Mobile Learning and Multimedia Educational Software Requirements' Evaluation

Giovanna Avellis
Technical Assistance and Consulting Division
InnovaPuglia SpA
Valenzano, Bari, Italy
g.avellis@innova.puglia.it

Abstract— Mobile Learning and Multimedia Educational Software play a key role in Educational Technologies. Their features have been identified and used to critically evaluate and trace them. One of the challenges faced by the developers of current Mobile Learning systems and Multimedia Educational Software is their assessment and traceability to meet users' requirements. This work is based on a technique to represent Requirements in Software Engineering in a Quality Function Deployment style [30]. We extend the model for the representation of design rationale [16] by making the evaluation goals explicit and providing the means to improve the quality of Mobile Learning and Multimedia Educational Software.

Keywords— Mobile Learning (m-Learning), Multimedia Educational Software (MES), Non Functioonal Requirements (NFRs), system architectures, evaluation, quality requirements, cooperation.

Introduction

Education is undergoing a major period of experimentation and transition, largely due to the significant development in computing technology. The need of educational multimedia for vocational training purposes for lifelong learning and on-the-job training in Small Medium Enterprises (SMEs) and large organisations is widely recognised. However, users of educational multimedia (teachers, trainers, educators, managers of educational settings) cannot appraise educational resources because they are not able to evaluate their characteristics, potentialities, and limits [38], [40].

Obviously, teachers and instructors must acquire a reasonable comfort level with computer technology in general to make effective use of Multimedia Educational Software (MES), but this is still not sufficient. It also is necessary that they acquire a thorough comprehension of what educational multimedia is, and be provided with a set of evaluation criteria in order to be able to identify educational multimedia suitable to their needs and valuable as educational resource to also achieve students' satisfaction [28].

The reason why it is not so easy to carry out a critical evaluation of educational multimedia lies in the problem these resources are relatively new compared to traditional print-based learning materials. Most people are still not used to handling them, nor they are aware of the educational potential.

In order that educational multimedia contribute positively into the educational process, it is important to consider quality requirements, that is Non Functional Requirements (NFRs) [18]. The need to have a framework for evaluating Collaborative Learning systems by analysing three quality dimensions, namely the educational, the economical, and the technical dimension, has been underlined in [18]. Content & activities, pedagogy & abilities, interaction & communications are the key issues to be addressed in the economical dimension. NFRs such as user interface, reliability, maintainability, performance, functionality, adaptation, connectivity and security are instead included in the technical dimension. Finally, the costs, contracts & licenses and cost-effectiveness have been encompassed in the economical dimension. The framework offers a decision support system for designers and developers and evaluators of Collaborative Learning systems in the same way in which we addressed this topic in our scheme of annotation of NFRs for traceability into architectures and, most important, to evaluation [11].

Finally, educational multimedia has also an intrinsic complexity for it encompasses two aspects: it is at the same time, a software running on a computer, and an educational resource. Evaluating both these aspects is very different from those when evaluating for example a book or any traditional educational resource. The blurred distinction between software and supporting learning complicates the assessment of its educational effectiveness as well as the educational purpose underlying the design of the software.

Further, it is difficult to develop predefined standards against which to assess the educational value of the software, because there is not a unique and general instructional approach. Thus, the educational value of MES is very difficult to define in practice.

A major problem in using computer-based educational programs is the difficulty of finding valuable software. Valuable software is available only to those who knows how to identify it. Therefore, evaluation of educational software is a fundamental step in the process of adopting it as a learning resource.

The educational potential of multimedia, both as a learning and teaching tool, is now widely acknowledged, and various initiatives undertaken by Governments all over Europe strongly encourage the integration of educational multimedia resources in the school practice [9], [10].

Mobile Learning (ML) is relatively new, so we are only beginning to see the potential of mobile devices in training and performance support. Although many experts in the field see great potential for the use of mobile devices in e-learning, there are presently very few successful implementations on which to base a study of best practices. Because of this, and the fact that some mobile devices are similar in functionality to conventional computers, it is only natural that the first generation of mobile learning will closely resemble conventional e-learning, only presented on a smaller screen. Mobile devices are small, portable and compact. They can often fit in a pocket or purse. Unlike laptop computers, which are expensive, heavy and power hungry, mobile devices are relatively low-cost, lightweight, and some work for a very long time on an electrical charge or using a couple of standard disposable or rechargeable batteries. The small screen size of mobile devices makes some people question their worth as e-learning delivery tools. Some of these devices also have good audio capability, allowing students to listen to a narrated lecture, rather than read material on a small screen. However, some critics do point to their restricted input capabilities of some of these devices, questioning students' ability to enter large amounts of text into a device in order to take notes or answer an essay-type question. Many of these devices are, however, extremely adaptable and can be attached to a full-size folding keyboard that makes entering large amounts of information every bit as fast as with a conventional computer.

As mobile devices evolve and people discover new ways in which the functionality of mobile devices can be applied to training, mobile learning will, most likely, become something increasingly different from conventional e-learning and no longer be a miniaturized version of conventional e-learning. Coupled with this, internet-connected phones may be applied to mentoring, and they may be used to register students on courses and pay their fees, as well as present training contents through the use of audio.

Another development may be that content development tools will become available that will provide the ability to publish learning content adaptively to a wide range of mobile devices. In addition, the student may well have control over reading or listening to the content using voice-synthesised XML technologies. Since mobile learning technology is so immature, there are presently more possibilities relating to what could be done with this technology than concrete examples.

However, with the number of mobile devices predicted to surpass the number of conventional computers for Web access in the near future, and with bandwidth for mobile devices predicted to increase dramatically in the short-term, mobile learning appears certain to become an important part of training in the future. Many national and international activities in mobile learning are currently partially funded by the European Commission, involving private and public sector organisations [12].

Many current applications of e-learning have highlighted the risks of learning in isolation while the advantages of the collaborative work areas in e-learning environments have often been underestimated. An important concept is also the one of developing a community of learners for building a Virtual Community. This is because in the majority of the cases, the focus of Technology Enhanced Learning (TEL) environments has been put more on the learning objects and the supported ontologies, rather than on the communication and interaction mechanisms. The key issue to investigate, besides the suitable mobile learning contents, is the interaction and communication itself among the learners, which can help the learner to maintain its social contacts, and by retaining the feeling of being part of a community.

Learning always takes place in a social context, as suggested by the socio-cognitive view of learning. In [15], we have addressed, more specifically, the issue of training Long Term Absents (LTAs) in a collaborative environment, where each LTA is contributing each other to the perceived need to exit from isolation and restructure one's cognitive schemas to better address the problems of the occupational stress. We embrace the view of learning of the *MOBILearn* project [31], [42], [44] putting the emphasis on its parts concerning rapid communication and access to resources via mobile technologies, and will test and evaluate the results of *m-learning* project [32], in particular the m-platform and the contents developed in the framework of the m-learning project for young adults training. Much research is being undertaken in general into learning styles, and in particular into the ways that these learning styles can be supported effectively by well-designed software systems. However, the educational success of educational products is still very varied, and unpredictable, relying largely on the expertise of the subject specialist and an intimate working relationship with the technologists and the specialist of the application area. This working relationship needs to be maintained if successful learning objectives are to be achieved. For this reason, the core objective of [15] research is to provide methodologies to ensure that the educational scope of learners is maintained throughout the application of TEL, in particular web learning.

Key users' requirements are Non Functional Requirements (NFRs). This is because functional requirements set out services expected by the system user whereas NFRs set out the constraints of the system and the product and process standards to be followed. As such, they play a central role in evaluating the quality of ML and MES.

There is a need to develop richer models for capturing, analyzing and assessing NFRs [24], [36]. However, this is not a simple enterprise. Examples of difficult tasks are as follows:

- assessing NFRs during system development
- choosing an architecture to satisfy some NFRs
- evaluating the impact of a change of NFRs on the system structure
- modifying the architecture

One of the open problems in our research is to map the NFRs to architectures to analyse the impact of changing the NFRs on the architecture. Understanding how proritisation and evolution of NFRs affects the requirements' traceability problem and choices of software architecture is another open issue to address. We refer with requirements' traceability [19] to the ability to describe and follow the life of a requirement, both forwards and backwards, through the design process. The requirements' traceability problem is perceived not to be uniform due to the several definitions and fundamental conflicts in [22], where it has been detected a lack of a common definition of the requirements' traceability (purpose-driven versus solution-driven versus information-driven versus direction-driven). The need for improved requirements specification traceability is reported in the literature [22]: the NFRs have yet to be incorporated at the core of product and process specification, design and implementation techniques and tools, and progress in this area has been limited.

Software quality[27] is gaining more attention for two reasons: on the technical side, it is usually not clear to those involved in the development how to measure the various quality criteria on a day-to-day basis (i.e. formative analysis), nor how to achieve and measure them on completion (i.e. summative analysis). On the customer's side, the issue is simply not knowing what to ask for. To this end a distinction has been made between basic quality factors, such as functionality, reliability, ease of use, economy, safety, and extra quality factors such as flexibility, repairability, adaptability, understandability, documentation and enhanceability [21].

The latter are quality factors related to the external, or observable, quality of a piece of software and are particularly important in the world of MES where technical strategies are emerging in parallel with educational and pedagogical strategies. However, it is important to grasp the internal quality of a system. Ultimately, the external quality of a system depends on its internal quality. For example, the enhanceability of a system is directly related to how well structured the internal design is, i.e. the size, definition and relationships between modules and subsystems. Internal quality factors include completeness, consistency, parsimony, traceability, rationality, structure, paradigm, and quality of algorithms and representations, as well as understandability and documentation. The nature of these factors is not well understood, which is why we propose to research how to evaluate quality factors in ML and MES, and apply the research results to several domains and scenarios to validate the scheme.

The aim of this paper is to address the main issues of assessing ML and MES and to tackle the problem of assessing and traceability of NFRs by developing a scheme, which also help to trace and annotate NFRs to the architectures.

Section 1 provides the evaluation methodology by addressing the m-learning and MES features as evolved from the ERMES ESPRIT project.

Section 2 introduces the tool support of the evaluation methodology above.

Section 3 points out the objectives of the research.

Section 4 presents the annotation scheme to the NFRs. It introduces the feedbacks from the users and the links to the SW architectures of the system under evaluation.

Section 5 gives some examples of application of the annotation scheme.

Finally, the conclusions and further work identify further research issues in building links between NFRs and system architectures.

I. The Evaluation Methiodology

What we mean by ML and MES assessment and traceability? When we use the word "assessment", we usually mean a judgement of merit or worth against a pre-defined set of standards or expectations. The evaluation process is used to assign a value to the "object" being evaluated, so that its worth or intrinsic value can be conveyed to others.

The term "assessment" as applied to educational multimedia acquires a different emphasis and covers a much broader range of activities. The process of evaluating ML and MES does not simply consist in assigning some intrinsic value to it, but in answering the many questions that arise considering its different aspects and facets.

There is no absolute answer to the very important question "is this multimedia software a better way of teaching and learning than using the traditional approach?": making the best use of what the software offers mostly depends on the instructor's or user's sense and experience, and on the context and educational setting in which it is used. For example, a teacher might use a multimedia educational title in order to achieve learning objectives that are not the same as those envisaged by the authors of the software. Or they might use only some parts of it and not others, or use it in environments and situations different from those in which the product was originally designed to function. Or a piece of software which has weak interactive features may still be considered useful as reference learning resource; another one lacking in accuracy in the definition of learning objectives may still be valuable thanks to its engaging game-like approach, and so on.

Therefore, the educational value of a piece of software, being it ML or MES, is something very difficult to define in practice [9], [7].

This will produce an integrated set of mobile learning training modules, and an analysis and assessment of evaluation criteria to understand their requirements for advanced mobile and wireless technologies. To this end, we will collaborate with current standardisation working groups, especially the evaluation and assessment of NFRs of mobile learning and e-learning modules. The current industry standards such as Aviation Industry CBT Committee (AICC), Instructional Management System (IMS), Dublin Core Metadata Initiative (DCMI), Institute for Electrical and Electronic Engineering/Learning Technology Standardization Committee (IEEE/LTSC), Information Society Standardization System/Learning Technology Workshop (CEN/ISSS LTWS), Alliance of Remote Instructional Authoring and Distribution Network for Europe (ARIADNE), PROmoting Multimedia access to Education and Training in EUropean Society (PROMETEUS) have already addressed the problem of metadata tagging of educational resources to allow easier access and retrieval through e-learning systems. Further improvements in standardisation could be achieved by extending the NFRs (eg. target delivery device) to include the set of characters currently adopted to describe and classify learning modules. This will result in an increased capability of the user to assess the suitability of selected educational material for a specific application environment (e.g. mobile learning).

II. Evaluation Features and Tool Support

Defining the object of evaluation is a key step in the process of evaluation, because it further involves developing criteria on the basis of which assessment is possible.

Therefore, it is important to define the domain of Multimedia Educational Software (MES) establishing by which features it can be identified, and to try to make a classification of the various types of software available for educational purposes.

What is Multimedia Educational Software? Multimedia Educational Software is a computer program, which performs a specific educational task, providing the users with knowledge about a specific topic or instructing them to acquire specific skills. The multimedia component can be identified in the use of a variety of media (text, graphics, animation, audio and video) to deliver instruction or support the learning activities. Multimedia educational software is also characterised by the presence of interactive components, which should enable the user to control the learning environment [10].

Our approach is based on the evaluation methodology adopted in the ESPRIT project ERMES [14] consists of identifying aspects of the object under evaluation, and then defining quality indicators in relation to these aspects. Defining the object of evaluation is a key step, because it suggests the evaluation criteria to be used. We group the characteristics of MES under the following four evaluation categories:

- educational features
- technical features
- aspects relating to the ease of use (usability)
- aspects relating to the content.

The aspects that fall under each one of these categories have been further grouped into sub-categories or sub-headings. For example, the educational features are divided into target users, pedagogical characteristics, instructional support materials and so on. That means that when evaluating the educational features of a multimedia product, the aspects relating to the target users, the pedagogical characteristics, the instructional support materials etc. have to be taken into account.

MES is a computer program which performs a specific educational task. The multimedia component can be identified in the use of a variety of media to deliver instruction or support for the learning activities. MES is also characterised by the presence of interactive components, which should enable the user to control the learning environment.

The evaluation criteria described in this section try to take into account all the elements most relevant to teachers and trainers, but they cannot be expected to consider and analyse the software form all points of view. It should be kept in mind that an educational multimedia may have such original characteristics as to prevent it from falling under a pre-defined set of evaluation criteria.

Features of mobile learning contents include the following:

- the *content* which has to be taught;
- the *delivery media* used to provide information;
- the *user interface* - that is the way the educational software presents itself to the user;
- the *interaction* devices - by which the user interacts with the computer, making choices, answering questions or performing activities, and is provided with *feedback* to each response;
- the instructional strategy adopted
- *access* - which refers to the navigational paths available to the user to reach the needed content
- *navigation* allowing the user to go from one piece of content to another
- *presentation* - which can provide guidelines for defining the visual communication strategies or presenting the content, navigation strategies and operation to the user
- *user operations* are those operations that are visible to the users and the only ones the user must be aware of
- *system operations* that are not visible to the users, but are essential in building user operation [20], [45].

We developed an Evaluation Tool on CD-ROM [14] accompanying the Evaluation Guidelines [9] providing an electronic version of the evaluation questionnaire. It enables users to complete the questionnaire more easily, and has the facility to automatically produce evaluation reports, based on the given answers.

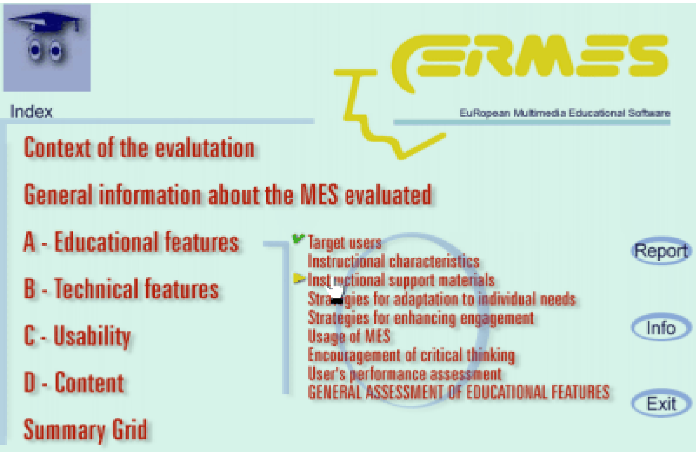


Figure 1

The first screen of the software presents the index of sections in which the evaluation questionnaire is structured. Clicking on any heading will bring up a further menu listing the content of each section (see Figure1).

Clicking on any of these sub-headings will bring up the screen containing the questions relating to that aspect of the educational software. The sections that have already been visited will appear ticked off in the index.

The ERMES icon in the top right corner gives access to the ERMES Web site, using the default browser installed on the PC (obviously this facility can run provided that a connection to the Internet is available). The icon in the top left corner giver access to the help on line. At the right of the screen there are three buttons:

- Clicking the REPORT button the user can produce the evaluation report.
- Clicking the INFO button brings up a window with more information about the software.
- Clicking the EXIT button the user can exit the software.

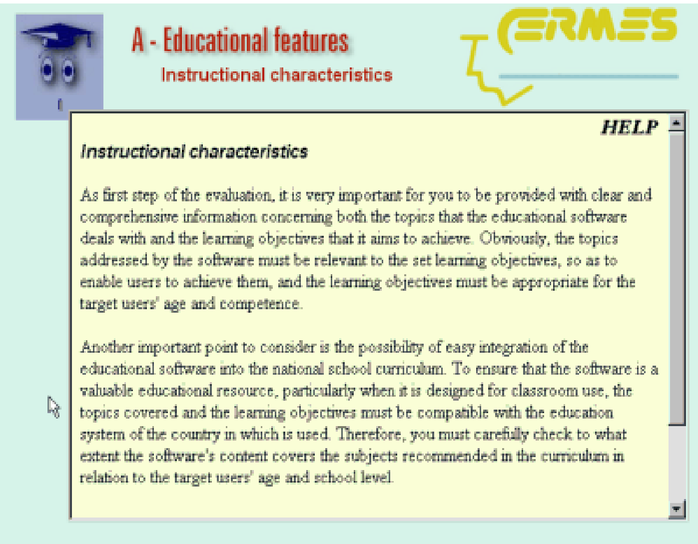


Figure 2

Clicking on the icon in the top left corner the user can access the online help (see Figure 2). A help balloon pops up at all times, containing information about how to move through the software. When the user calls up the help in a screen of the questionnaire, the help balloon contains a description of the aspect of the educational software currently under evaluation, so as to help the evaluator in answering the questions. The content of the help on line regarding the questionnaire is the same as that in the Evaluation Guidelines. Each section of the questionnaire is accessible through the index, by clicking on the relative heading and sub-heading. The evaluator can answer the questions clicking with the mouse. A block of questions can take more than one screen. The user is free to answer as many sections and as many questions as he or she likes. They can leave out questions that are in their opinion not relevant or clear answers to change them. If it is necessary to move to another screen to complete the section, the user must click on the arrow in the bottom right corner of the screen. Additional comments may be added using the space provided at the end of each section.

Clicking the INDEX button takes back to the Index. From the Index it is possible to access the Summary Grid (by clicking on the relative heading), in which the results of the general assessment of each section are automatically shown. Obviously, if the user has not made a general assessment of a section, the relative row is shown empty. After completing the questionnaire (or just a section or few sections of it), it is possible to automatically produce a report, in which the answers given to the questionnaire are expressed in full length. Clicking on the REPORT button in the Index screen brings up a window where short statements based on the answers given by the evaluator are shown. The window has the functionality of a text editor: the evaluator can make changes, additions and deletions to the evaluation report, and print it out. The evaluation report can also be sent via e-mail to the ERMES Web site for Users, clicking the button SEND BY E-MAIL. Evaluations sent by e-mail will be put online in the Showcase of Learning Materials, in order to provide useful information about the strengths and weaknesses of a particular piece of educational software. Publishers and software developers can also profitably use it to gather feedback about that software.

III. OBJECTIVES

Our research on traceability of NFRs in ML and MES has two main objectives. Tracing to the external factors, objective one is to assess the educational value in ML and MES by tracing users' NFRs to observable contextual behaviour, including the interacting devices, the user profiles and functional requirements; tracing to the internal factors, objective two is to justify the architectural design choices amongst large available MES according to the important NFRs for ML and MES.

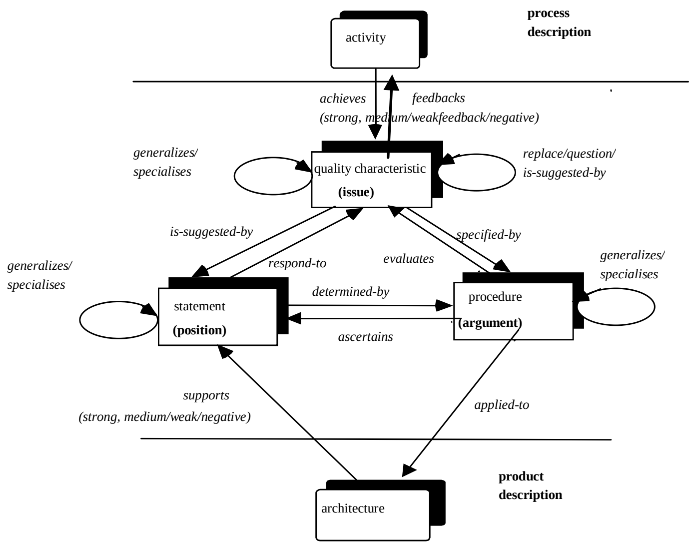
Objective 1. Tracing NFRs to contextual factors and making them exoskeletal. Unlike Functional Requirements (FRs) that prescribe the solution expected by a user, *NFRs*, such as usability, privacy/security and mobility, prescribe the quality attributes that are important for users to select from among solutions of the same functionality. Many external factors can influence the users' quality judgement, including the fitness of the running context, the satisfaction of users' skills and preferences. As such, contextual factors play a major role in evaluating the quality of systems especially because ML and MES interact more directly with end-users. ML and MES represent a broad class of software systems with complex characteristics that tend to make the one-size-fits-all evaluation difficult, also because there are no existing comprehensive frameworks for formative evaluation in MES, only limited frameworks have been developed for evaluation in specific contexts. The project will contribute mainly to the current state-of-the-art in providing methods and procedures to make the NFRs '*exoskeletal*', i.e. visible and tangible to the external users of software systems.

Objective 2. Tracing NFRs to architectural design factors and making them monitorable. It is widely recognised that early identification of architecture can assist in elicitation of detailed requirements, in design and reuse. Further, ML and MES development environments are usually populated with new technologies, tools and paradigms, which generate new NFRs and architectural styles in the ML and MES domains. The research will assess important NFRs against ML and MES architectural decisions and externalise controllable tuning parameters. By delaying a design decision to the runtime, the research aims to give end-users more freedom in reconfiguring the MES for the particular needs. Hence, feedback collected from monitoring the changed NFRs can propagate to the MES architectures, making MES more adaptive to the changing needs of end-users. The research aims to support traceability of NFR to the software architectures by applying Quality Function Deployment (QFD) [30] to assure that the user requirements, especially the NFRs, are traced in all development stages and thus sufficiently supported in the final system. This will place the "*generation of a value model*" such as is used in classical engineering disciplines, at the centre of the development, achieving a model of what is valued in the resulting system. As a result, quality characteristics are no longer externally imposed on a development process but "*constructed*" within it. Implementing this concept in the scheme to represent NFRs traceability contributes to leverage this research project at a high scientific and technological level in the current "state of the art"[22], [34]. [36]. Multidisciplinarity aspects of this project address several R&Ds fields, such as software quality, requirements engineering, software architectures, domain modelling, software maintenance, information retrieval, artificial intelligence, human computer interaction, and human learning [25], [26], [43]. These aspects are related to the schema for tracing the NRFs to architectures given in the next section. Our research novelty is to apply the research methodology of NFRs to a critical domain for future time-to-market applications, such as ML and MES. The main result of our research is a quantified NFR traceability to ML and MES to facilitate the evaluation of educational values of both ML and MES.

IV. The Scheme to Annotate NFRs

Techniques are needed to express NFRs, which include quality requirements [20]. The scheme developed to express NFRs is based on the work done by [29], particularly in the area of design rationale [16]. We also take into account the 'issue-position-arguments' model of [16], [41]. In our scheme, an 'issue', that is a problem to solve, is an 'NFR, or quality characteristics/sub-characteristics to evaluate'. An 'argument', that is, a supporting justification of the issue, is a *procedure* that helps to determine which design alternative to choose to implement in the related NFR. Finally, a 'position' that is a solution to the problem, is either a 'statement' of the NFR, which gives a quality goal to be supported by the final design, or 'design alternatives'. A statement is an ascertainable property (possibly measurable) characterising NFRs. The set of links is given in Figure 3.

Figure 3 Quality characteristics' representation schema



It is important to underline that the statement contains measurable elements by which the NFR can be 'constructed' in software systems. It is a procedure that applies to different architectural choices. In this way, we relate NFRs to architectures, by linking statements and different system architectural choices.

We have enhanced the representation of NFR with quality function deployment (QFD) features. Since the late 1960s, [30] have established a new systematic method of design-oriented approaches to ensure that customer needs drive the product design and production process. They developed a method called 'quality deployment and/or quality function deployment' (QD/QFD) [30]. We have enhanced the scheme of NFR representation by introducing the context of evaluation and weights to the links as in [5]. To be assured that we will achieve a particular software quality characteristic it is helpful to associate it with some activities within the software evaluation and development process. *Activity* is the evaluation and/or implementation activity of the quality characteristic that provides the context of evaluation. A quality characteristic is obtained in a *strong/medium/weak/negative* way as a result of performing an activity.

In a quality-function-deployment (QFD) style [30] we attach some weights –strong/medium/weak/negative – to this link, to let the end users (teacher, trainers, students, administrators) assign a weighted value to the characteristic of the system under evaluation.

Although a quality characteristic can be constructed independently of the description of the development process of a product, it is useful to link the product and process descriptions to the quality characteristics. [2] provides insights into how to relate this process view to a product view, by introducing the role played by the architecture of a software system and relating it to the NFRs. Although a quality characteristic can be constructed independently of the description of the development process of a product, it is useful to link the product and process descriptions to the quality characteristics [35]. We introduce the explicit representation of architecture in the annotation scheme of NFR given before in order to set a link between the process view and the product view of the software system under evaluation. The complete scheme for the representation of the links between NFRs and architectures, provides the explicit representation of the architectural description of the software system and new links to architecture and statement (position) and procedure (argument), as follows:

- *supports (*a statement is grounded on the specific choice of an architectural description and upkeep it. It becomes obsolete if the statement/position changes in the software system. An architecture can be chosen as the alternative which satisfy the statements in strong/medium/weak or negatively way, following the QFD style);
- *applied-to (*a procedure has to be implemented by the related architecture, that is the architecture accomplishes or neglects a given procedure both formal or informal, which can also be provided as argument during the evaluation to improve the current software system). This link is useful for several purposes such as reuse of design and requirements, reuse of design decisions and related architecture, explicit representation of rationale of an architectural description, evaluation of the architecture of a software system with respect to the non-functional requirements [2]. In the following, we illustrate the quality characteristics' representation schema with some examples.

In [6], we give an example of the application of the scheme to a Banking Information System that has undergone a re-engineering process to improve some NFRs, including its reusability.

We further have introduced to the representation schema of quality characteristics the following link between activity and quality characteristics:

- *feedbacks* (a quality characteristics evaluation can produce some results that need feedbacks from the end user performing the activity of evaluation/development in order to modify the subsequent activity of evaluation/development). This is to involve the end user in the evaluation of the quality characteristics more actively avoiding close loops/dead lock situations by asking the end user to perform a subsequent activity and intervene in the assessment process.

V. EXAMPLES

The first example is taken from [20]. Further examples are given for the domain of Mobile Learning and Multimedia Educational Software systems.

The **NFR** related to a software tool is "*the tool should be reliable "* . The **activity** is "*the evaluation of the reliability of the software tool under test*" which *strongly* **achieves** the related **quality characteristics** "*reliability*". This is further specialised into the sub-characteristics " *fault tolerance*".

The reason underlining this choice among the other sub characteristics of the reliability, it is suggested by the **requirement statement** "*the probability that software faults do not cause the tool to fail during a typical working session shall not exceed 0.99*". This can also be specialised into "*the probability that software faults do not cause the tool to fail during a typical working session, generally of a duration no greater than 8 hours, shall not exceed 0.99*"".

The **procedure** to measure software reliability is "Let F be the class of the faults, defined arbitrarily, and T be a measure of relevant time, the units of which are dictated by the package at hand. Then the reliability of the tool with respect to the class of faults F and to the metric T is the probability that no faults of the class occurs during the operation of the tool for a pre-specified period of relevant time."

This can be specialised in several procedures according to several models, such as *Fault Seeding Model*,[1] or *Fault Count Models*,[2] or *Time between Failure Model* [3]. The last one can

[1] The basic approach in this class of models is to "seed" a known number of faults in a program which is assumed to have unknown number of indigenous faults. The program is tested and the observed number of seeded and indigenous faults are counted. From these, an estimate of the fault content of the program prior to seeding is obtained and used to assess software reliability.

[2] The focus of this class of models is in the number of faults or failures rather than in times between failures. The failure counts are presumed to follow a known stochastic process with a time dependent discrete or continuous failure rate. Parameters of this failure rate can be estimated from the observed values of failure counts or from failure times. Estimates of software reliability can be obtained from the relevant equations.

further be specialised into other procedures such as *Littlewood&Verrall Bayesian Model*, the *Jelinski&Moranda De-Eutrophication Model*, or the *Goel&Okumoto Imperfect Debugging Model*. Other examples from the ML and MES domain are as follows.

The **NFR** of the second example is "the ML course should fit the subject/topics and learning objectives of my course".

The activity related to this example is "*evaluate the educational aim of the ML content*", which *strongly* achieves the **quality characteristics** "*educational features*" .

"*Educational features*" quality characteristics has several **sub-characteristics** to be taken into account, such as "*instructional characteristics*", which is suggested by the **requirement statement** "*appropriateness of learning objectives suitable for age and competence of target users*" and is measured by the **procedure** "*verify that content and learning objectives are consistent with the national curricula requirements*".

The NFR third example is "the MES package should be easy to operate".

The **activity** related to the third example is "*understanding the usage of a MES package*", which achieves in *medium* form the **quality characteristics** of "*usability*".

This in turn, can be further specialised into the **sub-characteristics** "*ease of use*", which is suggested by the **requirement statement** "*the way software operates*" and which is suggested by the requirements' statement 'the way software operates' and several procedures are used to measure usability: 'What are the IT skills required to operate the software? Is on-screen help available? Are directions clear and accurate? Are directions available at all times? Is the management of assessment instruments easy?

VI. Conclusions and Further Work.

The paper topic is in Computing and has its application domain in Education &Training (E&T) software systems to specifically address the problem of evaluation of educational multimedia and mobile learning.

We take a process-oriented view vs. current product-oriented views, being influenced by the work of decision support systems. Our approach extends the model for representing design rationale by making explicit the evaluation goals presupposed in the argument of the rationale representation, and provides the means to improve the quality of MES and ML.

Constructing, evaluating and evolving complex software systems to meet users requirements is one of the goal of software engineering technology [1].

This paper presents work in progress to improve the current assessment methodology based on the first results of the framework of the project ERMES [14]. The key issue is how to incorporate in the scheme the architectures to annotate NFRs to ML and MES. Further research is needed in this context as discussed above.

The innovative nature of this research is to apply evaluation and traceability methods and techniques from Quality Management, Software Engineering, Educational Technologies and HCI to a critical domain, such as ML and MES. This implies some *interdisciplinary* elements to be taken into account during the research, especially to address the educational features, in which the NFRs are very difficult to define in practice and to cope with, during their development and use in educational settings. Despite the amount of discussion, little research effort has been devoted to techniques to support both assessment and traceability of requirements, especially quality requirements, in ML and MES.

There is a need for a more comprehensive and structured approach to assessment based on sound and transparent principles. There are a variety of problems associated with assessing ML and MES, where the most fundamental problem is defining the characteristics of a good or acceptable assessment, though, of course, the issue of assessing ML and MES will also take us back to the issue of defining and conceptualising ML and MES.

Our research aims to enhance the current "state of the art" [17bis] by developing a representation scheme to evaluate and trace the requirements based on a design rationale and argumentation, and addressing increasing awareness of information, requirements evolution history, explanation, justification and change management, at a high technological level of the ongoing modelling, design and implementation techniques [1]. Our research will take a process-oriented view vs. current product-oriented views, being influenced by the work of decision support systems.

In the long-term future of Requirements Management Tools, it is emerging that the selection of a suitable architecture for a system is critically dependent upon the NFRs. Further, in the current "state of the art" of Requirement Management, the requirements are not organised so that the impact of changing a requirement on other requirements or on the system design can be determined [3], [4], [8].

It is widely recognised that NFRs are crucial in software development and that different architectural choices can have different impacts on the quality of the final system. However, there is a gap in the way current software development methods build and keep track of the links between requirements, especially NFRs, and architectures used in constructing and evolving complex systems. The aim of the project is to provide an explicit mapping between the NFRs and the systems and use the map, respectively, to reason on the "value" of a system, and to incrementally evaluate the NFRs during software development [7].

Further research will be performed to investigate the enabling technology to explore in this context the reuse of

---

[3] In this class of models the process under study is the time between failures. The most common approach is to assume that the time between, say the (i-1)-th and the i-th failure follows a distribution whose parameters depend on the number of faults remaining in the program during this interval. Estimates of the parameters are obtained from the observed

design, which leads to identify which components and relationships in the architecture satisfy the requirements, which architectures can be reused in an evolutionary change process, and which parts need incremental changes to derive improvement in the architectural artifacts.

The originality of the research is that here the research will focus on the high level part of this process, that is the analysis and reasoning on the process of building a "value" model of a software system, by explicitly adopting design rationale and quality management techniques to represent on NFRs.

The relationship between NFRs and architectures is an area of active enquiry research in the wider Software Engineering community. This project aims to contribute to this discussion and enhance the current "state of the art", by introducing novel approaches to represent the traceability links between NFRs and architectures, impact analysis of changes to NFRs to architectures, and finally a method to evaluate MES, and the suitability of architectural styles of MES with regard to NFRs.

The methodological approach to further research in order to achieve the objectives above is described as follows:

Objective 1 – Tracing NFRs to contextual factors. Assess the NFRs of several applications in the domain of ML and MES, obtaining general NFRs by mixing direct and indirect elicitation approaches. The elicited NFRs have the advantage to be analysed following Software Engineering best practices, such as goal-decomposition. The proposed evaluation methodology consists of identifying the goals under evaluation, and then defining quality indicators in relation to these goals. Defining the goals of evaluation is a key step, because it suggests the evaluation criteria to be used. Using the Problem Frames approach [23] [25], the characteristics of contextual factors in relation to the goals of ML and MES (i.e., context diagrams) can be grouped in categories, and sub-categories, such as educational features and usability.

Objective 2 – Traceability NFRs to ML and MES architectural design - Assess the architectural decisions in existing ML and MES systems, analyse how the documented architectural styles in the literature support the NFRs of ML and MES domain [34], and deploy NFRs monitors to collect end-users feedback. This can be a difficult task in the ML and MES domain because the architectures of ML and MES systems are usually not represented explicitly. The best approach is to identify them by comparison with documented architectural styles in the literature. The aim is to investigate ML and MES with similar NFRs and architectures, and to search for evidence whether there are common challenging ones that characterise the chosen domain. Based on these results, the project will develop a lightweight but effective method to address the evaluation of style suitability for fulfilling NFRs, based on innovative techniques from quality management and design rationale. Finally, the research will design a novel scheme for representing the traceability of NFRs, in particular focusing on the links between NFRs and architectures, by using semantic hypertext representation and the most innovative techniques from HCI, and the monitoring and diagnosis theory in AI [17]. The method to analyse the suitability of architectural styles, which fulfil a given NFR, is an important project result, which can serve a broader community including e-learning planners, managers, architects and developers. The architectures of ML and MES will be shown to exhibit characteristics of various architectural styles. By analysing how these styles support the NFRs, the project can identify those styles that offer the "best-fit" and provide guidelines for the engineering of MES. Validation of this approach will be carried out by conducting key case studies in Europe. This has the advantage to test the evaluation and traceability of NFRs in industry and SMEs interested to exploit the project results with respect to their applicability in the near future time-to-market products [37]. Feedbacks will be taken into account from the case studies to improve the evaluation methodology, the scheme and methods above.

ACKOWLEDGEMENT

This work has been partly financially supported by the European Commission under the Human Capital and Mobility scheme and the projects ERMES (EuRopean Multimedia Educational Software) [14] and MACS (Maintenance Assistance Capabilities for Software) [3], [4], [8], [13], and MOBITECH (European SMEs Challenge in Mobile Communication Technologies) [33].

The author acknowledges Prof. Bashar Nuseibeh, Prof. A.M. Kukulska Hulme, Prof. Yijun Yu, Prof. Josie Taylor Open University, and Prof. Anthony Finkelstein, University College of London, for their support and encouragement to this research,

REFERENCES

[1] G. Arango and R Prieto-Diaz R., Domain Analysis and Software System Modelling, California, IEEE CS Press, (1991).

[2] G.Avellis, *Annotating multimedia educational software with non functional requirements: an approach to evaluate educational multimedia.* University College London internal report (2000).

[3] G.Avellis "MACS ESPRIT project and the software evolution expert system", In *Annual Conf. of Associazione Italiana per l'Informatica e il Calcolo Automatico,* Bari: Laterza Publishing Company, (1), pp. 3–7, (1990).

[4] G.Avellis "CASE support for software evolution: a dependency approach to control the change process", In G Forte, N Madhavji, HA Muller (eds), *Proc, of the Fifth International Workshop on Computer-Aided Software Engineering*, Montreal: IEEE CS Press, pp. 26–73, (1992).

[5] G.Avellis, Enhancing Quality of Learning Contents with Quality Function Deployment, in *Proceedings of iLearn2008 Conference*, Palais des Congres, Paris, 4-5 February (2008).

[6] G.Avellis, Traceability of Non Functional Requirements to IS architectures, In *Proceedings of the IADIS International Conference e-Society 2008.* Algarve, Portugal, 9-12 April (2008).

[7] G.Avellis, Evaluation of Mobile Learning Contents and Mobile Services and Applications, in the Book *The Evolution of Mobile Teaching and Learning*, Retta Guy editor, Information Science Press, 249-269, (2009).

[8] G. Avellis and L. Borzacchini, A blackboard model to design integrated intelligent software maintenance environment, In *Proceedings of the Fourth International Conference on Software Engineering and Knowledge Engineering*. Capri: IEEE CS Press, pp. 325–332, (1994).

[9] G.Avellis and M Capurso "ERMES evaluation methodology to support teachers in skills development". In E Tuomi, M Salonen, P Saarinen, M Sinko (eds), *Proceedings of IFIP WG3.1 and 3.5 Open Conference on*

Communications and Networking in Education: Learning in a Networking society. Hameelinna: Painopaikka Yliopistopaino, pp. 12–19, (1999).

[10] G. Avellis and M. Capurso "ERMES Services for Education", In J Roger, B. Stanford-Smith, P.T. Kidd (eds), *Business and work in the information society: new technologies and applications.* Stockholm: IOS Press, 520–527, (1999).

[11] G.Avellis, A.Finkelstein, How to Annotate Educational Multimedia with Non Fnctional Requirements. Educational Technology & Society, vol.5, (2002).

[12] G.Avellis and A.Fresa "Education and training: a challenge between industries and cultures", In J Roger, B Stanford-Smith, PT Kidd (eds), *Business and work in the information society: new technologies and applications*. Stockholm: IOS Press, pp.401–407, (1999).

[13] G Avellis., A Iacobbe, D Palmisano, G Semeraro, C Tinelli. "An analysis of incremental capabilities of a software evolution expert system". In *Proceedings of Conference on Software Maintenance, 1991*. Sorrento: IEEE CS Press, pp.220–227, (1991).

[14] G. Avellis and A Ulloa S.. ERMES technical annex. *ESPRIT 24111 Project*, ERMES Consortium (eds), (1997).

[15] G.Avellis, F.Zaccaro, G. Tanucci, C.Epifani, *Mobile Technologies in Education and Training: Evaluating Mobile Learning Contents for Occupational Stress,* In Proceedings of the 2005 WSEAS, International Conference on ENGINEERING EDUCATION, Vouliagmeni, Athens, Greece, July 8-10.336-341, (2005).

[16] J. Conklin, ML. Begeman, "gIBIS: a hypertext tool for exploratory policy discussion". *ACM Transactions on Office Information Systems, 6(4)*, pp.303–331, (1988).

[17] P. Devanbu, R. Brachman, P. Selfridge, B. Ballard, "LaSSIE: a knowledge-based software information system", In Communications of the ACM, 34(5), pp. 34-49, (1991).

[18] A.A. Economides, Evaluation of Collaborative Learning Systems, In Proceedings of the 2005 WSEAS, International Conference on ENGINEERING EDUCATION, Vouliagmeni, Athens, Greece, July 8-10, 169-175, (2005).

[19] A. Finkelstein, "Tracing back from requirements", In IEEE Colloquium on Tools and Techniques for Maintaining Traceability During Design, Computing and Control Division, Professional Group C1, Digest No 1991/180, (1991).

[20] A. Finkelstein, "Quality arguments: product and process interaction". In *Proceedings of Software Quality Management '94,* (1994).

[21] P.Freeman, Software Perspectives – The System is the Message, Addison-Wesley Publishing, (1987).

[22] O.Gotel, A. Finkelstein, "An analysis of the requirements traceability problem", in Arnold R. and Bohner S. eds, Software change impact analysis, IEEE CD Press, (1996).

[23] M. Jackson, Problem Frames: Analyzing and structuring software development problems, Addison-Wesley, (2001).

[24] D. Haley, B. Nuseibeh , H. Sharp, J. Taylor, "Managing Requirements for Mobile Learning", Technical Report N2004/18, Dept. of Computing The Open University, (2004).

[25] J.Hall, M.Jackson, R.Laney, B.Nuseibeh, L.Rapanotti, Relating Software Requirements and Architectures using Problem Frames, In Proceedings of IEEE International Requirements Engineering Conference (RE'02), Essen, Germany, 9-13 September, (2002).

[26] M. Harandi and J. Ning. "PAT: a knowledge-based program analysis tool". In Proceedings of Conference on Software Maintenance 99. Phoenix, IEEE CS Press, 312-319, (1988).

[27] ISO/IEC 9126: Information technology - Software Product Evaluation - Quality characteristics and guidelines for their use - http://www.iso.org., (1991).

[28] C.Kolias, Evaluating Students' Satisfaction: The Case of Informatics Department of TEI Athens, In Proceedings of the 2005 WSEAS, International Conference on ENGINEERING EDUCATION, Vouliagmeni, Athens, Greece, July 8-10, 120-129, (2005).

[29] W. Kunz, H. Rittel. "Issues as elements of information systems". *Technical Report S-78-2, Institut fur Grundlagen der Plammung*. Stuttgart: Universitat Stuttgart, (1970).

[30] S. Mizuno and Y. Akao. Quality function deployment. *Nikkagiren-Syuppan*(ed.), Tokyo, Japan, (1978).

[31] MOBILearn Consortium, MOBILearn Technical Annaex (2002).

[32] m-learning Consortium, m-learning Technical Annex (2002).

[33] MOBITECH Consortium, MOBITECH Technical Annex, (2005).

[34] B. Nuseibeh: "Weaving Together Requirements and Architectures". IEEE Computer 34(3): 115-117 (2001).

[35] C. Potts, G. Bruns, "Recording the reasons for design decisions. *Proceedings of the Tenth International Conference on Software Engineering*. Singapore, IEEE CS Press, (1988).

[36] S. Roberstson, J. Roberston, Mastering the Requirements Process. Harlow, England, Addison-Wisley (1999).

[37] E. Scanlon, A. Jones., J. Barnard, J. Thomson and J.Calder, "Evaluating information and communication technologies for learning", Educational Technology&Society 3(4) 2000 ISSN 1436-4522, (2000).

[38] C.Sgouropoulou, I. Chalaris, M.Fouka, Bridging Theory and Practice for Systematic Learning Design, In Proceedings of the 2005 WSEAS, International Conference on ENGINEERING EDUCATION, Vouliagmeni, Athens, Greece, July 8-10, 336-341, (2005).

[39] H. Sharp, J. Taylor, A. Lober, D., Frohberg, D. Mwanza., E. Murelli., "Establishing user requirements for a mobile learning environment", Proceedings of Euroscom 2003, 29 Sept.-1Oct 2003, Heidelberg, Germany, (2003).

[40] M. Sharples, N. Jeffery, J. du Boulay, D. Teather, B. Teather, and G. du Boulay, G. "Socio-CognitiveEngineering: A Methodology for the Design of Human-Centred Technology," *European Journal of* Operational Research (Elsevier) 136(2), pp. 310- 323, (2002).

[41] S.B.Shum, Hypermedia Discourse: Contesting Networks of Ideas and Arguments, u.Priss, S.Polovina, R.Hill (eds.), ICCS 2007, LNAI 4604, 29-44, Spinger-Verlag Berlin Heidelberg, (2007).

[42] J.Taylor, *A task-centred approach to evaluating a mobile learning environment for pedagogical soundness*, In Learning with Mobile Devices - research and development, LSDA Agency, London, 13-20, (2003).

[43] J. Taylor, and D. Evans, "Pulling together: keeping track of pedagogy, design and evaluation through the development of scenarios – A case study", *Learning, Media and Technology,* 30(2), 131-145, (2005).

[44] J.Taylor, M.Sharples, P.D. Rudman, *Towards a Task Model for Mobile Learning: A Dialectical Approach,* International Journal of Learning Technology, Special Issue:"Interactions, objects and outcomes in learning", P.McAndrew and A. Jones (eds.), (2006).

[45] UWA Consortium, The UWA approach to modelling ubiquitous web applications. In the Proceedings of IST Mobile and Wireless Telecommunications Summit 2002:Towards Ubiquitous Communications, Thessaloniki, Greece, (2002).