	<ul style="list-style-type: none"> <li>All cellphones and PDAs with a WAP, XHTML or HTML browser</li> </ul>	<ul style="list-style-type: none"> <li>No need to install any extra applications if you have a web browser</li> <li>Platform independent</li> </ul>	<ul style="list-style-type: none"> <li>Limited functionality in terms of user experience – everything has to be done on the server</li> </ul>
<i>Appforge Crossfire</i>	<ul style="list-style-type: none"> <li>Palm OS, Windows Mobile OS, Sony Ericsson P800/900/910, Nokia 60/9300/9500, Motorola A1000</li> </ul>	<ul style="list-style-type: none"> <li>Good cross-platform support</li> </ul>	<ul style="list-style-type: none"> <li>Need to buy Crossfire Client</li> </ul>

**Table 5:** Summary of some of the advantages and disadvantages of the mobile development platforms

There are many advantages and disadvantages for each of the presented development platforms. Table 5 lists some of them. It is hard to choose one platform as the best one. What platform that is best is totally based on what needs you have in terms of application functionality and what skills you already possess. Having knowledge about each of the platforms will at least enable you to make an informed choice of development platform.

In the case of the PALLAS system it is important to observe that the choice of mobile device development platform is independent of the development platform used for the PALLAS Server. The PALLAS Server exposes its functionality through standard Web Service using the SOAP protocol. So as long as the mobile platform supports consumption of Web Services it will be able to communicate with the PALLAS Server.

## **5.7 Summary**

This chapter dealt with step 3 of my four-step scenario-based approach (Figure 2) – the creation of a system architecture for the PALLAS system. The requirement models I created in the previous chapter were used as a starting point for the architecture development, but also to verify that the suggested architecture would satisfy all the needs of PALLAS. Research question 7 was answered through this process by suggesting how a personalized and context sensitive foreign language learning system should be designed. In the end of the chapter I also answered research question 6 when I presented five possible development platforms for creating mobile devices.



## 6 The PALLAS Prototype

### 6.1 Introduction

There are two main benefits of developing a prototype of the PALLAS system. The first one is that it is a great way to find out how well the proposed system architecture is suited for real-life implementation. The second one is that a prototype is the perfect tool to test how well the principles of mobile context-sensitive learning that I have proposed really work. As previously mentioned, I do not have any intentions to perform a thorough test of the prototype with actual students as users. This is mainly due to time restrictions. However, with the prototype well developed these kinds of tests will be easier to perform in future research.

All the source code for the prototype system is included with this report. It does require a lot of setup, with web servers, databases etc. Installation instructions are available in Appendix B. I have also made a short video showing the PALLAS prototype. This makes it possible to see the prototype in action without having to go through the installation process. The video is available online at the following URL: <http://pallas.prismesoft.com>

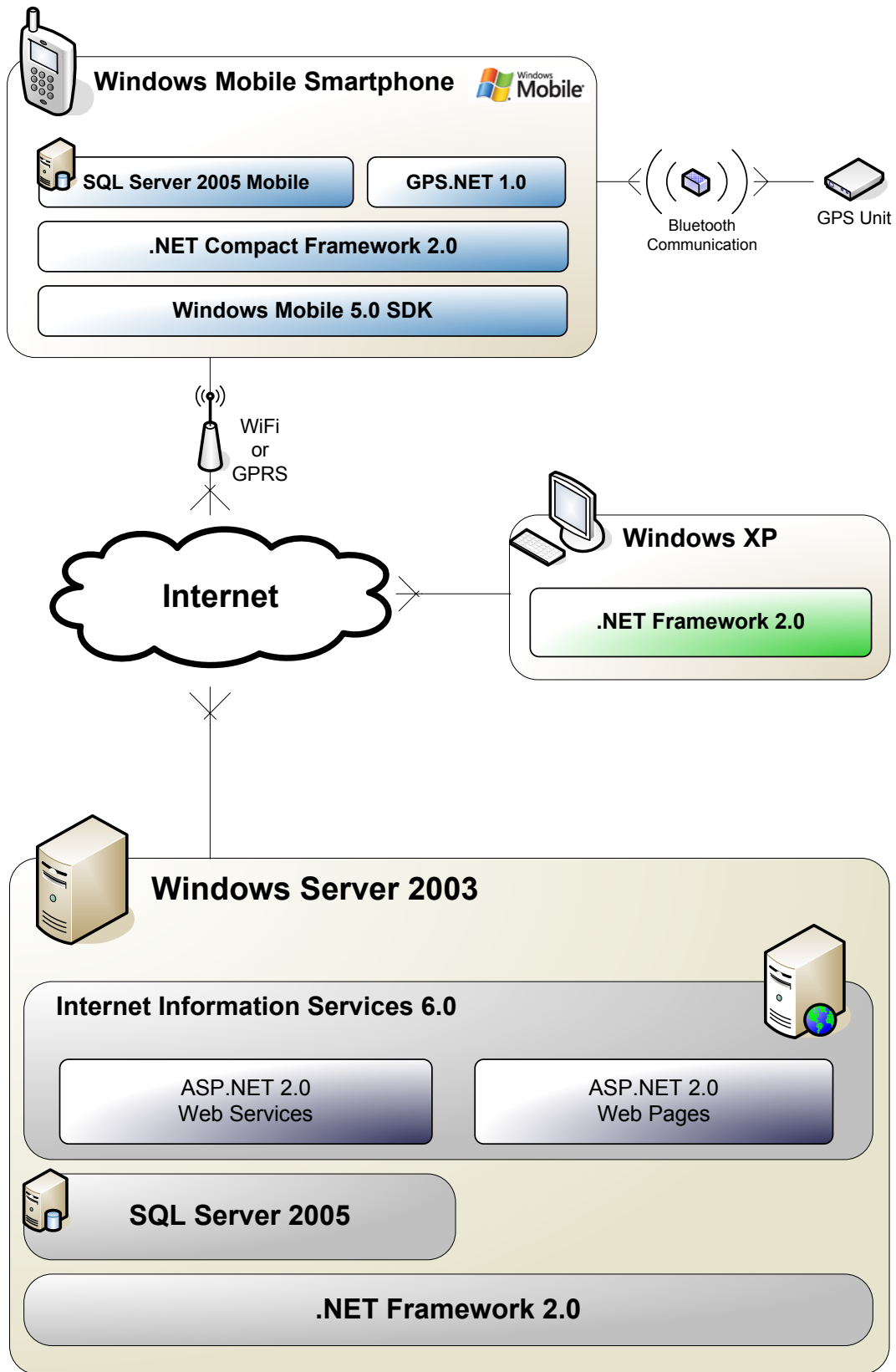
### 6.2 Technology Choices

In chapter 5.6 I covered five mobile device development platforms which can be used to develop the PALLAS Mobile Client. However, when creating a prototype of the complete PALLAS system we also need to develop a lot of extra infrastructure that is not mobile. Choosing development platform for this project was therefore based on an evaluation of the complete system. In the end I chose to use Microsoft Windows technology and the .NET development platform for this project. There are four main reasons why I ended up with this solution:

1. .NET was designed to support Web Services from the very beginning, and is therefore an excellent match for PALLAS' Web Service based architecture. The .NET platform also offers the ADO.NET data access API which makes data handling much more enjoyable.
2. Using .NET with Visual Studio is great for Rapid Application Development. The WYSIWYG window designer and database connection tools allows you to achieve a lot in a relevant modest amount of time.
3. I have previous experience with development on both the .NET and .NET Compact Framework platform.
4. I have Windows Mobile devices that are capable of running .NET CF available for development and testing.

Figure 13 covers the specific technology I used to develop the PALLAS prototype. The mobile device I used was a Qtek 8310 [43] Smartphone running the Windows Mobile 5.0 OS. The Smart Client Application was developed by using the .NET Compact Framework 2.0, with a SQL Server 2005 Mobile available for storage of data. I used a third-party library called GPS.NET 1.0, developed by Geoframeworks [44], to get data from my GPS device. The GPS device I used during development

was a Dell Bluetooth 309 unit. The Qtek 8310 Smartphone has a built in WiFi card which you can use to connect to the internet. You can also use GPRS if you are out of range a WiFi hotspots.



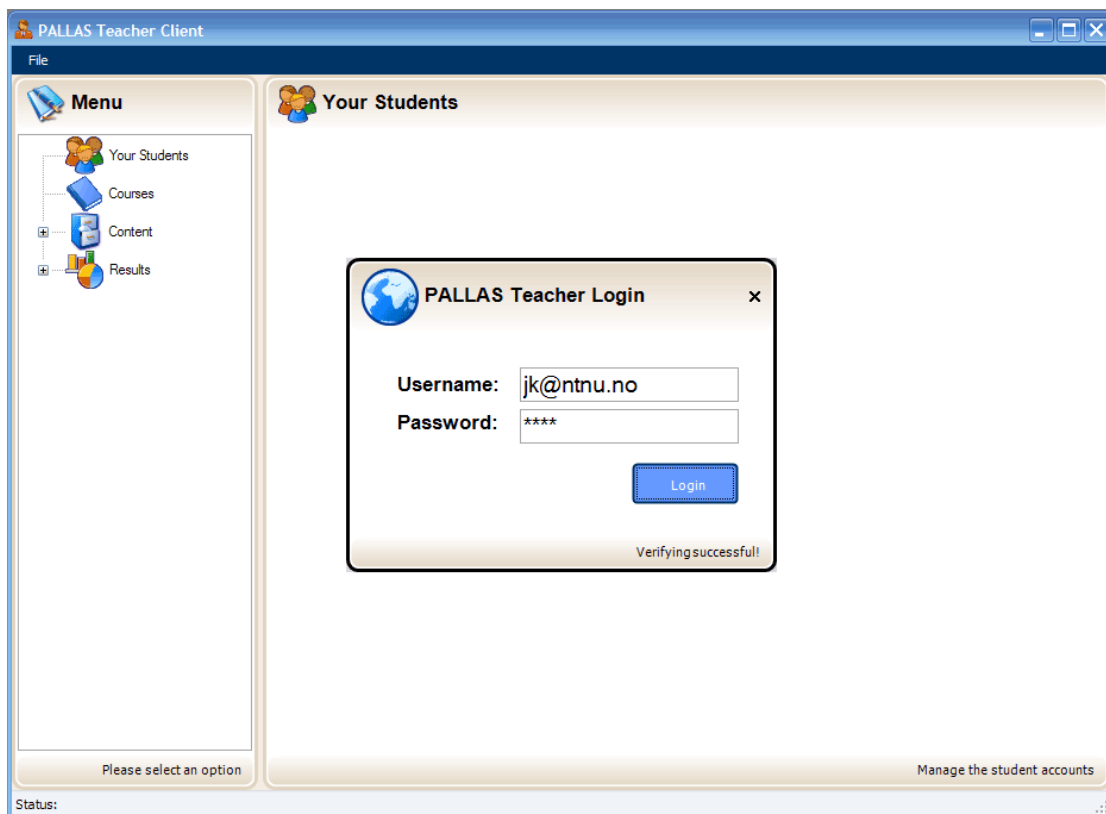
**Figure 13:** Overview of the technology used in the PALLAS prototype

The teachers' client application is a thin client developed on the .NET Framework 2.0. It is designed to run on the Windows XP operating system.

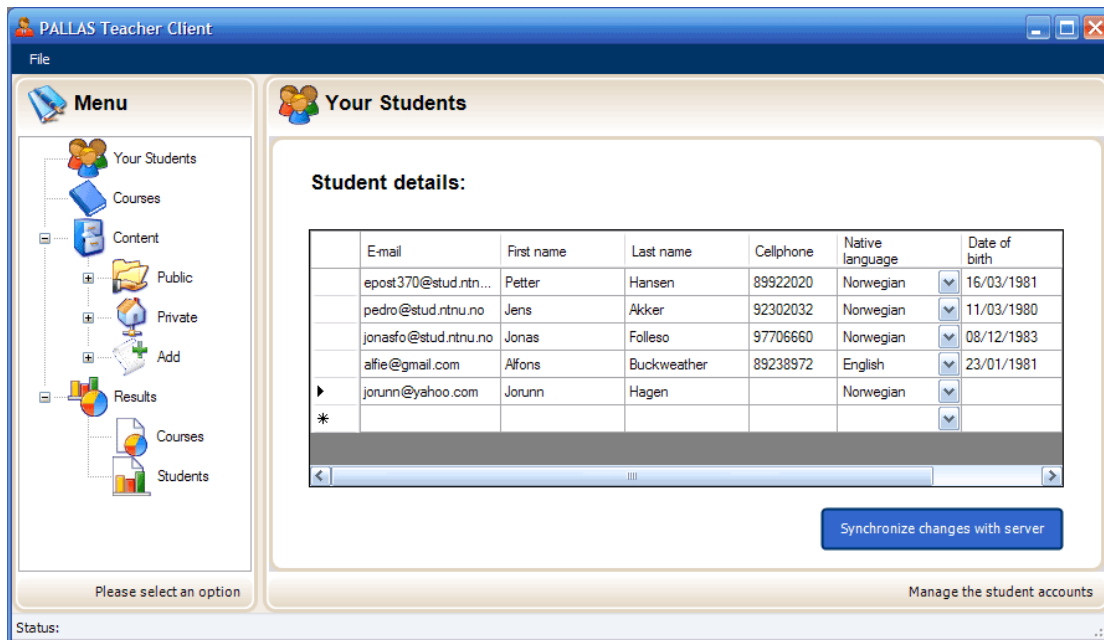
For the PALLAS Server I used a computer with the Windows Server 2003 operating system. I used Microsoft's Internet Information Services 6.0 to host the Web Services and web pages. Both were developed in ASP.NET 2.0, which is part of the .NET Framework 2.0. The server also hosts an SQL Server 2005 database for data storage and retrieval.

### 6.3 Teacher Client

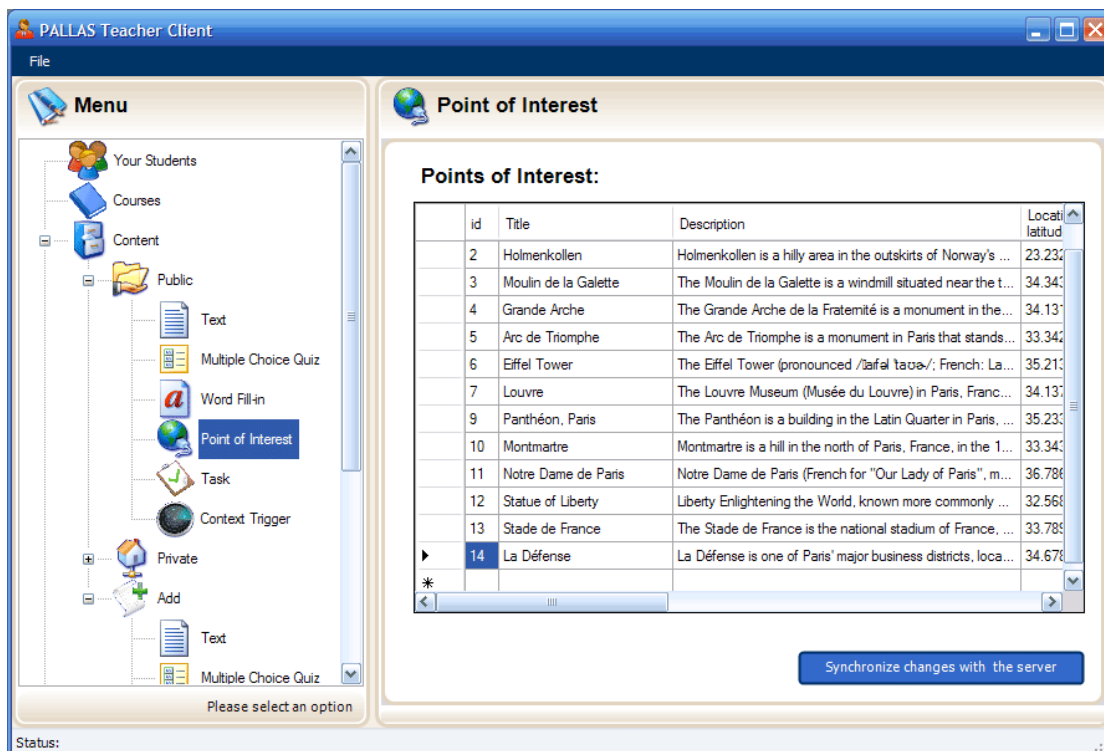
The teachers' client application is designed to allow teachers to manage their students and what content that should be made available to them. All the data they manipulate is stored on the PALLAS Server so that it is available to all the other client applications. The teacher client functionality is based on the use case diagrams and the user scenarios from which they were derived. The following screenshots demonstrate some of this functionality:



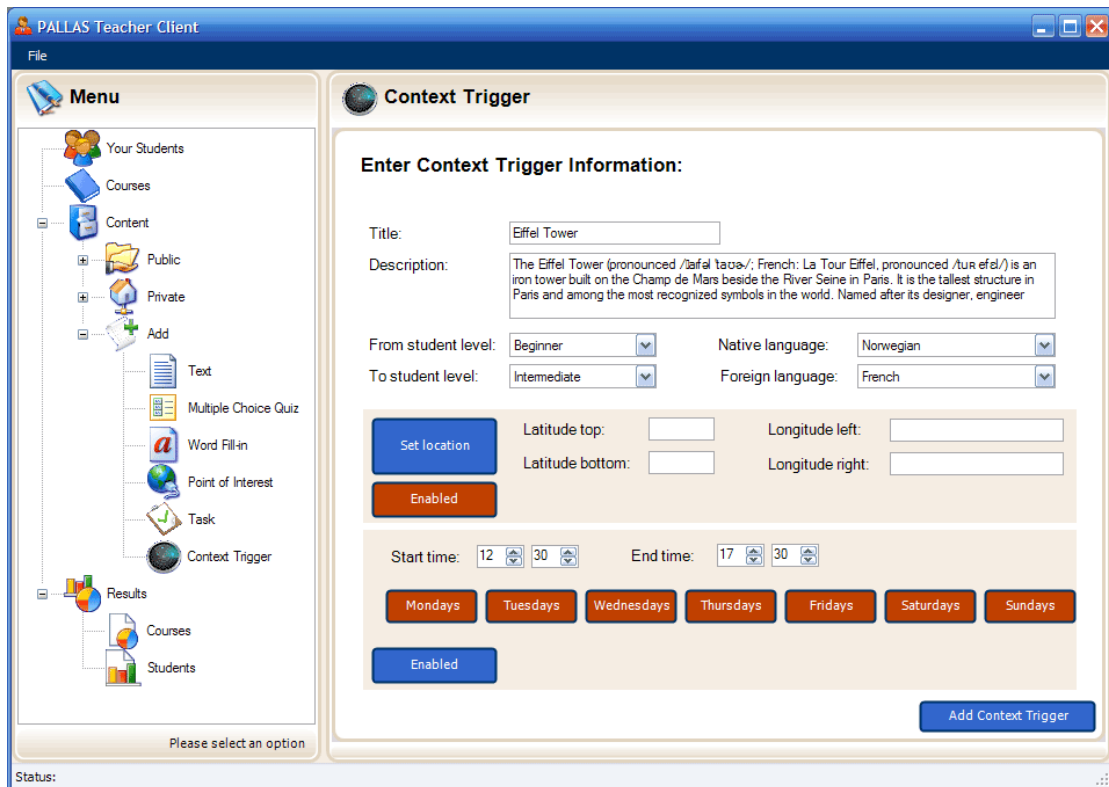
**Screenshot 1:** Each teacher have to login to the system with their personal username and password.



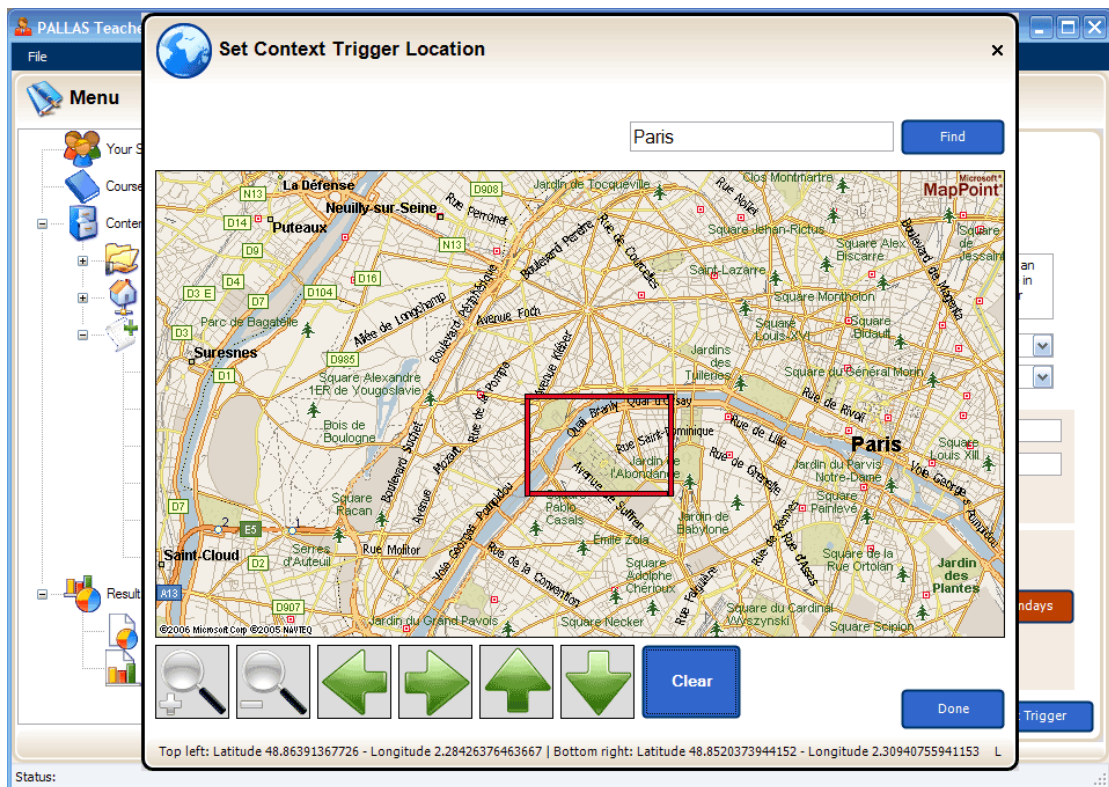
**Screenshot 2:** The teachers can add and delete student accounts. They can also edit the student information. Each teacher will only have access to their own students' profiles.



**Screenshot 3:** The language learning content is divided into several categories. This screen shows different points of interest. There are two sections for language learning content: Public and Private. Teacher can add their own content. The content added to the public section will be made available to other teachers, while the content added to the private section will only be available to the creator.



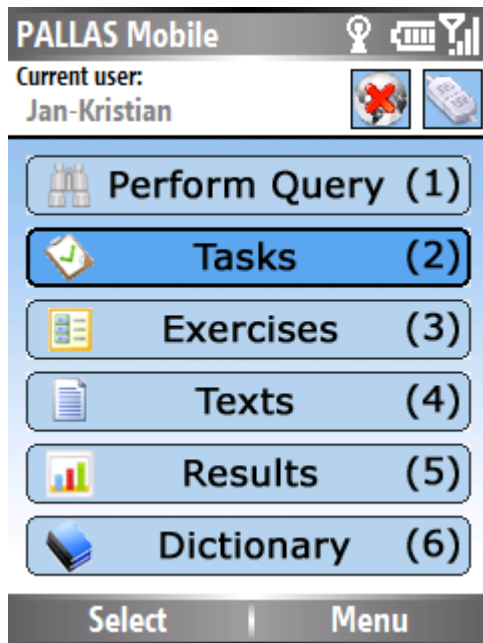
**Screenshot 4:** This screen shows how teachers can add new Context Triggers. They can set parameters like student level, location and time as triggers. Language learning content can be connected to the trigger so that it shows up when the trigger fires.



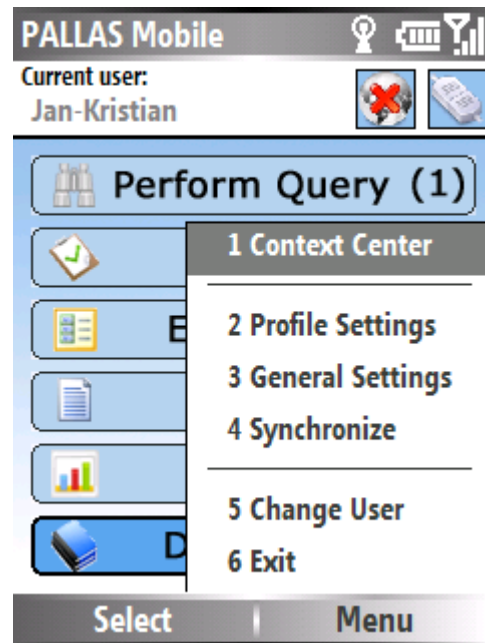
**Screenshot 5:** A map control can be used to help set the location trigger

## 6.4 Mobile Client for Students

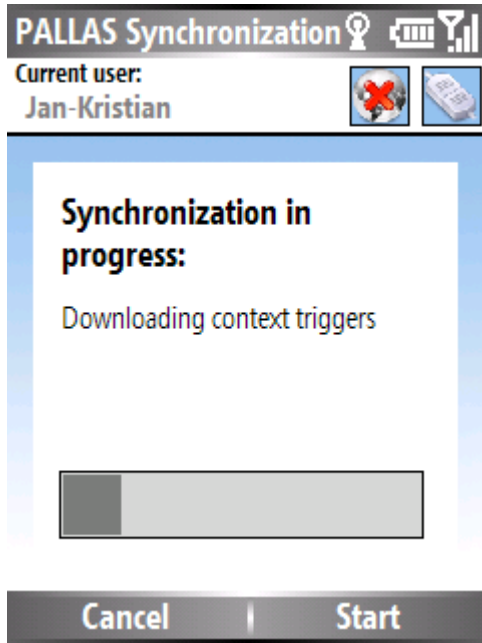
The mobile smart client allows students to access language learning content whenever and wherever. By using context information about the students' current environment and also their personal attributes the application can serve the user content adapted to their need and situation. The following screen shots demonstrate some of the possibilities of the PALLAS mobile client:



**Screenshot 6:** The main window of the PALLAS mobile client. The bar at the top is the context status bar. It shows which user profile that is load, if your location is known (the left box) and if you are connected to the internet (right box). You can chose items from the menu by selecting with the joystick on the cellphone, or by pressing the corresponding number button.



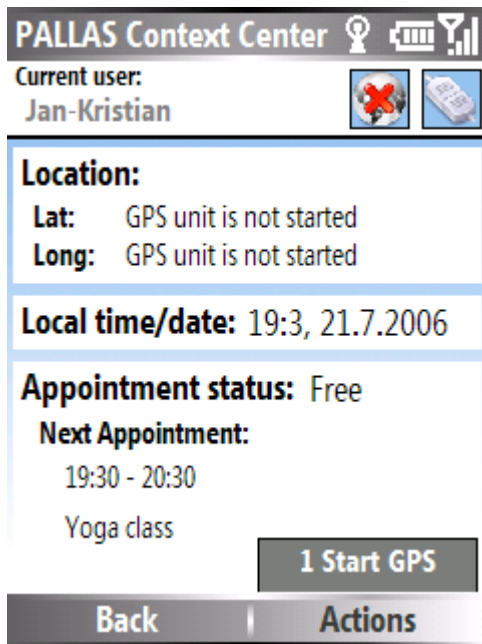
**Screenshot 7:** Under the Menu section you can find different PALLAS settings. Because we have an internet connection in this screenshot, indicated by the connection symbol in the top right corner, we can also synchronize with the PALLAS Server.



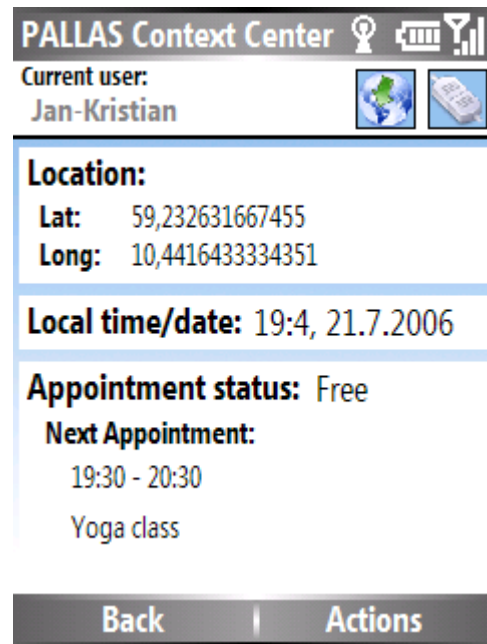
**Screenshot 8:** This screenshot shows the synchronization in progress.



**Screenshot 9:** This is the dictionary lookup, with a French word translated to English.



**Screenshot 10:** The Context Center shows some of the context information that is used to make PALLAS adaptive. We can see the local time, that we currently do not have any appointments in our calendar, when the next appointment is and our current location information status. This is also where setup your GPS unit.



**Screenshot 11:** This screenshot is from after the GPS has been connected. We can see that the status bar shows that we have location information, and our current position is also shown in the Context Center.





**Screenshot 12:** In this screenshot we can see that we do not have an internet connection, but we do have location information. Context Triggers can fire even if we do not have an internet connection because we have cached content stored locally on the device.

## **6.5 Limitations of the Prototype**

The whole point of developing a prototype is to figure out if a concept makes sense without having to implement a full working version. It would be very expensive to fully develop an advanced application like PALLAS. If we cut down on some features, and do not require commercial stability on the application, we can develop a concept prototype relatively fast. This allows us to test new ideas, such as context based language learning content.

The features that I decided not to implement were excluded because they do not add enough value compared to development time, they require special equipment that we do not have available or they simply take too long to implement. Among these features are: The possibility to query for information via SMS, automatic content import from publishing houses (this is not needed for us to be able to test if the concept of mobile language learning works, for this testing we can use manually entered content), the PALLAS administrator client and full support for all exercise types on the mobile client.

It is important to fully understand that this is only a prototype developed for testing purposes. Even the functionality that has been implemented can be guaranteed to work 100%. The limited time that goes into creating such prototypes limits the amount of time that can be spent on software testing. If this prototype were to be used for testing with students it would require further development to ensure good enough quality.



## **6.6 Summary**

This chapter has described the PALLAS prototype I have developed as part of this research project. Developing a prototype was part of my research method because it could be used to see how well suited my proposed system architecture really is. Based on my own personal experience I think the architecture worked very well. It provided coherent interfaces and a well defined layered structure which makes it easier to maintain an orderly code base. Developing a prototype also helped answering research question 6 by providing a real example of a platform that can be used to develop a mobile language learning application. The fact that I could develop a well functioning prototype in such a limited amount of time supports that the proposed architecture is a good starting point for such an application (research question 7).

## 7 Evaluation and conclusions

### 7.1 Evaluation of the PALLAS Prototype

In chapter 4.6 I made a functionality check list based on the functionality I could find described in the user scenarios. The list can be found in Appendix A. The text in *italic* is the evaluation of the PALLAS prototype compared to the original user scenarios. Based on the checklist over half (11/20) of the functionality is available in the prototype. If we deduct the three features that we did not expect to be included in the prototype we have 11/17 (ca. 65%) features correctly implemented according to the check list. Additionally we have one feature that was partially implemented. This result seems partially satisfying, but there is one more detail that makes the overall results of the prototype development process appear even better. 4 of the 5 features that were not implemented were supported in the PALLAS architecture. This means that they can be added without doing any fundamental changes to the system architecture suggested in chapter 5. The reason that they were not implemented at this point was mainly lack of time. Based on this we can confidently say that the prototype and the chosen system architecture seem to be able to cover the requirements of the user scenarios very satisfyingly.

### 7.2 Conclusion

This research project started out with a goal of developing a prototype of a mobile application for foreign language learning. Based on the evaluation of the finished PALLAS prototype, we can quite confidently say that we are on the right track. Because learning is such a subjective experience it is hard to create a learning system that fits everybody's needs. The use of context and adaptivity in PALLAS seem to be one possible solution to this problem.

The main research question of this project was: *How can context sensitive mobile devices be used to complement traditional foreign language learning in a classroom?* I broke this question down into smaller, more manageable research questions. All these research questions have been well covered throughout this report. This has resulted in a system that can be used to complement traditional language learning in a classroom – PALLAS. The prototype of PALLAS was the first step – an important step that showed that we have a working architecture to create such a system. The next step now would be to further develop the PALLAS prototype to a stable release suitable for testing on students. Only after conducting tests on the target users of PALLAS could we find out if the system provides real value to the field of foreign language learning.

## 8 References

---

- [1] *Inspira Web Page*. <http://www.inspera.no/>
- [2] James Milton. *Report 1: Literature Review in Languages, Technology and Learning*. A Report for NESTA Futurelab.
- [3] Wikipedia. *Article about Pallas*. <http://en.wikipedia.org/wiki/Pallas>, accessed April 2006.
- [4] Sue Long, Rob Kooper, Gregory D. Abowd and Christopher G Atkeson. *Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study*. Proceedings of Conference on Human Factors in Computing Systems, 1996.
- [5] V. Kumar, S. Misra and U. Kumar. *Goal-oriented or scenario-based requirements engineering technique - what should a practitioner select?* Canadian Conference on Electrical and Computer Engineering, 2005.
- [6] P. Haumer, M. Jarke, K. Pohl and K. Weidenhaupt. *Scenarios in system development: current practice*. IEEE Software March/April 1998, Vol. 15, 34-45.
- [7] Alistair Sutcliffe. *Scenario-based requirements engineering*. 11th IEEE International Requirements Engineering Conference, 2003.
- [8] Peter Skehan. *Individual differences in second language learning*. Studies in Second Language Acquisition June 1991, Vol. 13, 275-298.
- [9] Howard Gardener. *Frames of Mind: The Theory of Emotional Intelligence*. Basic Books New York, 1990.
- [10] Hubbard, Philip. *A methodological framework for CALL courseware development*. Computers in applied linguistics: An international perspective. Clevedon, England: Multilingual Matters, 39-65, 1992.
- [11] Hubbard, Philip. *Software Evaluation Guide*. <http://www.owl.net.rice.edu/~ling417/guide.html>, accessed May 2006.
- [12] Joy Kreeft Peyton. *Immersed in Writing: Networked Composition and Kendall Demonstration Elementary School*. Technology-enhanced learning environments, Elizabeth Hanson-Smith (Editor), TESOL Publications, 2000.
- [13] Ole Andreas Alsos. *Exploring interface metaphors for using handhelds and PCs together*. Master's thesis at the Norwegian University of Science and Technology, July 2005.

- 
- [14] L. Naismith, P. Lonsdale, G. Vavoula and M. Sharples. *Report 11: Literature Review in Mobile Technologies and Learning*. A Report for NESTA Futurelab.
- [15] George H. Forman and John Zahorjan. *The Challenges of Mobile Computing*. IEEE Computer Magazine April 1994, 38-47.
- [16] Sam Joseph, Kim Binstead and Dan Suthers. *PhotoStudy: Vocabulary Learning and Collaboration on Fixed & Mobile Devices*. Proceedings of the third IEEE International Workshop on Wireless and Mobile Technologies in Education 2005, 206- 210.
- [17] Maria Uther, Iraide Zipitria, James Uther and Pushpendra Singh. *Mobile Adaptive Call (MAC): A case-study in developing a mobile learning application for speech/audio language training*. Proceedings of the third IEEE International Workshop on Wireless and Mobile Technologies in Education 2005, 187- 191.
- [18] Rosa G. Paredes, Hiroaki Ogata, Nobuji A. Saito, Chengjiu Yin and Yoneo Yano. *LOCH: Supporting Informal Language Learning Outside the Classroom with Handhelds*. Proceedings of the third IEEE International Workshop on Wireless and Mobile Technologies in Education 2005, 182- 186.
- [19] S. Modafferi, B. Benatallah, F. Casati and B. Pernici. *A methodology for designing and managing context-aware workflows*. Procedures of Mobis, 2005.
- [20] Odd-Wiking. Rahlff, E. Stav, F. Vraalsen, J. Floch. *Context Model FAMOUS*. SINTEF report, 2004.
- [21] John Krogstie, Kalle Lyytinen, Andreas Lothe Opdahl, Barbara Pernici, Keng Siau , Kari Smolander. *Research areas and challenges for mobile information systems*. International Journal of Mobile Communications 2004, Vol. 2, No.3, 220-234.
- [22] Ayse Göker and Hans Inge Myrhaug. *User context and personalisation*. Workshop proceedings for the 6th European Conference on Case Based Reasoning, 2002.
- [23] William N. Schilit. *System architecture for context-aware mobile computing*. PhD thesis, Columbia University, 1995.
- [24] S. Greenberg. *Context as a Dynamic Construct*. Human-Computer Interaction 2001, Vol. 16, 257–268.
- [25] Tobias Zimmer. *Towards a better understanding of context attributes*. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshop 2004.

- 
- [26] Trude Heift and Devlan Nicholson *Web delivery of adaptive and interactive language tutoring*. International Journal of Artificial Intelligence in Education Vol. 12, 2001.
- [27] Rosa M. Carro, Estrella Pulido and Pilar Rodrigues. *TANGOW: Task-based Adaptive learner Guidance On the WWW*. Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.
- [28] Bonnie Nardi. *The Use of Scenarios in Design*. SIGCHI Bulletin, Vol. 24, 13-14.
- [29] Martin Fowler and Kendall Scott. *UML Distilled Second Edition, a brief guide to the standard object modeling language*. Addison-Wesley, 2000.
- [30] ElKoutbi, Khriiss and Keller. *Generating User Interface Prototypes from Scenarios*. Fourth IEEE International Symposium on Requirements Engineering, 1999.
- [31] Google. *Google SOAP Search API*. <http://www.google.com/apis/>, accessed June 2006.
- [32] Microsoft. *MapPoint Web Service*. <http://www.microsoft.com/mappoint/products/webservice/default.aspx>, accessed June 2006.
- [33] WikiPedia. Article about Three-tier (computing). [http://en.wikipedia.org/wiki/Three-tier\\_%28computing%29](http://en.wikipedia.org/wiki/Three-tier_%28computing%29), accessed June 2006.
- [34] Sun Microsystems. *Java ME Product Page*. <http://java.sun.com/javame/index.jsp>, accessed June 2006.
- [35] Sumi Helal. *Pervasive Java*. IEEE Pervasive Computing Magazine, January-March 2002.
- [36] WikiPedia. *Article about Java Platform, Micro Edition*. [http://en.wikipedia.org/wiki/Java\\_me](http://en.wikipedia.org/wiki/Java_me), accessed June 2006.
- [37] Sumi Helal. *Pervasive Java, Part II*. IEEE Pervasive Computing Magazine, April-June 2002.
- [38] Microsoft. *.NET Compact Framework Web Site*. <http://msdn.microsoft.com/netframework/programming/netcf/default.aspx>, accessed June 2006.
- [39] Craig Neable. *The .NET Compact Framework*. IEEE Pervasive Computing Magazine October-December 2002.

- 
- [40] Adobe. *Adobe's Web Site for Mobile & Devices*.  
<http://www.adobe.com/mobile/>, accessed June 2006.
- [41] Kris Read and Frank Maurer. *Developing Mobile Wireless Applications*. IEEE Internet Computing, January/February 2003.
- [42] AppForge. *Crossfire Product Web Site*.  
<http://www.appforge.com/products/crossfire/index.html>, accessed June 2006.
- [43] Qtek. *Qtek Norway's 8310 Web Site*.  
<http://www.qtek.no/norway/produkter/8310.aspx>, accessed July 2006.
- [44] GeoFrameworks. *GeoFrameworks Web Site*.  
<http://www.geoframeworks.com/>, accessed July 2006.

## Appendix A – Functionality Check List

### Features for Students

Functionality from scenario	Availability in the PALLAS prototype	Passed?
Lookup words in your native language and get a text explaining the subject	<i>You can perform queries which can return many types of content, including links to texts</i>	Yes
Difficulty of content is based on learners knowledge level	<i>Content is flagged with difficulty level which can be compared to what is stored in the students' user profile</i>	<i>The prototype's data structures and architecture support it, but it is not fully implemented yet</i>
Automatic notification for exercises you should practice more on	<i>Not currently available, but results are stored, so this data can be searched to provide this functionality</i>	No
Transparently use different computing devices to access the system	<i>PALLAS' architecture with a centralized storage and service interface is designed to support this, but due to limited time only one student client was implemented</i>	<i>No, but not expected in a prototype</i>
Automatic notifications if you are located close to points of interest (language learning vise)	<i>Contextualized Triggers do just that, they can contextualize any of PALLAS' content</i>	Yes
Dictionary	<i>The prototype has a dictionary (use the internet)</i>	Yes
Personalized profile	<i>Each student has their own profile to track their progress, interests and other attributes</i>	Yes
Shown a list of tasks to perform within a given time period	<i>Contextualized triggers based on time</i>	Yes
Tasks can be real-life tasks or quizzes on the mobile device	<i>Tasks can link to all of PALLAS' content types, or just be a description of real-life task to perform</i>	Yes
Connect mobile device to GPS so that you can see your current position on a map	<i>Can connect to any GPS device that communicates over the COM port and providers you with your current position, but does not show maps yet</i>	Partially
Take pictures to prove that tasks were performed	<i>Not a priority on the prototype</i>	No
Query a word and textbook content, dictionary lookups and quizzes	<i>Queries can lookup any of PALLAS' content</i>	Yes

Query over SMS, reply is based on sender's profile	<i>Not expected in prototype, but the system architecture has an SMS module designed to handle this</i>	<i>Not expected in prototype</i>
Access results through a web portal	<i>Centralized storage and service interface is designed to support this, but due to limited time only one student client was implemented</i>	<i>Not expected in prototype</i>
Automatic suggestions on areas of improvement	<i>Not currently available, result data is stored, so this could be added later</i>	<i>No</i>

### **Features for Teachers**

<b>Functionality from scenario</b>	<b>Availability in the PALLAS prototype</b>	<b>Passed?</b>
Create customized content, i.e. historical information, explanations of special words, quizzes and tasks	<i>This can be done through the Teacher Client</i>	<i>Yes</i>
Make ready made content from text book company's content management systems available to students	<i>The prototype does not support importing the content, but it does support distribution of it to students</i>	<i>Yes, importing of content was purposely not implement due to time restrictions</i>
Create geographic and time-based context linking to content	<i>This can be done through the Teacher Client</i>	<i>Yes</i>
Synchronize mobile devices so that they get the new content	<i>The mobile client prototype can synchronize and cache content locally on the device</i>	<i>Yes</i>
Get overview of how students performed when solving tasks	<i>The results are stored, but the result tracking is not fully implemented yet</i>	<i>No</i>



---

## Appendix B – Prototype Installation Guide

To install the prototype development kit you need the following software:

- Windows XP or Windows Server 2003
- SQL Server Express / SQL Server 2005 / SQL Server 2000
- Internet Information Services 6.0
- .NET Framework 2.0
- Visual Studio 2005
- Windows Mobile 5.0 SDK

You can run the mobile client on the device emulators that come with the Windows Mobile 5.0 SDK. If you want to run it on a cellphone, then you need a cellphone with the Windows Mobile 5 operating system and preferably with a QVGA resolution screen. Any GPS device can be used with the mobile client, as long as it communicates over a COM port.

In the accompanied zip-file there are two directories. *PALLAS SQL 2005 Database* contains a full backup of the PALLAS database used for development, *PALLAS Visual Studio Solution* contains the complete Visual Studio 2005 solution and *PALLASWebService* contains all the PALLAS Web Service files. Follow these steps to setup the development environment:

1. Import the database backup into your copy of SQL Server
2. Copy the *PALLASWebService* folder to your IIS 6.0 path
3. Open the Visual Studio solution
  - a. Change all connection strings in app.config and web.config files to point to your SQL Server
  - b. Update all Web Service references to point to the location where you stored the *PALLASWebService* folder

Quick overview of the important projects in the PALLAS solution:

- **DataAccess:** Data access layer that uses ADO.NET to get data from the PALLAS database. Uses strongly typed data tables.
- **Mobile Smart Client:** (Not finished!) The beginning of a PocketPC PALLAS client, was discarded when I got a hold of Bluetooth GPS and ported this solution to the SP Mobile Smart Client.
- **SP Mobile Smart Client:** The PALLAS smart client for Windows Mobile 5.0 Smartphones.
- **Teacher Client:** The teacher client for Windows PCs.