The London Charter and the Seville Principles as sources of requirements for e-archaeology systems development purposes

Juan M. Carrillo Gea, Ambrosio Toval, José L. Fernández Alemán, Joaquín Nicolás y Mariano Flores

Software Engineering Research Group, Departamento de Informática y Sistemas de la Universidad de Murcia. España.

Resumen

La Ingeniería de Requisitos (IR) es una disciplina de importancia crítica en el desarrollo de software. Este artículo proporciona un proceso y un conjunto de artefactos software para ayudar en la producción de sistemas de e-arqueología con énfasis en reutilización de requisitos y estándares. En particular, dos guías relevantes en el campo de la e-arqueología, la Carta de Londres y los Principios de Sevilla, se han mostrado como dos fuentes de requisitos a tener en cuenta como punto de partida para el desarrollo de este tipo de sistemas.

Palabras Clave: *DESARROLLO DE SISTEMAS DE E-ARQUEOLOGÍA, INGENIERÍA DE REQUISITOS, REUTILIZACIÓN, ESTANDARIZACIÓN.*

Abstract

Requirements engineering (RE) is a discipline of critical importance in software development. This paper provides a process and a set of software artefacts to help in the production of e-archaeology systems with emphasis on requirements reuse and standards. In particular, two important guidelines in the field of e-archaeology, the London Charter and the Principles of Seville, have been shown as two sources of requirements to be considered as a starting point for developing this type of systems.

Key words: *E-ARCHAEOLOGY SYSTEMS DEVELOPMENT, REQUIREMENTS ENGINEERING, REUSE, STANDARDIZATION.*

1 INTRODUCTION

Requirements engineering (RE) proposes the use of repeatable and systematic procedures to ensure obtaining a set of requirements (reqs) which results relevant, complete, consistent and easily understandable and analysable by different stakeholders. According to the Standard Glossary of Software Engineering Terminology IEEE 610, a req is: (a) "a condition or capability needed by a user to solve a problem or achieve an objective"; (b) "condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents"; or (c) "a documented representation of a condition or capability as in (a) or (b)".

During the RE process, the following activities are performed cyclically: (1) identification and consensus (elicitation), (2) analysis and negotiation; (3) documentation (specification); and (4) validation of reqs. Also, an additional activity, reqs management, is often characterised, which refers to schedule, coordination and documentation of the other activities, controlling the changes in reqs.

The main objective of RE is to specify what a system has to do, and the design constraints that determine how it should be implemented, with the aim of developing correct software, i.e. software that works according to the customer needs. Hence it seems clear that the construction of a new system or the modification of an existing one should be made based on accurate knowledge of what is to be developed or changed. However, the RE process is often done inappropriately, so that the reqs specification is reduced to a simple, generic mission statement of a few pages many times.

All this despite the fact that many empirical studies in the last years support the claim that reqs management is a critical process in the development of correct software. One of the most common causes of runaway and failed projects are unstable reqs, along with making an inadequate estimation of project time and cost (GLASS, 2002). In most cases, instability of reqs is due to the fact that the customer or users do not really know their needs. Many authors point out errors related to reqs as the most expensive to correct when building software. In this sense, the most difficult problem to address is that reqs relevant to the project are not discovered in time.

System and software reqs documents play a crucial role in software engineering (SE) in that they must both communicate reqs to customers in an understandable manner and define them in precise detail for developers. Archaeological environments can be seen as a complex socio-technical system with many different stakeholders involved (e.g. heritage managers, archaeologists, conservators, developers, analysts). Besides common reqs concerning any software (basic functionality, quality or security) there are specific e-archaeology domain reqs (constraints imposed by e-archaeology standards, expected or desired specific product functionality).

A systematic and rigorous RE approach contributes to obtain interoperable, high-quality e-archaeology systems in a productive way. Using guidelines, recommendations and standards encourages interoperability (common terminology, concepts and procedures) but when these documents are used in a software development context, it is very useful—often necessary—to adapt, refine and express their contents in the form of explicit software and system reqs (see Software Requirements Specifications IEEE Std 830 and System Requirements Specifications IEEE Std 1233).

The main contributions of the proposal described in this paper are: (1) definition of an infrastructure for a reusable reqs repository in the e-archaeology realm, integrating common reqs imposed by SE standards (e.g. Software Engineering Product Quality ISO/IEC 9126 or SQuaRe ISO/IEC 25000), e-archaeology guidelines and recommendations, provided by international consortia active in e-archaeology (i.e. the London Charter[17] and the Principles of Seville[18]) and new product reqs; and (2) using the results above, definition of an RE process specific for e-archaeology systems development.

The repository of reusable reqs catalogues provides a starting point for the subsequent software development, and supports the definition of e-archaeology software product lines (SPL). This approach is based upon previous work of our research group regarding a broad approach to reqs reuse, named SIREN (*SImple REuse of RequiremeNts*) (TOVAL, 2002; 2008).

After this introduction, Section 2 introduces the SIREN method and the use of formal sources of information for reqs identification; Section 3 presents the infrastructure proposed (artefacts and process) and finally, Section 4 summarizes our conclusions and further work.

[17] http://www.londoncharter.org/

[18] http://www.arqueologiavirtual.com/carta/

2 REQUIREMENTS REUSE AND STANDARD GUIDELINES AND RECOMMENDATIONS

In order to take advantage of the benefits of reuse at the reqs level, our group proposed the SIREN method, a practical approach for creating, selecting and specifying the reqs of a software system based on reuse and software engineering standards. SIREN encompasses a spiral process model, reqs documents templates, a reusable reqs repository organized by catalogues, and a supporting reqs management tool (SirenTool). These catalogues are organized in a hierarchy of reqs specification documents, which are structured according to IEEE standards (IEEE Std 830, IEEE Std 1233). The textual information of reqs is complemented by a set of attributes. There is a set of attributes common to all reqs (including priority, rationale, source, state, etc.), and additional attributes can be defined. Besides attributes, different traceability relationships can be defined to relate reqs: to sum up, these are inclusive, exclusive and parent-child traceability relationships. SIREN also deals with variability, e.g. through parameterized reqs, which contain some parts that have to be customized and that have to be instantiated when reused, and generic reqs, used as reqs patterns.

Standard guidelines and recommendations contain technical specifications which ensure that materials, products, processes, services, systems, or persons are fit for their intended purpose, consequence of the experience both in industry and academy. Standards also provide common concepts and practices that encourage interoperability and technology transfer. However, in many cases these sources express the information in a too general form, sometimes introducing problems associated with natural language usage (e.g. ambiguity, imprecision or inconsistencies), sometimes with very different abstraction level information (all in one place or just lack of detail), making the adoption of these standards and their application difficult. Thus, we think it is necessary to adapt, refine and express their contents in the form of explicit reqs. SIREN catalogues and SIREN reqs structure provide the necessary means to achieve this goal as shown in our previous work in the field of e-learning (TOVAL, 2011; COS, 2012).

Currently, the field of e-archaeology is growing very fast. There is a need for establishing solid foundations for this discipline, including both theoretical and practical issues. People involved in computer-based visualisation in the field of cultural heritage are confronted with new technological advances which have to be exploited, whereas barriers and difficulties are to be avoided as well. The London Charter is the most relevant international document dealing with government practices in the field of 3-dimensional visualisation in the research and communication of cultural heritage, and captures basic principles in this growing field. It was proposed by an interdisciplinary panel of researchers belonging to the community of experts in the use of 3-dimensional visualisation technologies in heritage research.

There is a need for achieving greater rigour in the projects within the field of cultural heritage. The London Charter is aimed at filling this gap by means of a set of recommendations and specific guidance. Moreover, the London Charter offers "a robust foundation upon which communities of practice can build detailed London Charter Implementation Guidelines". Given the high amount of valid approaches to the visualisation of cultural heritage and their complexity, specific implementation guidelines for each community of experts should be created.

The International Forum of Virtual Archaeology drafted an international document governing the implementation of best practices in computer-based archaeological visualisation. The Principles of Seville followed the London Charter so as to "increase the conditions of applicability of the London Charter in order to improve its implementation specifically in the field of archaeological heritage". Like its predecessor, this document provides recommendations and guidance, but it is focused on the specific needs of archaeological heritage in relation to cultural heritage. Since the theoretical framework for the Seville Principles is the London Charter, this document adopts the objectives approved by the Advisory Board of the London Charter.

Both documents, the London Charter and the Principles of Seville, can be used as reqs sources for building SIREN catalogues as explained in the following section.

3 Basis for an e-archaeology systems SPL infrastructure

The software factory (SF) concept (GREENFIELD & SHORT, 2004) refers to those software development organizations which can group together an important number of their products in the so called software product lines (SPL) (KÄKÖLÄ & DUEÑAS, 2006). Typically, SFs are able to systematize their production systems, mainly based upon reuse of a variety of artefacts (such as reqs, analysis or design models, code, and documentation). The current trend is towards SPLs in particular domains (such as banking or automotive fields). An SPL is defined by a set of systems sharing common features, which satisfy the specific needs of a particular sector. These systems are developed within the SPL from a pre-established set of assets or reusable artefacts. SPL development consists usually of two complementary processes: *domain* and *product engineering*. Reqs in a SPL define the common and variable features (domain level) as well as concrete products in such line (product level). Therefore, there is a process for building reusable catalogues (Domain Engineering) and another one for using them (Product Engineering). Although we define these processes sequentially, these are performed in an iterative and incremental way.

The Domain Engineering process

The activities in this group are carried out at the beginning of defining a new SPL and each time we wish to generate new catalogues within this SPL. Considering an SPL for computer-based visualisation of archaeological heritage, then a set of generic reqs catalogues of computer-based visualisation tools within the e-archaeology domain can be created:

a) SPL Definition: A list of textual descriptions regarding high level issues which should be solved by means of the products of the SPL. Business processes potentially involved are also identified and textually described in a few paragraphs, to better describe a context.

b) Problem Domain scope: In short, it consists of:
b.1) Identifying main sources related to the domain to which the SPL belong: e-archaeology recommendations and guidelines, SE standards, legislation, problem-specific documents, etc. For example, the London Charter, the Principles of Seville and the ISO/IEC 25000 can be selected for the construction of a computer-based visualisation tool. Then, select, prioritize and schedule the generation of the detailed catalogues.
b.2) Generating first version catalogues, for each one of the sources selected in the previous step. Typically, these will consist of a mapping from relevant text in the sources to software reqs, where common and variable features in the domain are described. Examples of reqs at this level are:

S1. Digital preservation strategies should aim to preserve the computer-based visualisation data, rather than the medium on which they were originally stored, and also information sufficient to enable their use in the future, for example through migration to different formats or software emulation (Section 5.2, Principle 5: "Sustainability" from the London Charter).

S2. The computer-based visualisation tool shall allow virtual recreation of archaeological heritage (Problem-specific domain document).

c) Solution Domain scope: Generating detailed reusable reqs catalogues for each one of the sources selected. These catalogues will contain and refine the part of the reqs at the previous level, corresponding exactly to the kind of products provided by this SPL. Common and variable features, not just in the domain but in the particular SPL, are now described and detailed. Specific techniques, algorithms or procedures (even keeping variability) for implementing particular products in the SPL are now considered. Examples of refined reqs at this level are:

C1. The computer-based visualisation tool shall allow to preserve the computer-based visualisation data (Refined from req *S1*; included in the London Charter catalogue).

C2. The computer-based visualisation tool shall store all the raw data and metadata to enable the migration of the computer-based visualisation data to [non empty set of file formats] (Refined from req *S1*, with parameter *[non empty set of file formats]*; inclusive traceability relation with the req *C1*, that is, the reuse of *C2* will imply the reuse of *C1*; included in the London Charter catalogue).

C3. The computer-based visualisation tool shall allow to use a virtual model to visually recover an archaeological site at a given moment in the past, including [non empty set of archaeological heritage] (Refined from req *S2*, with parameter *[non empty set of archaeological heritage]*; included in the Computer-Based Visualisation Tool (CBVT) catalogue).

At this point, reqs attributes such as identifier (e.g. *C1*), priority (e.g. high), source (e.g. Section 5.2, the London Charter) and person in charge (e.g. John) are incorporated. Both the reqs attributes and traceability relationships can also be reused.

The Product Engineering process

The activities in this group are carried out each time we wish to develop a new, specific e-archaeology product or evolve an existing one. For example, let us suppose that we intend to build a computer-based visualisation tool, addressed to landscape archaeology. The computer-based visualisation systems allow heritage experts to study landscape and to explore rich archaeological contexts. These systems collect data that can be useful in forming theory and rules of practice. Geophysical prospection is a good example of this, taking advantage of software that provides a host of data processing, georeferencing, and data display algorithms. As a result, landscape archaeology can use this technology to model a given landscape and move from data acquisition to content generation (CH'NG, 2011).

a) Product Definition: Decide the main required features of the product (such as data acquisition, software quality, etc.), selecting them from a pre-defined form and providing weights within a homogeneous scale. This form identifies high-level abstraction functional and non-functional reqs, both archaeological and general purpose. For example, Sustainability: *HLR1. Where digital archiving is not the most reliable means of ensuring the long-term survival of a computer-based visualisation outcome, a partial, two-dimensional record of a computer-based visualisation output, evoking as far as possible the scope and properties of the original output, should be preferred to the absence of a record*, and Software quality: *HLR2. Check that each test requirement is linked to a test case.* A first specification, the so called "Initial product specification", including a prioritized list of main required features, is generated.

b) Select Sources and Catalogues: Select the available catalogues related to features in the "Initial product specification". For example, the available London Charter catalogue might be reused to describe how to store the visualisation outcome mentioned in HLR1. Then, for those features not considered in our existing repository of catalogues, identify related and applicable sources of interest. For example, when the complexity of the datasets captured is very high, the feature "*The computer-based visualisation tool shall implement data mining techniques to analyse the visualisation data*" is not in the SPL catalogues. In this case, information sources related with the data mining techniques should be selected. At this point, and according to the existing budget, we can decide either it is worthy to build new catalogues corresponding to these features/sources (e.g. if they can be reused in further projects) or just to use the sources directly to obtain new reqs for our product requirements specification (PRS) (part of the so called *deltas*).

c) Instantiate/Reuse reqs: By using the selected, available or newly built catalogues, obtain a first version of the PRS populated with reused reqs. Examples of product reqs at this level are:

R1. The computer-based visualisation tool shall allow preserving the computer-based visualisation data (Used as it was in previous step).

R2. The computer-based visualisation tool shall store all the raw data and metadata to enable the migration of the computer-based visualisation data to SVG (From parameterized req C2, with alternatives SMIL, SVG and X3D; the trace from R1 would also be instantiated).

R3. The computer-based visualisation tool shall allow to use a virtual model to visually recover an archaeological site at a given moment in the past, including landscape (From parameterized req C3, with alternatives material culture, environment, landscape, customs, and general cultural significance).

d) Elicitate reqs: Add to the PRS new, specific, reqs for this product (*deltas*).

e) Analyse and Negotiate: Check possible inconsistencies and problems coming from the integration of reused and newly elicitated (*deltas*) reqs. Solve possible different views and interests of participating stakeholders.

f) Validate & Verify reqs: To ensure the quality of the PRS created, check it to guarantee both that the resulting product will perform as expected in the user's environment, and whole compliance with regard to the RE processes and standards used for the PRS (e.g. IEEE Std 830).

Activities *d*, *e*, and *f* are typical of any RE process, while the others help to configure a specific e-archaeology SPL. Note that these steps are applied iteratively and incrementally until an approved PRS is achieved (the so called *baseline*).

4 CONCLUSIONS AND FUTURE WORK

This approach aims at helping in the production of e-archaeology systems according to SE and RE best practices. Independently of any particular process model, our proposal can be integrated with common practices in organizations devoted to software development in the cultural heritage realm, just by adapting the structure of the reqs documents. Catalogues provide quality and a well-organized set of reqs, improving the related information from the original sources.

The identification and selection of suitable sources of reqs for e-archaeology systems development is quite difficult, given that few standardisation initiatives have been taken so far. There is a need for further and more technically detailed guidance and regulation in computer-based visualisation in e-archaeology and related domains, since these efforts will provide the foundations for improving product quality and process productivity. The expansion and refinement of the guidelines included in the London Charter and the Seville Principles should be gradually addressed. Then, reqs catalogues can be created starting from the top priority and most used or more important sources (e.g. those that are mandatory, by law). The more catalogues we have in our repository, the better rates we will obtain both in productivity (reusing and instantiating reqs is faster than defining them from the scratch), quality (reused reqs are improved, potentially in each iteration) and interoperability (catalogues based upon standards).

In the near future, we plan to define a taxonomy of cultural heritage systems, with the purpose of: (1) making easier the identification of common features/reqs for the different groups found; and (2) helping in deciding the direction to be taken with regard to the definition of the corresponding implementation guidelines and standards.

ACKNOWLEDGMENTS

This research is part of the project PEGASO-PANGEA (TIN2009-13718-C02-02), financed by the Spanish Ministry of Science and Innovation (Spain).

REFERENCES

CH'NG, E. et al. (2011): "From sites to landscapes: how computing technology is shaping archaeological practice", en *Computer*, vol. 44, n. 7, pp. 40-46.

COS, J.A. et al. (2012): "Internationalization requirements for e-learning audit purposes", en *Proceedings of the 3rd IEEE Global Engineering Education Conference*, EDUCON 2012, pp. 90-95.

GLASS, R.L. (2002): *Software engineering: facts and fallacies*. Addison-Wesley. Boston.

GREENFIELD, J. and SHORT, K. (2004): *Software factories: assembling applications with patterns, models, frameworks, and tools*. Wiley. Indianapolis.

KÄKÖLÄ, T. and DUEÑAS, J.C. (Eds.) (2006): *Software Product Lines. Research issues in engineering and management*. Springer. Berlin Heidelberg.

TOVAL, A. et al. (2002): "Requirements reuse for improving information systems security: a practitioner's approach", en *REJ Requirements Engineering Journal*, vol. 6, n. 4, pp. 205-219.

TOVAL, A. et al. (2008): "Eight key issues for an effective reuse-based requirements process", en *IJCSSE International Journal of Computer Systems Science and Engineering*, vol. 23, n. 6, pp. 373-385.

TOVAL, A. et al. (2011): "Learning systems development using reusable standard-based requirements catalogs", en *Proceedings of the 2nd IEEE Global Engineering Education Conference*, EDUCON 2011, pp. 907-912.