ElicitO: A Quality Ontology-Guided NFR Elicitation Tool

Taiseera Hazeem Al Balushi, Pedro R. Falcone Sampaio, Divyesh Dabhi, and Pericles Loucopoulos

School of Informatics, University of Manchester, PO Box 88, Manchester M60 1QD, UK
Taiseera.Al-balushi@postgrad.manchester.ac.uk
{P.Sampaio,P.Loucopoulos}@manchester.ac.uk
D.Dabhi@student.manchester.ac.uk

Abstract. Despite the importance of capturing a precise and complete set of requirements in the requirements engineering stage, there are few tools that adequately support requirements analysis in the process of capturing quality related requirements (non-functional requirements). This paper presents ElicitO, a requirements elicitation tool aimed at empowering requirements analysts with a knowledge repository that helps in the process of capturing precise non-functional requirements (NFRs) specifications during elicitation interviews. The approach is based on the application of functional and non-functional domain ontologies (quality ontologies) to underpin the elicitation activities. The tool is used as a memory aid to structure elicitation interviews, guide requirements analysts with regard to the important quality aspects relating to a class of applications, and support the development of precise requirements based on characteristics and metrics available in quality model standards.

Keywords: non-functional requirements (NFRs), requirements engineering, requirements elicitation, ontologies, tools, Protégé.

1 Introduction

Requirements elicitation is often regarded as the most critical stage of the entire requirements engineering effort [1]. An adequate set of requirements, as defined by [2], should enable users to have a comprehensive view of their system related needs and a proper understanding of the constraints that will affect the quality of their experience in using the system. On the other hand, the set of requirements should also enable developers to obtain a precise and complete description of the functional and non-functional aspects of the system. The IEEE Guide to Software Requirements Specifications [3] defines a proper requirements specification as being: unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during operations and maintenance. To help achieving this, the requirements elicitation process should consider: (1) the functional requirements which are associated with specific functions, tasks, or behavior that the system must support and (2) the non-functional requirements (NFR) or quality requirements that represent constraints on functional requirements. NFRs are often regarded as the key success factor in building high

P. Sawyer, B. Paech, and P. Heymans (Eds.): REFSQ 2007, LNCS 4542, pp. 306–319, 2007.

quality software [4], [5] enabling a systematic and pragmatic approach of building quality into software systems [6]. Current elicitation approaches and tools such as JAD [7], Domain Analysis [8], CORE [9] and Scenario based elicitation [10], [11], [12] have focused on the identification, specification and management of functional requirements, however, only a handful of tools addresses the issue of adequately supporting non-functional requirements elicitation. The key challenges linked to supporting NFRs elicitation are:

1. *The depth/breadth of the scope of the qualities or NFRs involved in a particular domain:* Requirements analysts usually lack a deep understanding of relevant quality requirements of an application domain, therefore needing additional knowledge support in the process of asking the right question to elicit requirements [13].
2. *The precision of the NFRs elicited:* Quality requirements are usually stated informally (e.g., the system should be fast or the user interface should not be cluttered) and few approaches define a quality model and/or attach metrics to non-functional requirements (qualitative or quantitative measures of the requirements).
3. *Tool and process support:* Elicitation of NFRs is still treated as a pencil and paper exercise with little support for processes and tools aimed at requirements identification, validation and management.

This paper presents ElicitO, a requirements elicitation tool aimed at empowering requirements analysts with a knowledge repository that helps in the process of capturing precise non-functional requirements specifications during elicitation interviews. The approach is based on the application of functional and non-functional domain ontologies (quality ontologies) to underpin the elicitation activities. The tool is used as a memory aid to structure elicitation interviews, guide requirements analysts with regard to the important quality aspects relating to a class of applications, and support the development of precise requirements based on characteristics and metrics available in quality model standards.

The remainder of this paper is structured as follows: Section 2 gives a background on ontologies and their application. Section 3 describes the quality ontology underpinning ElicitO. Section 4 describes the ElicitO tool architecture. Section 5 presents a small case study to evaluate ElicitO. Section 6 presents some related work on tools for supporting elicitation activities followed by a discussion in section 7. Section 8, summarizes the paper, key contributions, and the future work.

2 Background

An ontology is an explicit specification of a shared conceptualization [14]. Ontologies provide a vocabulary for structuring a knowledge domain and for describing specific situations in a domain [15], fostering a common understanding of the structure of information among people or software agents [16] (e.g. GeneOntology [16], WordNet[17]). Ontologies also support the reuse of domain knowledge (e.g. Enterprise Ontology [18]), helping to make domain assumptions explicit. Ontologies

can also be applied to support requirements engineering activities providing the following benefits:

1. Promote a shared domain vocabulary that can be used to avoid ambiguities arising in projects involving teams of multiple requirements engineers and stakeholders.
2. The representation and reasoning capabilities enable the description of quality constraints associated with the functional domain.
3. Ontologies are often used to encode specialized knowledge to support the formulation of competency questions with regards to the quality requirements relevant to a particular domain, facilitating the elicitation of a complete set of quality requirements during stakeholders' interviews.

Recently, quality ontologies are being used to capture quality properties of helpdesk and customer relationship management systems [19] and others such as in the TOVE quality Ontology[20], Bioinformatics applications such as in the Qurator project [21], and quality of service requirements for service centric systems [22].

3 The Quality Ontology Underpinning ElicitO

ElicitO is based on the use of functional and non-functional ontologies to develop an ontology driven requirements elicitation method, guided by a standard quality model. The quality model is encoded in the quality ontology, and automated by a requirements elicitation tool. Fig. 1 illustrates the ElicitO framework. To develop the functional and non-functional ontologies underpinning ElicitO, the ontology development process proposed by Falbo and Menezes [23] was followed to identify the goal of the ontology, structure the ontology, and formalize/implement the ontology and its describes in more details as follows:

Identify the goal of the ontology
The ontology's main objective is to help in promoting a shared understanding about a functional domain as well as the relevant quality aspects of the domain. The ontology encodes knowledge relating to the characteristics and metrics available in a standard quality model (ISO/IEC 9126) [24], enabling the development of assertions stating the quality properties of a functional element of the domain.

Structuring the ontology
The ElicitO approach/tool is based on two ontologies; the quality ontology and the functional domain ontology. In this paper we provide an example ontology relating to the functional domain of a university helpdesk. To structure the university domain knowledge and its quality characteristics we used textbooks, quality and industry standards, and interviews with domain experts (e.g. head of information services and five help desk operators with more than 5 years of experience each) see Fig. 2. The OMG's Software Process Metamodel (SPEM)[25] was used to represent the ontology development process.

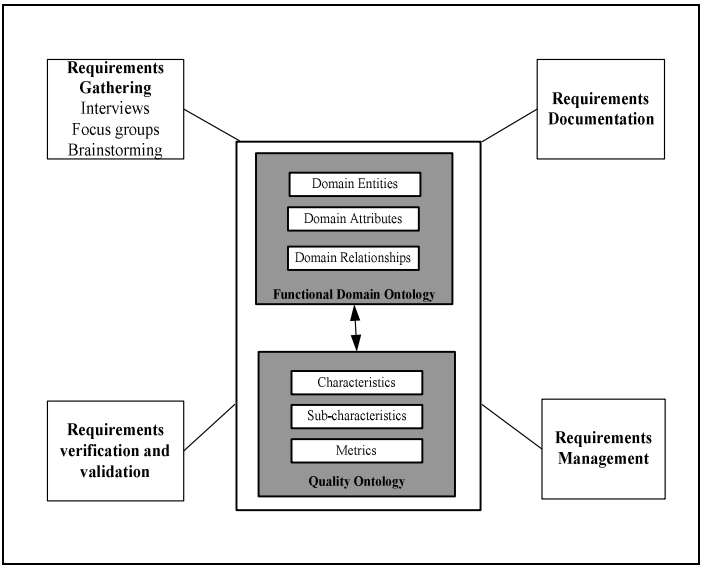

Fig. 1. The ElicitO Requirements Elicitation Framework

The ISO/IEC 9126 quality model codified in the quality ontology supports the representation of reusable knowledge about different quality characteristics, sub-characteristics, and metrics. These quality factors are general and can be applied to any application domain, however, the level of quality required and the order of importance of these quality factors may vary from a domain to another and will be further detailed during elicitation interviews.

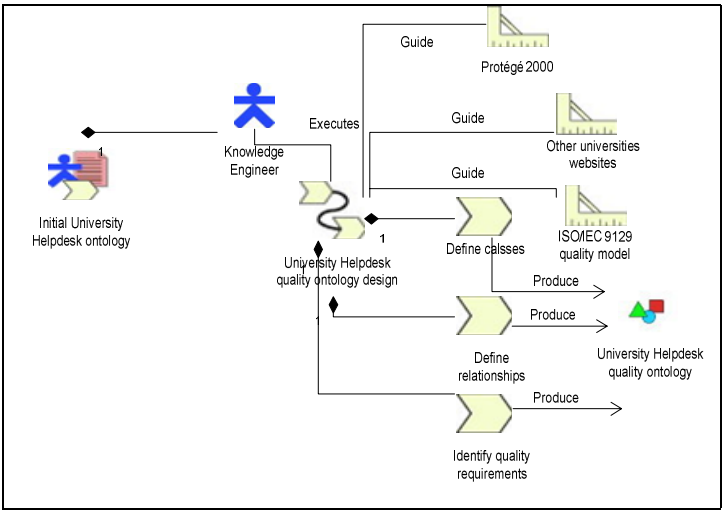

Fig. 2. Initial University Ontology Development Process

Ontology formalization/implementation

After all the elements of the domain ontologies and the quality ontologies are identified, they are implemented using Protégé-2000 [26], a comprehensive tool for developing knowledge-based systems. The advantages of using Protégé to support elicitation activities stem from:

1. An extensible knowledge model supporting the declarative representation and reuse of requirements specifications.
2. A reasoning framework supporting the development of quality requirements as constraints relating to the functional requirements.
3. Robust and well-documented tool supporting a customizable output file format and user interface.

The implementation of the ElicitO ontologies in Protégé-2000 is illustrated in Fig 3. The left-hand side of Fig 3 (A) shows how quality characteristics, sub-characteristics and metrics of the quality model are mapped into a hierarchy. Further details of the quality model implemented in ElicitO can be found in [19]. Fig 3 (B) shows part of the functional domain ontology. The right-hand side of the figure defines each class, relationships, and asserted conditions Fig 3 (C). One of the important features of the Protégé is the built in reasoning capabilities allowing the development of constraints on how the ontology should be used. This is achieved through OWL expressions denoting domain restrictions/constraints [27], [28].

In developing ElicitO, there are two main sets of asserted conditions:

1. Metrics identification; on which all related metrics to a certain application domain are identified:
   - has_a _QualityMetric ∋ Num_of _links_ per page
   - has_a_QualityMetrc ∋ Max_num_of_links_in_an_index_page
   - has_a_QualityMetric ∋ Avg_num_of_words_per_page
   - has_a_QualityMetric ∋ num_of_images_per_page
   - has_a_QualityMetric ∋ page_download_speed
   - has_a_QualityMetric ∋ Avg_num_of_colours_per_page

   Later these metrics are defined as to which quality characteristic and sub-characteristic they represent. This is to ensure ElicitO tool's compliance to ISO/IEC 9126 standard. The example below shows how the metric (Num_of _links_ per page) is represented:
   - ∃ has_a_Quality Characteristic Usability
   - ∃ has_a_Quality_SubCharacteristic Understandability
   - ∃ measured_by Number
2. Relate the above defined quality metric to a certain domain function, The example below shows the most important quality metric to the activity (FAQ):
   - has_a_QualityMetric ∋ page_download_speed
   - has_a_QualityMetric ∋ Avg_num_of_colours_per_page
   - has_a _QualityMetric ∋ Num_of _links_ per page

These metrics are defined once and they are reusable across any other functional domain activity (e.g. email support, library support, etc). The knowledge codified is reusable across elicitation sessions enabling requirements analysts to configure a new set of requirements for a specific systems development scenario.

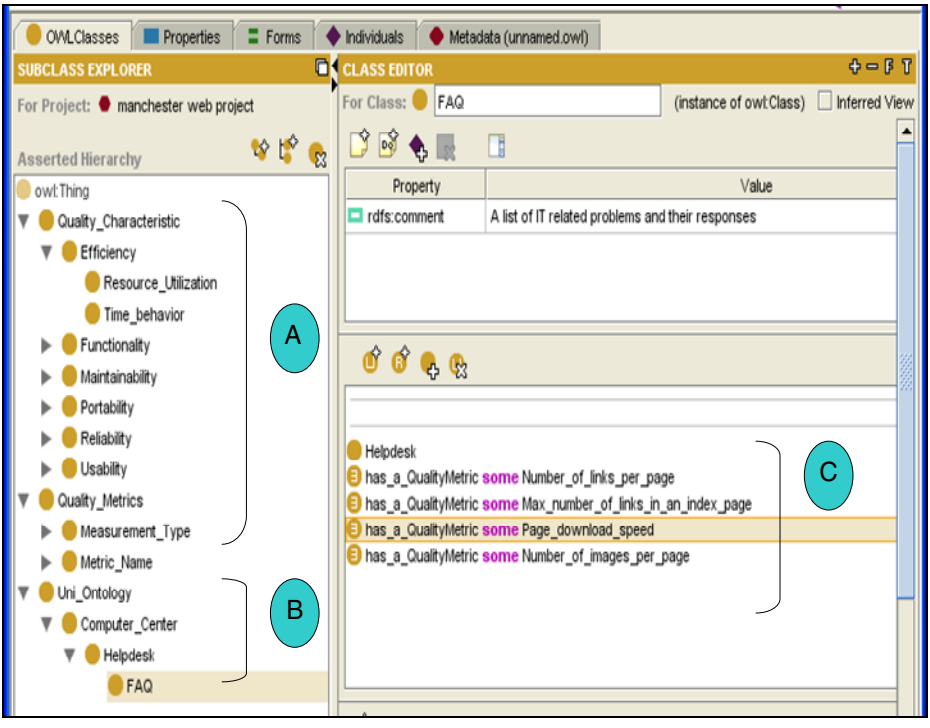

Fig. 3. Quality Ontology in Protégé; (A) quality ontology, (B) functional domain ontology, (C) rule/restriction

4 ElicitO Tool Goals and Architecture

Some of the goals of the ElicitO tool are:

1. Help to automate the time consuming process of identifying NFRs relevant to a certain domain by having all relevant knowledge encapsulated in the ontology.
2. Help the requirements analysts in the process of requirements elicitation disregarding his/her level of expertise in obtaining a rapid understanding of all relevant functional and non-functional requirements of a given domain.
3. Decrease the occurrence of problems of understanding between stakeholders (e.g., enabling that all NFRs are uniformly treated across different elicitation interviews conducted by different requirements analysts), thus, reducing the chances of missing out important requirements or not treating requirements uniformly.
4. Capture the quality requirements for any other functional domain provided that the specifications are made when the functional domain ontology is constructed. This supports the reusability of the quality ontology.

The architecture of ElicitO is displayed in Fig. 4. The bottom layer is the ontology layer where both the functional domain ontology and quality ontology are stored in

Protégé database. The application layer communicates with the ontology layer when querying for domain knowledge and the related quality attributes via the Protégé API. All query results and information that is displayed to the user is done via the graphical user interface layer.

The implementation language used to build the application-layer and user-interface layer was Java, as the Protégé environment is itself implemented in Java. The underlying database for the storage of the requirement sessions was chosen to be MySQL. The tool gives to users two options to store requirements elicitation sessions: as a text file for importing in a word processing package; or as a proper relational database in the MySQL database. The NetBeans Integrated Development Environment (IDE) 5.0 platform was used and all the user-interfaces were built within.

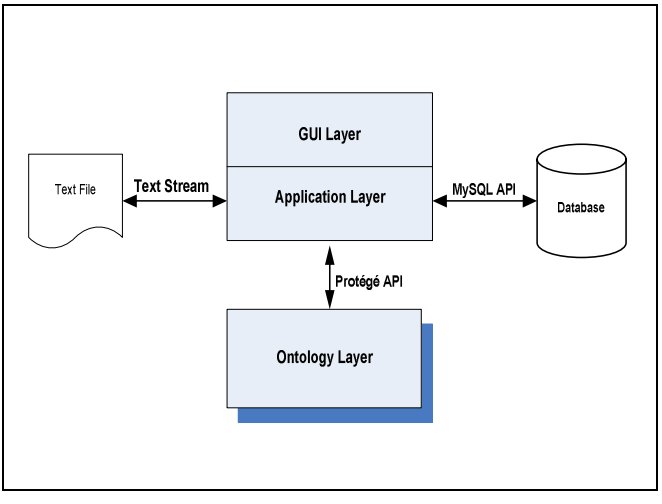

Fig. 4. ElicitO Tool Architecture

5 Using ElicitO in a Web Development Project

The ElicitO tool is currently being used to support requirements engineering activities in connection with the Manchester Unity Web Project. The objective of the project is to enhance the current website of the university by adding extra features specified by different stakeholders' views. To evaluate ElicitO, the authors attended a focus group session which was one of the ongoing sessions aimed at enhancing the current helpdesk website of the university. In a two hours focus group session the participants were asked for what they want to have available on the website and what the problems they come across using the website. Table 1 presents requirements elicited from the focus group session. The amount of requirements collected was limited and some can be regarded as very general and not clear enough. Requirements

are also unstructured with a mix of functional and non-functional requirements across the document.

In contrast to the unstructured and ad-hoc pencil and paper exercise conducted during focus group sessions, the interview process of requirements elicitation using ElicitO benefits from the tool guidance with regard to the relevant quality characteristics, sub-characteristics, and metrics relating to a functional element of the domain that will underpin the formulation of a precise requirement statement. Fig. 5(a) shows the user interface of the tool from which the requirements analysts and the stakeholders interact. Once a certain activity is selected in the tool (e.g. FAQ) relevant quality characteristics that can be discussed with stakeholders towards developing NFR specifications are presented.

Table 1. Requirements captured without tool support

	User Requirements
R1	Provide information/pathway onto how to access web services (i.e. web mail, network drive, etc.)
R2	FAQ should be clear and simple in answering users technical problems
R3	Make the websites among different schools consistent
R4	Provide campus map when required
R5	Make the university regulations and policies easy to access
R6	Make students user names accessible to faculty when using WebCT (e-learning) to register students
R7	Provide information on how to report a problem and to whom
R8	Provide information about exam timetables and venues
R9	Provide links to the outside world
R10	Highlight important events or alerts
R11	Update the staff directory frequently

The add requirements button allows the stakeholders to detail a quality requirement, in the given example, the quality characteristics (efficiency) and their associated sub-characteristics (time behavior) related to the functional activity FAQ. The tool also allows the requirements analyst to ask more specific questions about their quality requirements through metrics such as (page download speed) and the stakeholders specified (15 seconds), see Fig. 5(b). Fig. 5(c) presents an example of requirements obtained using ElicitO tool and after interviewing two of the participants (Intranet project manager and the IT services manager) from the focus group (same amount of time used during the focus group).

The NFRs captured in Fig. 5(c) using the ElicitO tool has an enhanced level of precision and scope when compared to the general requirements elicited in focus group sessions. The metrics in the ontology help in promoting a precise metrification of the relevant quality aspects. This is due to the fact that the tool leverages the knowledge repository of functional and non-functional requirements relevant to the domain

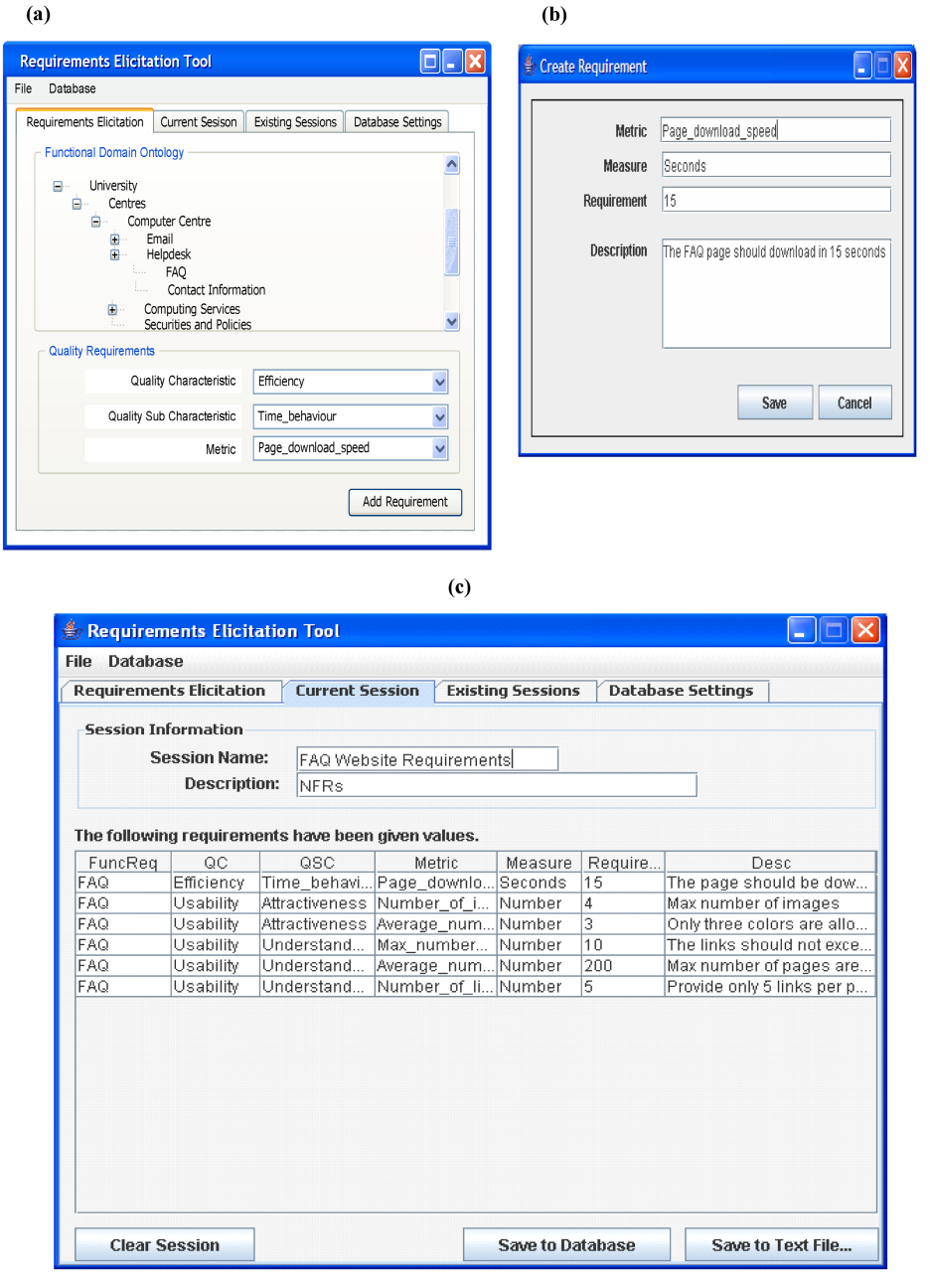

Fig. 5. (a): ElicitO GUI; (b): eliciting specific requirements; (c): requirements document

of discourse. The knowledge encoded in the ontology has a positive impact in reducing the problem of scope (helping requirements analysts to focus on the relevant aspects of the domain) and reducing the chances of missing out important aspects of quality requirements. The tool also helps to promote effective communications as the quality/functional requirements are better communicated with the stakeholders as they are defined and broken down into a set of measurable metrics.

6 Related Work on Tools for Supporting Requirements Elicitation

Table 2 compares ElicitO with some other commercial and academic tools available with respect to the focus of the tool, domain knowledge support, quality model support, and metrics support. Focus of the tool, evaluates the tools on the basis of their coverage to requirements engineering activities. Domain knowledge support, evaluates the tools based on the availability of knowledge base in guiding the requirements engineering activities. Quality model support, evaluates the tools with respect to their explicit support for a quality model to help in providing a comprehensive and standardized set of NFRs to be integrated in the elicitation process. Metrics support, which helps in proving precise set of NFRs.

An important point to highlight is that ElicitO is unique in providing knowledge support to the elicitation process based on ontologies, and jointly with QFD, provides support for a standard quality model with well-defined quality metrics. Another important point to emphasize is that whilst other tools have emphasized requirements management, traceability, and prioritization, ElicitO focuses on support for non-functional requirements elicitation and requirements reuse across different application domains.

7 Discussion

There are four key aspects relating to the elicitation approach supported by the ElicitO tool:

- *Emphasis on requirements reusability*, the reusability notion is often explored in connection with code reuse, design reuse and object oriented development approaches [29]. However, reuse can also be applied in connection with the requirements phase where product quality requirements and general domain features and functions are made explicit to be used in different projects. ElicitO advocates this feature via the use of ontologies which support the reusability of knowledge (requirements in our case). This helps in making the requirements elicitation phase more effective as the domain assumptions are made explicit to stakeholders and NFRs can be tailored depending on the needs of each individual scenario.

Table 2. Comparison of requirements elicitation tools

Tools	Focus of the tool	Domain knowledge Support	Quality Model Support	Metrics Knowledge Support
ElicitO	**FR and NFRs elicitation & reusability**	**Application Domains encoded as ontologies**	**ISO/IEC 9126**	√
CaliberRM 2005[1]	Enterprise Requirements management (tractability & collaboration)	Repository of existing project requirements		
DOORS[2]	Requirements management. Requirements modeling for understandability and reusability	Template of requirements documents without specific domain knowledge		
IBM Rational Requisite Pro 03/06[3]	Groupware for Requirements management (traceability & impact analysis)	Reuse requirements from existing projects		
QFD /Capture V.4[4]	Requirements identification & prioritization		QFD	√
NFR Assistant [6]	NFR identification and conflict resolution		List of NFRs without relating to quality model	
QM tool [30]	Define a quality model for an application	Business application software features	ISO/9126	√

- *Use of quality models to capture precise quality metrics*; ElicitO is based on the quality model ISO/IEC 9126 which encompasses a comprehensive set of product quality characteristics. ISO/IEC 9126 has also been applied to many software engineering projects/applications [4], [30], [31], [32], [33], [34]. Quality models help in highlighting which quality attributes are important, their level of importance, and their measurement methods. Adopting a quality model also helps project managers with software product evaluation and risk identification [35]. For example, there is significant research on website quality models [36], [37], [38] and also research that emphasizes a single quality dimension such as usability [39] and security [40].

[1] http://www.borland.com/us/products/caliber/index.html/
[2] http://www.telelogic.com/corp/products/doors/
[3] http://www-306.ibm.com/software/awdtools/reqpro/
[4] http://www.qfdcapture.com/

- *Emphasis on product/service quality*; it is important to note that the authors use NFRs and quality attributes interchangeably, this is because NFRs are often viewed as systems properties or constraints [41] which are key elements to assess the effectiveness of functional capabilities of a system, (e.g.; all call centers need to handle calls and deal with customer's requests (functional requirements). However, factors such as how long the customer waits until he/she gets to speak to an agent and/or how many calls can a call center handle at a time are key factors representing the quality of call centers activities. Hence, NFRs help to express the effectiveness of the functional capabilities of a system (product quality).
- *Focused at enhancing productivity*; stored ElicitO ontologies help requirements engineers to speed up requirements capture by navigating and completing NFRs forms. The tool also helps in standardizing requirements across teams of engineers.

8 Conclusions and Future Work

This paper presents ElicitO, a requirements elicitation tool providing automated support for non-functional requirements elicitation. The tool applies functional and non-functional domain ontologies to support requirements analysts with domain knowledge to develop a comprehensive and precise set of requirements during elicitation interviews. The paper discusses the elicitation approach supported by the tool, the ontologies underpinning the tool, the tool architecture and the paper also provides an example of how ElicitO is being used to support the development of NFRs for a web engineering project at the University of Manchester. Future work will be focused in developing requirements specifications across different domains to assess the reusability of the quality ontology. We are also using the reasoning capabilities supported by the knowledge management environment (Protégé) to develop validation checks for captured requirements, enabling consistency checking across requirements developed by teams of requirements analysts.

References

1. Brooks, F.: No sliver bullet-Essence and accidents of software engineering. Computer 20(4), 10–19 (1987)
2. Saiedian, H., Dale, R.: Requirements engineering: making the connection between the software developer and customer. Information and Software Technology 42(6), 419–428 (2000)
3. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
4. Azuma, M.: Applying ISO/IEC 9126-1 Quality Model to Quality Requirements Engineering on Critical Software. In: Proceedings of the 3rd International Workshop on Requirements Engineering for High Assurance Systems. Kyoto, Japan (2004)
5. Bordewisch, R., et al.: Non-Functional Aspects: Systems Performance Evaluation. In: THOME, B. (ed.) Systems Engineering: Principles and Practice of Computer-Based Systems Engineering, pp. 223–271. John Wiley & Sons Ltd, Chichester, UK (1993)

6. Chung, L., et al.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishing, Norwell, Massachusetts. 472 (2000)
7. Wood, J., Silver, D.: Joint Application Development. Wiley, New York (1995)
8. Kang, K., et al.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Software Engineering Institute. Technical Report CMU/SEI-90-TR-021, Pittsburgh, PA (1990)
9. Mullery, G.P.: CORE: A method for controlled requirements specification. In: Fourth International Conference on Software Engineering (1979)
10. Holbrook, H.I.: Scenario-based methodology for conducting requirements elicitation. ACM SIGSOFT Software Engineering Notes 15(1), 95–104 (1990)
11. Maiden, N.A.M.: CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements. Automated Software Engineering 5(4), 419–446 (1998)
12. Sutcliffe, A., et al.: Supporting scenario-based requirements engineering. IEEE Transactions on Software Engineering 24(12), 1072–1088 (1998)
13. Kassel, N.W., Malloy, B.A.: An Approach to Automate Requirements Elicitation and Specification. In: Proceedings of the 7th IASTED International Conference on Software Engineering and Applications. Marina Del Rey, CA, USA (2003)
14. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing, in Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers, Deventer, The Netherlands (1993)
15. Fikes, R., Farquhar, A.: Distributed repositories of highly expressive reusable ontologies. IEEE Intelligent Systems 14(2), 73–79 (1999)
16. GeneOntology, http://www.geneontology.org
17. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
18. Uschold, M., et al.: The Enterprise Ontology. AIAI, The University of Edinburgh (1997)
19. AlBalushi, T., et al.: Performing Requirements Elicitation Activities Supported by Quality Ontologies. In: Eighteenth International Conference on Software Engineering and Knowledge Engineering. San Francisco, Knowledge Systems Institute Graduate School (2006)
20. Kim, H.M., Fox, M.S., Gruninger, M.: An ontology of quality for enterprise modeling. In: Proceedings of the Fourth Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1995 (1995)
21. Qurator (2005) http://www.qurator.org
22. Dobson, G., Lock, R., Sommerville, I.: Quality of Service Requirements Specification using an Ontology. In: Proc. Workshop on Service-Oriented Computing Requirements (SOCCER) Paris (2005)
23. Falbo, R.A., Menezes, C.S., Rocha, A.R.C.: A Systematic Approach for Building Ontologies. In: Coelho, H. (ed.) IBERAMIA 1998. LNCS (LNAI), vol. 1484, Springer, Heidelberg (1998)
24. ISO/IEC 9126-1:2001 Software engineering –Product quality – Part 1: Quality model.
25. Object Management Group. Software Process Engineering Metamodel Specification. Version 1.1. January 2005, Technical Report 05-01-06,OMG (2005)
26. Protege (2000) The Protege Project, http://protege.stanford.edu
27. McGuinness, D.L., Harmelen, F.v.: OWL Web Ontology Language Overview. W3C Recommendation (2004)
28. Knublauch, H., et al.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In: Third International Semantic Web Conference. Hiroshima, Japan (2004)
29. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesely Professional (1999)

30. Carvallo, J.P., et al.: QM: A Tool for Building Software Quality Models. In: Proceedings of the 12th IEEE International Conference of Requirements Engineering 2004, IEEE Computer Society, Kyoto, Japan (2004)
31. Bhatti, S.: Why Quality?ISO 9126 Software Quality Metrics (Functionality) Support by UML Suite. ACM SIGSOFT Software Engineering Notes 30(2), 1–5 (2005)
32. Cote, M., et al.: The Evolution Path for Industrial Software Quality Evaluation Methods Applying ISO/IEC 9126:2001 Quality Model: Example of MITRE's SQAE Method. Software Quality Journal 31(1), 17–30 (2005)
33. Firesmith, D.: Using Quality Models to Engineer Quality Requirements. Journal of Object Technology 2(5), 67–75 (2003)
34. Doerr, J., et al.: Non-functional requirements in industry - three case studies adopting an experience-based NFR method. In: Proceedings of the 13th IEEE International Conference on Requirements Engineering (2005)
35. Hayatt, L., Rosenberg, L.: A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In: 8th Annual Software Technology Conference. Utah (1996)
36. Cox, J., Dale, B.G.: Service quality and e-commerce: An exploratory analysis. Managing Service Quality 11(2), 121–131 (2001)
37. Webb, H.W., Webb, L.A.: SiteQual: an integrated measure of Web site quality. Journal of Enterprise Information Management 17(6), 430–440 (2004)
38. Olsina, L., Rossi, G.: Measuring Web application quality with WebQEM. Multimedia, IEEE, vol. 9(4), pp. 20–29 (2002)
39. Nielsen, J.: Designing Web Usability: the practice of simplicity. New Riders Publishing (1999)
40. Firesmith, D.: Engineering Security Requirements. Journal of Object Technology 2(1), 53–68 (2003)
41. Sommerville, I.: Software Engineering. 7th edn. Essex, England, Pearson Education Limited (2004)