Abstract
This chapter presents research around pervasive and ubiquitous computing, particularly oriented in the field of human learning. We are studying several solutions to deliver content over a heterogeneous networks and devices. Converting and transmitting documents across electronic networks is not sufficient. We have to deal with contents and containers simultaneously. Related work in interface adaptation and plasticity (the capacity of a user interface to withstand variations of both the system physical characteristics and the environment while preserving usability) is presented and some examples of context-aware adaptation are exposed. We present an adaptive pervasive learning environment, based on contextual QR Codes, where information is presented to learner at the appropriate time and place, and according to a particular task. This learning environment is called PerZoovasive, where learning activities take place in a zoo and are meant to enhance classroom activities.

Keywords
Human-Computer interaction, ubiquitous computing, pervasive computing, m-Learning, e-Learning, mobility, adaptation, adaptive learning issues, context-awareness, plasticity, multimodality, multi-channel interaction, multiplatform e-Learning.

1. Introduction
In the coming years, learning through heterogeneous telecommunication networks will probably become the rule and not the exception. Studies show that information which circulated in the world is progressively being stored in a numerical form. In theory, we should take advantage of improving access to this information, since it is immediately available and consumable. But in fact, with the multiplicity of possible connections to Internet and over heterogeneous networks, is it not always the case.

The new kinds of networks such as WIFI or Bluetooth offer new perspectives of research for the ICT-based education and hypermedia community. The goal is to satisfy all types of users by proposing data-processing solutions available on almost all the products or peripherals available on the market. The rise of a great number of hardware such as mobile phones, PDA or WebTV, having variable capabilities, leads to a reflection on independent interfaces specification, in order to avoid specific developments. The ubiquitous role of the computer makes each day more unsuitable the screen-keyboard-mouse model posed on a corner of a desk (Beaudouin-Lafon, 2000).

The large success and rise of the Internet network is mainly due to the technical standards used and the adoption of languages such as HTML, WML (WML), or VoiceXML (VoiceXML). But, we observe some incompatibilities in spite of the standards promulgated. Indeed, on the one hand, various types of media such as texts, graphics, sounds or video can easily be used and transmitted through networks, but on the other hand, the fact that machines are not necessary from the same vendors, or do not support the same operating system, leads to situations where information processing systems and/or databases are incompatible with...
other data coded in particular formats. Therefore this consumes additional costs and time for each end-user, like students and teachers, whatever his platform, can obtain a product or a satisfying service. The need for easier access to information; whether at the office, home, in the train, etc., is felt all the more with the arrival of new materials, and the success of the pocket computers as well as mobile telephones. One wishes to lean towards the transformation of end-user’s interface for “anyone and anywhere” (Lopez & Szekely, 2001). With the multiplicity of the means of connecting to Internet, it is necessary to conceive generic interfaces and mechanisms of transformation to obtain concrete interfaces for each platform. That’s why the W3C launched an activity in the Device Independence field.

Adaptable interfaces in multi-device e-learning environment or m-learning are playing a very important role in improving the accessibility of these applications, and are leading to their increased acceptance by the users. After the switch from Learning to E-Learning, we are now facing another switch towards M-Learning. Thus, we are entering an era of pervasive computing with the challenge of providing services available anytime and anywhere. In this context, data management is obviously the heart of concerns in what is now called pervasive or ubiquitous computing.

Consequently, recognizing that mobile computing is one of the most rapidly growing areas in the software market, some researches explore the role of adaptation in ubiquitous learning and particularly in the area of mobile computing. Mobile computing has a very strong potential due to the extremely high market penetration of mobile and Smartphone. The significant development of wireless networks and mobile devices, such as phones and laptops, PDA, sensors, or Smartphone, that we know since the past fifteen years leads to profound changes in applications and services offered to users. The terminals available on the market today are more and more powerful. Their autonomy is sometimes limited, but they provide equipment increasingly rich, with multiple connections, GPS receivers, etc.

These systems operate in a dynamic environment particularly because of frequent disconnections or user mobility. They must be able to respond dynamically to changes in these different settings. They therefore must be sensitive to the context in order to be able to adapt dynamically and so provide an important quality of service to users. If only a few applications accessible to the general public have now emerged, some should be available soon in areas such as transport, health, commerce or education.

This document presents some aspects of this scientific problem and is structured as follows: Section two explains the background and motivation of our work. Section three gives an overview about interface adaptation. Section four addresses the plasticity of the user interface, context-awareness and adaptations. Section five exposes in details the problem of adapting to the platform, the user, the task and the environment. Future trends and ideas for further work are presented in section six before the conclusion of the chapter.

2. Background and motivations

Since several years, we are seeing the miniaturization of electronic devices and their integration into everyday life objects. For example, mobile phones are almost all equipped with a good quality camera, diverse connections to networks such as WiFi or GPRS, “free hand” feature, etc. With some kinds of personal digital assistants (PDA) that use GPS, users can be helped and vocally guided to follow a specific route. This trend that consists on systematically digitalizing resources enabling access to data needed anywhere, anytime is sometimes called, in the literature, ubiquitous. However, there is a wide variety of terms used
to describe this paradigm that is opposed to the more conventional desktop metaphor, with one computer per person. This is known as ambient intelligence, ubiquitous and pervasive computing. This refers to the increasing use of widespread processors that exchange small and spontaneous communications with each other and with sensors. Thanks to their much smaller size, these sensors will be integrated into everyday objects, until it become almost invisible to users.

Indeed, as Mark Weiser explains: “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” (Weiser, 1991). He gives here a first definition of pervasive computing. The pervasive computing refers to access the same service through various channels of communication, such as a desktop computer, a PDA or even a phone to use voice, phone keypad (DTMF) or SMS, depending on the needs and constraints of the user. Adam Greenfield, meanwhile, uses the word “Everyware.” This word, formed from “everywhere” and “hard/software” is a neologism to encompass the terms of ubiquitous computing, pervasive computing, ambient computing and tangible media. He explains in more details his thought: “When I talk about surfacing information that has always been latent in our lives, I mean putting precise numerical values on one present location, on what task we might happen to be currently engaged in, and in whose company.” (Greenfield, 2006). He therefore suggested that information as diverse as the tone of our voices, our caloric intake, or the composition of our urine can be used, sometimes even without our knowledge.

The question of adaptation in computer sciences is often studied from three different points of view. The first one, coming from the Artificial Intelligence domain, tries to increase the performances by providing an adapted system to the user. Thus, some help messages could be adapted (Browne & al., 1990); some adaptive hypermedia documents could be generated (Brusilovsky, 2001) etc.

With the second point of view, close to the software engineering domain, the focus is put on the adaptation according to available resources changing. The system is supposed to be adapted in an autonomously and optimally way. The principle of aspect-oriented programming (AOP) (Kiczales & al., 1997), Model Driven Architecture (MDA) and Model Driven Engineering (MDE) are often used. The basic principle of MDA (Gokhale & al., 2002) is developing models of “Platform Independent Model” (PIM) and transforming them into “Platform Specific Model” (PSM) for the concrete implementation of the system. It is not required that all the code is generated automatically, but the overall architecture of the system at least must be obtained as well.

The third point of view, referring to Human-Computer Interaction (HCI) domain, is mainly dedicated to the user interface. The idea is to strictly split the design and the execution levels. The interface is specified only once, at a high level, and transformed according to different interaction contexts (Thevenin & Coutaz, 1999). Unfortunately, none of these approaches allow adapting to an unknown context. This way, we agree with the criticism of (Balme 2008): “The contexts of interaction targets should be known by beforehand, which is contrary to the principles of ubiquitous computing where unexpected and opportunism prevail.”

Another interesting and complex aspect of the adaptation is the semantic facet. Actually, the intention of the author must be respected during the different levels of adaptation. For example, a teacher proposes an interactive e-service composed by a video with some explanations: if a learner accesses to this document through a small device, which is not able to play the video, the intention of the author is not respected, because for him, the video is not
separable from the rest of the documents, and a simple picture, for example is not sufficient to replace this video.

We will present in the rest of this chapter some researches around the notion of adaptation, context awareness, and plastic interfaces for ubiquitous learning. We will see how to provide support for the learner, particularly in mobile pervasive situations, where information is presented to learner in appropriate time and place, and according to a specific task.

3. Interface adaptation and related work

The notion of adaptive interfaces has many facets. Some general works and taxonomies are available (Malinowski & al., 1992), (Brusilovsky, 1996), but concepts change from time to time according to authors, study domains and points of view. As (Paramythis 2004) said: *The field of adaptive systems is infamous for its lack of standards, or even commonly accepted approaches in this respect.*

A first perspective on adaptation is provided by looking at the temporal sequences involved in adaptation processes. Different tasks that occur in adaptation processes can be grouped in stages. From a system-centered point of view, the main stages cited by (Totterdell & Rautenbach, 1990) are the following: variation, selection, and testing. From a user-centered point of view, adaptation stages are known as initiative (it is the decision of one of the agents to suggest an adaptation), proposal (alternatives proposed for adaptation), decision (one of the alternatives is chosen) and execution (Dietrich & al., 1993).

In order to achieve a specific goal, a designer creates a system according to a personal point of view in a context C. A user may have a different point of view from the designer’s, if s/he is in a context C’. Hence, an adaptation of the system is necessary to switch from C to C’.

According to (Trigg & al., 1987) there are four ways in which a technical system can exhibit adaptability:
- A system is **flexible** if it provides generic objects and behaviours that can be interpreted and used differently by different users for different tasks.
- A system is **parameterised** if it offers a range of alternative behaviours for users to choose among.
- A system is **integrable** if it can be interfaced to and integrated with other facilities within its environment as well as connected to remote facilities.
- A system is capable of being **tailored** if it allows users to change the system itself, by building accelerators, specializing behaviour, or adding functionality.

For (Oppermann, 1994), *application systems are not designed for a particular user and a particular task. They are designed and distributed for a class of users and a set of tasks.* According to this author, a way to increase user freedom without increasing the complexity of the system is to provide systems that can the tailored. The two terms commonly found in the literature are adaptable and adaptive systems. A system is called **adaptable** if it provides the user with tools that make it possible to change the system characteristics. A system is called **adaptive** if it is able to change its own characteristics automatically according to the user’s needs.

For other researchers, **adaptability** can be considered as a prerequisite for achieving **adaptivity** and vice versa. However, in the literature, adaptation and adaptivity are often used synonymously (Khalil 2001). Sometimes, the terms “configurable” or “customisable” are used to refer to this kind of adaptation (Weibelzahl 2002).
We have represented some of the main dimensions of adaptation on a map. Each branch presented in Figure 1 is detailed in the following paragraphs. The map is not exhaustive but it helps clarify the main dimensions of adaptation.

Figure 1: Adaptation concepts (limited map)

**Why**
The first question about adaptation is “why adapt?” Basically, the main goal of adaptation is to speed up and simplify usage by presenting to the user what s/he wants to see and thus make complex systems more usable. Presenting easy, efficient, and effective interfaces is the main goal of adaptation (Malinowski & al., 1992). To reach these goals it is necessary to have a user interface that is suitable for heterogeneous user groups and considers increasing experience of a user (Dietrich & al., 1993). The role of adaptivity in the digital document field is to minimize the user’s effort devoted to the exploration of the capacities of the system in order to optimize the effort necessary for the resolution of a problem. As we know, it is difficult to write software that will fit millions of users perfectly, but it is possible to develop systems able to adjust the interface according to the user’s skills, knowledge and preferences, for instance. In order to achieve this adaptation, we will see that underlying models of at least user and task are essential, as well as separating the user interface from the application (Fischer 2000). Adaptive Systems are used in many domains to solve different tasks. The following list of functions, given by (Weibelzahl 2002) and adapted from (Jameson 2001) gives some example applications: help the user find information, tailor information to the user, recommend products, for example for e-commerce, help with routine and repetitive tasks, give appropriate help, support learning, conduct a dialog, and support collaboration….

**What**
Another important interrogation is to consider what part of the application is adapted. Sometimes just the help or tutorial is adapted. In other examples, the dialogue between the system and the user is modified by the adaptation. The different dimensions of adaptation can be illustrated with some examples of adaptation to the delivered content, the navigation, the presentation, the functionalities, etc. Generally, only the presentation interface is adapted. In other cases, both the presentation and some functions of the system are adapted. This last point is called malleability by some authors (Morch, 1995), (Morch & Mehandjie 2000) and is close to the notion of flexibility proposed by (Scapin & Bastien, 1997).
Who
In adaptive interfaces, both user and system can be in charge. Who controls the adaptation and who is really doing adaptation? According to (Browne & al., 1990), when the system controls adaptation, we talk about **personalization** and more generally about **adaptivity**. When the user controls adaptation, we talk about **customization** (choosing among several options) and more generally about **adaptability**. Naturally, it could be a mix of the two, when adaptation is controlled by both the system and the user. Personalization and customization aim at answering the needs and particular characteristics of each user (Cingil 2000). Indeed, these words come from the e-commerce field and insist on an individual dimension of the adaptation. According to (Rosenberg 2001), personalization is specific to the end user and based on implied interest during the current and previous sessions. Some authors point out **personalization**, **customization** and **adaptation** are synonymous in the works of (Mobasher & al., 2000), (Kappel & al., 2000) or (Rossi & al., 2001).

When
Adaptation may take place at very different times. **Design time** adaptation is often opposed to **runtime** adaptation. Some authors also differentiate static adaptation (before and between sessions) and dynamic adaptation (Dietrich & al., 1993). During a session, adaptation can be done continuously, at predefined junctures, before or after predefined functions, in special situations or on user’s request. In (Stephanidis & al., 1998) terminology, “adaptive” denotes adaptations that occur at runtime and that may be produced both by the system and by the user, and “adaptable” denotes adaptations before runtime, e.g., when the system is first installed (Kobsa & al., 2001).

Where
Adaptation can be internal or external. **Internal**, also called “closed”, indicates that adaptation mechanisms are embedded in the system itself while **external**, also called “open”, assumes that adaptation is done outside of the system, for instance with a Web service (Oreizy & al., 2004). Open adaptation mechanisms seem to be preferred in projects related to pervasive/ambient technologies because they allow the possibility to discover new services at runtime, and to do adaptation on the fly, in the service oriented architecture approach philosophy. We will return to that point in section 4.

To what
What is the target of the adaptation? In other words, what does the system adapt to? Indeed, it can be adapt to the **user**, to the **platform**, to the **environment** or a mix of them. Users can be seen as typical user, individual user, subcategories of users such as groups, categories, etc. Adaptation can also take place according to user’s roles, rights, skills, abilities, preferences, handicaps, culture…. Adaptation to the platform implies software and/or hardware adaptation. It means that the same application can be executed on e.g., different operating systems or physical devices, or that the application can adapt to the computing resources available. Adaptation to the environment requires means of sensing physical variables, such as location, ambient light, or temperature. When interfaces are able to adapt to the usage context, some authors talk about **plasticity** (Thevenin & Coutaz, 1999). The term **plasticity** is inspired from materials that expand and contract under natural constraints without breaking, thus preserving continuous usage.
How

The answer to the question “how to adapt applications?” is treated in the literature from two main points of view: strategies and methods. Four basic strategies of adaptation are proposed by (Cockton 1987): enabling, switching, reconfiguring, and editing. Enabling is adaptation by activation/deactivation of components and features in process control design systems. Switching is adaptation by selecting one of several different user interfaces, pre-configured user interface components, like dialog configuration, or user interface settings, like colors, font or size. Adaptation by reconfiguring is the modification of a user interface using pre-defined components. Editing is adaptation without any restrictions on the basis of the dialog model. Other features were added to this list: adjusting, transforming, altering/merging and exchanging/combining (Balint, 1995). Concerning the methods, there are four main ways identified in order to achieve an adaptation. It can be done by translation by reverse engineering or migration, by markup languages and by model-based approach.

The two first methods, translation and reverse engineering, are considered as bottom-up adaptations, because they use preexistent software or system and try to adapt them according to a new context, different from the original one. The two last methods, markup languages and model-based approach, are considered as top-down adaptations. They start from specifications and try to create interfaces adapted to a particular context. Obviously, a mix is possible when adaptation is made thanks to the two approaches presented above used conjointly.

We have presented different dimensions of adaptation. As we have seen, there is not yet a commonly adopted definition of adaptation. For us in this paper, adaptation is the ability for a system to reconfigure itself or otherwise perform actions in reaction to changes in the context of use, while preserving its usability.

The first works around adaptation where principally based on the four questions: What, When, Why and How to adapt? (Karagiannidis & al., 1996). Now, more criteria are used and design spaces for context-sensitive user interfaces are provided (Vanderdonckt & al., 2005), nevertheless it is difficult to choose a toolkit or architecture according to the needs of development, e.g. adaptation targets, models needed, distribution and dynamicity envisaged.

4. Plasticity of the user interface and context-aware adaptations

Developers have to design and realise interfaces having in mind many constraints such as production optimization, conception cost reductions, portability, scalability, reusability, quick prototyping, and easy maintenance etc. Currently, a few abstract representation languages have been developed in order to achieve this goal. The availability of many types of computers and devices has become a fundamental challenge for designers of interactive software systems. So, to reach N information through M peripheral is equal, for the developer, to write N * M programs, as we can see on Figure 2.
However, from the user point of view, the service offered should always be the same one, for example to reach an E-learning service. It does not matter that the user reaches this service by a telephone Wap, Palm Pilot, a PC connected to Internet, or any other means; its goal remains the same, and only the way to interact with the collaborative system is different. Instead of coding N * M applications, researchers try to offer one model for many interfaces (Paterno & Santoro 2002); see, for example, the Cameleon project for more details. Those kinds of languages allowed describing interface on an abstract way. This description is used to generate adapted interfaces. This factor is called plasticity of the user interface (Thevenin 2001). Plasticity is the ability of a user interface to be reused on multiple platforms that have different capabilities. The personalization may depend on many factors such as the client device, the user profile with roles, access rights, skills, abilities, handicaps, etc., the location, the access history, and so on.

For the World Wide Web, HTML combines data and presentation into one document. Oppositely, XML (Extensible Markup Language) provides separation between data and its presentation format. Languages like UIML (User Interface Markup Language) are used to describe and generate such kind of interfaces. UIML is an XML language used for defining the actual interface elements. This means the buttons, menus, lists and other controls that allow a program to function in a graphical interface like Windows or Motif.

It is important to identify the different elements which will take a part in the personalization. The three primary axes to investigate are the target level, the conception level, and the runtime level. Concerning the target level, there are three sub axes that can be followed: the user model, the device model, and the environment model. Concerning the conception level, tools of production can be qualified by two aspects: the specification of the models and the transformation of those models. The specification should describe the abstraction level (model or Meta model) and the scope of the specification. This scope could target one or many models among the concepts, the task, the user, the device, the environment, the evolution, and the transition. In addition, transition models are used to cover the transition between concepts to tasks. This in turn, will provide the final abstract interfaces, followed by the concrete interfaces and final interfaces. Concerning the runtime level, it is important to notice the capabilities of the interface to recognize context shifts, context capture, and identification of a particular context during the execution such as noise, light or network's break. At this level, the software must be able to recalculate a part or the entire interface, in order to present a user interface adapted to this new context.

---

1 See Cameleon project (http://giove.cnue.cnr.it/cameleon.html)
5. Adapting to what (slight return)

Among all the dimensions of adaptation presented on the map in Figure 1, one of the most important questions is: “Adapting to what?” We will illustrate some aspects of possible answers to that question, across different models such as user, platform, task or environment, used separately or concomitantly. Within the framework of the plasticity of the interfaces, we conceived and developed the Plastic ML (Rouillard 2003). This language allows developers to describe input and output elements, on a high level, without referring to the peripheral which will be used at the runtime. Thanks to XSLT transformations, the Plastic ML documents are automatically translated to markup languages: HTML for the Web, WML for the WAP, and VoiceXML for the phone. It is possible to declare inputs, outputs, choice, secret elements, on an abstract level, and thanks to XSL files, the interface will be displayed on the appropriate device.

5.1 Adapting to the platform

Let’s envisage a scenario where a developer wants to prepare a generic document, for instance, poetry, and would like to distribute this document among different devices. Plastic ML was used for our prototypes. Hence, a document based on XML is created, containing different tags: a title, a body, an author name, and a few other emphasized elements. Using XSLT transformation, it is possible to obtain an HTML document, which is viewed on a single scrollable page, all the poetry, with a picture of the author, and the other emphasized elements, in bold, as we can see on the left of the Figure 3.
The same transformation, based on another XSL document is possible in order to obtain a WML document. Each part of the poetry is sent to a suitable card, and the user can navigate from one card to another. No picture is created for this device, even if a WBMP transformation is possible, and emphasized elements are presented in bold font; we also tried underline, italic and other presentations that were available.

The right of the Figure 3 represents the result of the transformation for a cellular phone using WAP protocol. Contrary to the web version which comes with the picture of the author, here the presentation is simpler. The user can only see one strophe by card. The document presentation is guided by the device properties; a small screen for example. XSL transformation allows choosing the appropriate tag that match a specific device. For example, the tag <emphasis> of the original XML source document was transformed in <B>Helen</B> in HTML and <U>Helen</U> in WML.

Naturally, the presentation of the same document on a mobile phone will be entirely aural. We use the VoiceXML language to generate the document. Emphasized elements are presented with the <emp> tag of the VoiceXML language, which allows changing the sound of the TTS (Text to Speech) synthesis. It allows to make “voice effect”, lower or louder voice for instance, to express important words. Thus, speech synthesis plays a major role for a correct restitution of the text and so for a good understanding by the student. The user can also navigate through the document with the DTMF (Dual-Tone Multi-Frequency) buttons or by pronouncing oral commands such as “next” or “previous,” in a multimodal way.

5.2 Adapting to the user

One tag particularly used in Plastic ML documents is the <block> tag. Indeed, according to that block and the associated role, it is possible to show or hide certain parts of the original document. This feature is used in order to adapt documents according to the different models. For instance, a first transformation is made according to the role, and then a second adaptation is done according to the device used.

It is also possible to declare that the roles are not hierarchic for a determined block. If the structure is flat that means that roles are not dependants from each other. For example, in Figure 4, a French user will have the role number 1 and an English user will have the role number 2. They will see the “same” document, but it will be adapted to them, according to their chosen language. The system respects the semantic cutting and decomposition proposed by the author of the Plastic ML document. For example, if it's possible, all the information contained in a block will be sent to a card for a Wap phone device.

The role “ALL” will be used to present information, whatever the role. In our example, we must be careful to use words understandable in the two target languages, such as "introduction", "conclusion", etc.

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<plasticML version="2.0">
    <interface title="Adaptative Hypermedia, document fragmentation and interface plasticity">
        <block role="1" align="center">
            <output><important2>Hypermedia Adaptatif, document fragmentaire et interfaces plastiques</important2></output>
        </block>
        <block role="2" align="center">
            <output><important2>Adaptative Hypermedia, document fragmentation and interface plasticity</important2></output>
        </block>
    </interface>
</plasticML>
```
Figure 4: Plastic ML document integrating information blocks coupled with non-hierarchical roles

Figure 5 shows the result obtained in a Web page after processing of Plastic ML in HTML, with the role number 1. The text of the document is then displayed in a French-language version. More exactly, only fragments of the document source, whose blocks are marked as "role = 1" are used to dynamically rebuild the target document.

Figure 5: The document obtained for the role 1 presents only the French part

Figure 6 shows the result obtained in a Web page after processing of Plastic ML in HTML, but with the role number 2 instead of number 1. Hence, the text is written in English.

Figure 6: The document obtained for the role 2 presents only the English part

Obviously, adaptation could be made according to more than one factor at the same time.

5.3 Adapting to the task

Adapting to the task is not an easy problem, because tasks described, envisaged at the design time and tasks realized, detected at the runtime, are not always systematically the same. Carboni (2004) explains this point of view: “Someone having a breakfast in his kitchen would check his mailbox without turning on the computer but rather by means of his interactive TV.
So tasks models should be tailored according to the context of use, e.g. place, time, and device available” (Carboni 2004). Sometimes, the task requires the user to act with their hands. Hence, it’s more difficult for the user to interact with the interface without disturbing their activity. Figure 7 shows two examples where the hands of the user are required to execute the task; cooking or repairing a car, for instance. The interface is adapted to the user's language, but the aural channel is also used in order to take into account the fact that the user can not use the stylus during the entire session. Other parameters for the adaptation of the system can be used, such as the number of people available for a task. The instruction given to set up a furniture kit will not be the same for one, two, three, or more persons.

Figure 7: System adapted to the task, such as repairing a car or cooking

5.4 Adapting to the environment

As we previously explain, the context of use for a significant adaptation can be taken among the subsequent: user’s profile, current task, device used, location, time, and environment of the interaction. A context can be considered as shared or individual. Individual context includes information relevant to the interaction between the learner and the M-learning applications. For the learning domain, shared context includes information relevant to collaborative group work or learners sharing common interests. Individual context can be viewed as specifically tailored to each learner. Shared context is more relative to the collaborative work. The environment can be subdivided into many parts (physical, people, resources…). Smart systems are capable of detecting the environmental particular data influencing the task of the user. If the user is moving, the environment is too noisy or if there isn’t enough light to achieve a task, for example, the system will propose adapted solutions, in the respect of the usability rules.

5.5 Adaptation in ubiquitous learning

We are involved in researches on new interactive systems for ubiquitous e-Learning within the P-LearNet project (P-LearNet) which is supported by the ANR (Agence Nationale de la Recherche Française - National French Research Agency). P-LearNet means Pervasive Learning Networks. It’s an exploratory project on adaptative services and usages for human learning in the context of pervasive communications. The main goal of this project is to explore the potential of ubiquitous and pervasive communications, over heterogeneous networks, for a large and important field of application, e.g. human learning. To achieve this, we take into account the maximum available amount of information including places, times, organisational and technological contexts, individual and/or collective learning processes, etc.
We will now present some results of our work in adaptive pervasive learning environment, based on contextual QR Codes. This is where information is presented to learner at the appropriate time and place according to a particular task. (Specht & Zimmermann 2006) already showed the interest of a contextualization in the learning experiences. Five fundamental categories for context information were identified in (Zimmermann & Lorenz 2007) as follows: individuality context, time, location, activity, and relations context.

Figure 8: Public and private parts of a contextual QR Code

The notion of contextual QR Codes was proposed in previous recent work (Rouillard 2008). It can be defined as the following: it’s the result of a fusion between a public part of information, encoded in a 2D barcode named QR Code, and a private part of information, the context, provided by the device that scanned the code. Figure 8 shows the public and private parts of a contextual QR Code. The private part can be one or more of the following user’s profile, current task, device, location, time, and environment of the interaction. The mobile device decodes the QR Code and merges it with private data obtained during the interaction. Next, the XML (Extensible Markup Language) resulting file is sent to a web service created in our laboratory that computes the code and returns personalized messages. Some private information can be stored in the profile owner’s of the phone, the class level for example, and some others are given directly by the user when the interaction takes place. It’s the case for the language or the task chosen, for example. A previous study of people engaged in a location-based experience at the London zoo was reported by O’Hara and colleagues. In this experience, location-based content was collected and triggered using mobile camera phones to read 2D barcodes on signs located at particular animal enclosures around the zoo. “Each sign had an enticing caption and a data matrix code (approx. 7x7 cm) which encoded the file locations for the relevant media files.” (O’Hara & al., 2007). By capturing a 2D barcode, participants extracted the file’s URIs from the codes and added the corresponding preloaded content files (audio video and text) into their user’s collection. The fundamental distinction
between that approach and our system is that the London zoo system always provides the same content to the user, while the PerZoovasive system provides tailored information according to a particular context. Now let us present the system functionalities through the following scenario.

**Pedagogical scenario**

A French elementary school decides to organize a visit to the zoo. This visit allows a follow-up of the pupil’s educational curriculum. The subject is the different levels of pupils and number of teachers accompanying them. The teachers usually do not have an indepth knowledge of every subject. Question that could arise are: Do we have to display, disseminate, and post the same information to all pupils? How do we make sure each group of students gets the appropriate information? The accompanying teachers usually have a cellular phone. We propose to use the phone to provide adapted information. When the group gets to the zoo, each teacher’s cellular phone connects to a dedicated server where applicable software is access for set up. For instance, Mrs Martin is preparing a visit to a zoo for her CE1 class. She thinks it is a good idea to establish a link between the lessons that she gave in her classroom during the morning about the animals and their environments in the real world (see Freinet’s pedagogical method). She starts her visit with her group and stops in front of the Gibbons’ cage. She takes a photo of the QR code attached to the cage. This image is automatically sent to the server. Instantly, a text in French is adapted to the pupils’ level and a description of the Gibbon is then displayed on the teacher’s phone. The pupils then can listen to a synthesized voice deliver the information on Mrs. Martin’s mobile phone. At the same time, another class from an English school stops in front of the same cage. Mr. Ford, the teacher of this class (a different level than the French one) will perform the same operation as Mrs. Martin with his mobile phone. The text presented is adapted to the level, the language, and the task of these pupils. The two groups continue their visit and the same thing happens again in front of each subsequent cage.

**Technical scenario**

Figure 9 presents a mobile application, called PerZoovasive, developed in our laboratory. It was written in the language of C# and runs, thanks to Tasman library, on a smartphone HTC TyTN II, also known as “Kaiser”, supporting Windows Mobile 6. The name of the user and the level of the class are automatically detected in the mobile registry (Control Panel\Owner). The user clicks on the upper left radio button if the task is a lesson or on the upper right radio button if the task is a Quiz knowledge control.

In addition to the plain text, it is also possible to choose the flag corresponding to the appropriate language, and to select the TTS option in order to obtain a Text-To-Speech response. The trace option is a debugging tool that displays the XML code data sent and received. The camera manager is invoked by clicking on the “Capture QRCode” button. Each cage number is coded with a particular QR Code. Then, according to the selected representation, the decoded information is presented to the user. The application becomes extremely context-aware by using a combination of many parameters. The web service that receives the contextual QR Code opens a file which has the following canonical form: Cage_Level_Language_Task.txt. For example, the file named “8245_CM1_FR_Quiz.txt” is related to a Quiz in French, for the CM1 class (level 3) and for the cage number 8245.

---

2 Grade 2 in French elementary school (CP, CE1, CE2, CM1, CM2).
3 Tasman barcode recognition for developers (http://www.tasman.co.uk/index.html)
In Figure 9, since the teacher did not select the TTS option, she can read the data provided by the system on her mobile screen and give the information to her pupils as she sees it.

Figure 9: Example of generated text (cage=123, level=CM2, language=French, task=lesson)

Figure 10 shows a French-speaking teacher in a zoo, using the PerZoovasive application with a 3G connection (provided by Orange France) scanning a contextual QR Code in order to obtain information about gibbons (left) and turtles (right). After a few seconds, she can read her mobile screen and give relevant information about these animals to her pupils.

Figure 10: A teacher is scanning a Contextual QR Code to retrieve information about gibbons and turtles.

We can see the same teacher asking information about the turtle of another cage. She could retrieve general information such as specie, origin, speed or food, about this kind of animal, but also personal data about that particular turtle like its name, age, birthday, etc. Figure 11 shows the PerZoovasive application used by an English-speaking user. As she selected the Quiz task and the TTS option, she obtained on a multimodal way, that uses screen and speech, some questions about the panther, living in the cage number 8245, adapted to the level of her class.
6. Future Trends

The idea behind ubiquitous ICTs is the convergence, or even fusion of real-world objects with information systems. We are changing from “e-everything” to “u-everything”. Physical objects equipped with barcodes or RFID chips, for instance, will be reachable in a numeric world. Multimodal and multi-channel adaptation will be the kernel of future ubiquitous, plastic and intelligence systems. Our future work is oriented to the investigation of the traces/logs and original strategies to capture environmental information. This will lead to documents being capable of modification on the fly, but also will help us to better understand the behaviour of the entire system driven by multiple and sometimes contradictory policies. Technologies and solutions like IPv6, SOA (including web services approaches) are already used in order to deliver the appropriate information to the appropriate person at the appropriate place and time. But the challenge will be, for the next generation of systems, to adapt themselves smoothly to situations not yet encountered. This will be possible only with a real dialogue between the systems and the users. Hence, the future ubiquitous systems will have to show their seams, instead of trying to hide them, to facilitate their usage.

7. Conclusion

We have shown in this chapter that interface adaptation is a challenge to implement, accomplish, and manage. In view of the fact that for years, many studies and works in various domains have been attempting to expose the why, what, when, where and how to create an adaptable interface. With the emergence of ubiquitous computing, the difficulties are still growing. It’s not only a matter of transformation or conversion from a format to another. It also includes for the most part many semantic issues. Research oriented around pervasive and ubiquitous computing, especially in the field of human learning, works to create a model precisely for the user, the task, the device, which additionally includes some parts of the environment in which the interaction will take place. Moreover, relevant adaptations have to “understand” the meanings of the manipulated components. Now it is technically possible to change an element by another or to transform an aspect from an abstract level to a concrete one, yet meanings and intentions are crucial for relevant adaptations. Furthermore, that fact that people are mobile and want to use multimodal and multi-channel systems directs us to take into account real issues encountered by users in situ. The preliminary results of our P-
LearNet project shows that we have to deal with contents and containers simultaneously. Context-aware adaptations must be made, both in the respect of the usability of the systems and in the respect of the author intentions. We are confident that education will be increasingly performed across mobile devices, anytime and anywhere. But, the challenge is to offer intelligent and adapted tools, for cohabitation between smart people and smart environment.

**Acknowledgment**

We are grateful to ANR P-LearNet project for providing support for this research and to Tasman, for special tools provided. The author gratefully acknowledges M. Kent Washington for his help in improving this paper.

**References**


O'Hara, Kenton; Kindberg, Tim; Glancy, Maxine; Baptista, Luciana; Sukumaran, Byju; Kahana, Gil; Rowbotham, Julie. (2007). Collecting and Sharing Location-based Content on Mobile Phones in a Zoo Visitor Experience, *Computer Supported Cooperative Work (CSCW)*, Springer, Volume 16, Numbers 1-2 (34), (pp. 11-44).


Mampaey, M. (Eds) Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services, (pp. 153–166), Springer.


