

Non Functional Requirements Assessment and Traceability in Multimedia Educational Software

Giovanna Avellis

Technical Assistance and Consulting Division

InnovaPuglia SpA
Valenzano, Bari, Italy

Abstract—One of the challenges faced by the developers of current Multimedia Educational Software (MES) is the assessment of MES to meet users' requirements. Non Functional Requirements (NFR) are key users' requirements, because unlike functional requirements, which set out services expected by system, NFRs set out the constraints of the system and the process standards to be followed. Therefore, they play a key role in assessing the quality of MES. Quality Engineering is a new paradigm playing a crucial role in the European market of Services and Applications. In this field the assessment of Non Functional Requirements (NFRs) is underlined as an important step to have time-to-market products satisfying the quality and user needs.

We aim to define a methodology and a representation of NFRs for assessment and traceability management. We have developed a scheme for representing critical NFRs, and applied it to the domains of MES for validation. Our approach extends the model for the representation of design rationale by making the evaluation goals explicit and providing the means to improve the quality of MES. Further research is how to relate NFRs to architectures.

Keywords— Non-Functional Requirements (NFRs), Multimedia Educational Software (MES), system architectures.

I. INTRODUCTION

Multimedia Educational Software (MES) adds intrinsic complexity to software engineering [24] because it involves interacting both with software that runs on a computer and with educational resources that may be obtained by using the software. Evaluating both aspects is very different from evaluating any traditional educational resource such as books because of the interleaving aspects of software and learning 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.

Non-functional requirements (NFRs) are crucial in the development of software and different architectural choices can have a different impact on the quality of the final system [1], [2], [12]. There is a perceived gap in the way that current software development methods build on, and keep track of, the links between such requirements, especially NFRs, and system architectures in constructing and evolving complex systems. In this paper, we provide a map to help identify the explicit links between the NFRs and MES, and use this map to consider the 'value' of the system and incrementally evaluate the NFRs during software development. We focus on the analysis of, and reasoning about, the process of building a 'value' model of a software system by explicitly representing NFRs. The techniques and representations in the paper are then demonstrated by using an application from the domain. MES systems represent a broad class of software systems with complex characteristics that tend to make evaluation difficult. The educational potential of multimedia, both as a learning and teaching tool, is widely acknowledged, and various initiatives undertaken encourage the integration of educational multimedia resources in school practice [7].

The aim of this paper is to address the main issues of assessing MES and to tackle the problem of assessing NFRs by developing a scheme for annotating NFRs to the architectures.

Section 1 provides the context of the problem, or the background.

Section 2 underlines the approach and points out the objectives of the research.

Section 3 introduces the first results, that is the annotation scheme to represent the NFRs, which is a process-oriented, rather than product-oriented, representation. It gives some examples of application of the annotation scheme to MES.

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

II. BACKGROUND

There is a need to develop richer models for capturing, analyzing and assessing NFRs [17], [23]. 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 prioritisation 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 [13] 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 [15], 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 [18]: 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 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.

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 MES, and apply the research results to several domains and scenarios to validate the scheme. Many national and international activities in multimedia educational software (MES) in general are currently partially funded by the European Commission, involving private and public sector organisations [8]. In this context, the need for educational multimedia for vocational training purposes is widely recognised. However, users of educational multimedia cannot appraise educational resources because they are not able to evaluate their characteristics, potentialities and limits [6].

III. APPROACH AND OBJECTIVES

Our approach is based on the evaluation methodology adopted in the project ERMES [10] 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 multimedia educational software under the following four evaluation categories:

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

Each one of these categories has been further divided into sub-categories. For example, educational features can be divided into target users, pedagogical characteristics, instructional support materials, and so on. That means that when evaluating the educational features of MES, the aspects relating to the target users, the pedagogical characteristics, the instructional support materials, and so on, all 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 technical criteria follows the evaluation guidelines suggested by ISO/IEC 9126, as a MES is a computer program itself. But to evaluate a MES title is also important to consider some features as:

- Presentation of the packaging
- Graphical aspects
- Aesthetical aspects
- Presence of credits
- Language and text characteristics
- Printing options
- User control
- Supporting documentation
- Assistance

and the features are defined in term of the sub-characteristics, such as:

- *Quality of the user interface*: a good user interface does not overloads the learner with more problems than he/she really has for the fact he/she is learning a new matter. Messages should be clear and expressed using user language, not information technology concepts. Error messages should suggest corrective actions.

Among educational features are the following:

- The user's previous knowledge
- Learning objectives
- Information clarity
- Flexibility
- Access to specific contents

with sub-characteristics such as:

- The user's previous knowledge.
 - *Description of the user profile.* Information on the approximate age of the users, the level of previous knowledge regarding the topic in question, and other aspects that can help in defining the public at which the product is directed.
 - *Initial evaluation.* A good MES should include an initial evaluation of the knowledge and abilities of the user with respect to the topic with which he or she is being presented.

The content and indicator features refer to how the subjects in question are dealt with in a specific MES. This block of features provides a vision of the contents presented within the discipline to which they belong. A product may be flawless on a technical, usability and educational level; however, it may be poor with respect to its contents. Learning could thus be insufficient or could contain conceptual errors. Therefore, it is very important to consider the study of the contents as a specific analysis dimension. Some examples of content features are:

- Structure
- Index
- Modular Structure
- Interest
- Current Information

with sub-characteristics such as:

- Structure.
With respect to this criteria we can observe the following indicators:
 - *Structure.* This is to check to see if the fragmentation of the contents between the different chapters or sections is correct in order to facilitate its progressive assimilation.
 - *Correction.* The presented contents must not contain errors.

Our research on traceability of NFRs in MES has two main objectives. Tracing to the external factors, objective one is to assess the educational value in 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 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 MES interact more directly with end-users. MES represents 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.

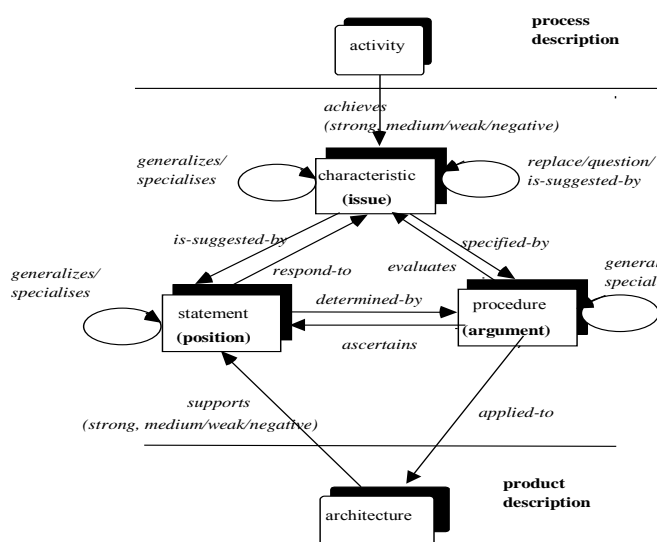
Objective 2. Tracing NFRs to architectural design factors and making them monitorable. It is widely recognised that early identification of architecture can assist in elicitation of detailed requirements, in design and reuse. Further, MES development environments are usually populated with new technologies, tools and paradigms, which generate new NFRs and architectural styles in the MES domain. The research will assess important NFRs against 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) [20] 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".

Multidisciplinary 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], [27]. These aspects are related to the schema for tracing the NFRs 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 MES. The main result of our research is a quantified NFR traceability to MES to facilitate the evaluation of educational values of MES.

IV. HOW TO ANNOTATE NFRs TO MES

Techniques are needed to express NFRs, which include quality requirements [14]. The scheme developed to express NFRs is based on the work done by [19], particularly in the area of design rationale [21]. We also take into account the ‘issue-position-arguments’ model [11]. 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 1.

Figure 1 Non Functional Requirements 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 [20] 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). We have enhanced the scheme of NFR representation by introducing the context of evaluation and weights to the links as follows. 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 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 [22]. 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 some examples.

Here are two examples of the application of the scheme above to MES.

An NFR related to a MES could be: ‘the MES should fit the subject/topics and learning objectives of my course’. The activity related to this example is to: ‘evaluate the educational aim of the MES package’, which strongly achieves the quality characteristics ‘educational features’. ‘Educational features’ quality characteristics have several sub-characteristics to be

taken into account, such as ‘instructional characteristics’, which suggest by their requirement statement that ‘appropriateness of learning objectives are suitable for the age and competence of target users’ and this is measured by a procedure to ‘verify that the content and learning objectives are consistent with the national curricula requirements’

The second example is the NFR ‘the MES package should be easy to operate’. The activity related to this 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 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?’

V. CONCLUSIONS AND FURTHER WORK.

This paper presents work in progress to improve the current assessment methodology based on the first results of the framework of the project ERMES [10]. The key issue is how to incorporate in the scheme the architectures to annotate NFRs to 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 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 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 MES, where the most fundamental problem is defining the characteristics of a good or acceptable assessment, though, of course, the issue of assessing MES will also take us back to the issue of defining and conceptualising MES. Our research aims to enhance the current “state of the art” 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. Our research will take a process-oriented view vs. current product-oriented views, being influenced by the work of decision support systems. The research approach aims to extend the model for representing design rationale by making explicit the evaluation goals presupposed in the argument of

the rationale representation, and providing the means to improve the quality of the system.

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. 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. Further research will be performed to investigate the enabling technology to explore in this context the reuse of 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 MES, obtaining general NFRs by mixing direct and indirect elicitation approaches. The elicited NFRs have the advantage to be analysed following Software Engineering (SE) 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 [16], the characteristics of contextual factors in relation to the goals of MES (i.e., context diagrams) can be grouped in categories, and sub-categories, such as educational features and usability.

Objective 2 – Traceability NFRs to MES architectural design - Assess the architectural decisions in existing MES systems, analyse how the documented architectural styles in the literature support the NFRs of MES domain [21], and deploy NFRs monitors to collect end-users feedback. This can be a difficult task in the MES domain because the architectures of MES 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 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. 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 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. Feedbacks will be taken into account from the case studies to improve the evaluation methodology, the scheme and methods above.

ACKNOWLEDGEMENT

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) [10] and MACS (Maintenance Assistance Capabilities for Software) [3], [4], [5], [9]. The author acknowledges Bashar Nuseibeh, A.M. Kukulska Hulme, Yijun Yu, Open University and Anthony Finkelstein, University College of London, for the support to the first results of the 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, H.A. Muller (eds), *Proc. of the Fifth International Workshop on Computer-Aided Software Engineering*, Montreal: IEEE CS Press, pp. 26–73, (1992).
- [5] 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).
- [6] 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*. Helsinki: Painopaikka Yliopistopaino, pp. 12–19, (1999).
- [7] 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).
- [8] G. Avellis and A. Fresa “Education and training: a challenge between industries and cultures”, In J. Roger, B. Stanford-Smith, P. T. Kidd (eds), *Business and work in the information society: new technologies and applications*. Stockholm: IOS Press, pp. 401–407, (1999).
- [9] 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).
- [10] G. Avellis and A. Ulloa S. *ERMES technical annex. ESPRIT 24111 Project*, ERMES Consortium (eds), (1997).
- [11] J. Conklin, M. L. Begeman, “gIBIS: a hypertext tool for exploratory policy discussion”. *ACM Transactions on Office Information Systems*, 6(4), pp. 303–331, (1988).
- [12] 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).
- [13] 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).
- [14] A. Finkelstein, “Quality arguments: product and process interaction”. In *Proceedings of Software Quality Management '94*, (1994).
- [15] 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).
- [16] M. Jackson, *Problem Frames: Analyzing and structuring software development problems*, Addison-Wesley, (2001).
- [17] D. Haley, B. Nuseibeh, H. Sharp, J. Taylor, “Managing Requirements for Mobile Learning”, Technical Report N2004/18, Dept. of Computing The Open University, (2004).
- [18] 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).
- [19] W. Kunz, H. Rittel. “Issues as elements of information systems”. *Technical Report S-78-2, Institut für Grundlagen der Planung*. Stuttgart: Universität Stuttgart, (1970).
- [20] S. Mizuno and Y. Akao. *Quality function deployment. Nikkagiren-Syuppan* (ed.), Tokyo, Japan, (1978).
- [21] B. Nuseibeh: “Weaving Together Requirements and Architectures”. *IEEE Computer* 34(3): 115–117 (2001)
- [22] C. Potts, G. Bruns, “Recording the reasons for design decisions”. *Proceedings of the Tenth International Conference on Software Engineering*. Singapore, IEEE CS Press, (1988).

- [23] S. Roberston, J. Roberston, *Mastering the Requirements Process*. Harlow, England, Addison-Wisley (1999).
- [24] 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).
- [25] M. Sharples, N. Jeffery, J. du Boulay, D. Teather, B. Teather, and G. du Boulay, G. "Socio-Cognitive Engineering: A Methodology for the Design of Human-Centred Technology," *European Journal of Operational Research* (Elsevier) 136(2), pp. 310- 323, (2002).
- [26] H. Sharp, J. Taylor, A. Lober, D., Froberg, D. Mwanza., E. Murelli., "Establishing user requirements for a mobile learning environment", *Proceedings of Euroscm 2003*, 29 Sept.-1 Oct 2003, Heidelberg, Germany, (2003).
- [27] 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).

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US