Computer Standards & Interfaces 47 (2016) 62–76

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

CDNFRE: Conflict detector in non-functional requirement evolution based on ontologies

Chi-Lun Liu

Department of Multimedia and Mobile Commerce, Kainan University, No. 1, Kainan Road, Luzhu Shiang, Taoyuan County, Taiwan

ARTICLE INFO

Article history:
Received 20 September 2015
Received in revised form 12 February 2016
Accepted 15 March 2016
Available online 17 March 2016

Keywords:
Systems analysis
Requirement engineering
Ontology
Conflict detection
Non-functional requirement

ABSTRACT

Analyzing conflicts in non-functional requirements is a major task in large software systems development projects. Many of the non-functional requirements that accumulate vary over time. Systems analysts often maintain non-functional requirements incrementally. Requirement information overload and employee turnover problems may complicate conflict detection in the non-functional requirement evolution process. This work proposes a conflict detector in non-functional requirement evolution (CDNFRE) system that uses ontologies as a theoretical foundation for automatically detecting conflicts. Requirement metadata and conflict detection rules and their associated requirement generation and conflict detection processes are proposed for the CDNFRE mechanism. A prototype is developed. A case study of electronic commerce in a television station company demonstrates the feasibility and effectiveness of the proposed CDNFRE system.

1. Introduction

Information systems development for the highly competitive consumer markets of today must consider intertwining technology and business requirements [3,32]. Technology and business requirements are numerous and constantly evolving. The hidden inconsistencies and conflicts are often difficult to identify in a large mass of requirements. Employee turnover is inevitable and may cause organizations to forget requirements. Therefore, formal requirement analysis is essential and requires computer support [20].

Customers are now focusing on non-functional requirement (NFR) [9,27]. Non-functional requirements determine the efficiency and effectiveness of a software system in completing a task [38]. Non-functional requirements are often presented chaotically without a common standard and without sufficient analysis [17]. Analyzing NFR conflicts is essential for successful information systems development [31]. Insufficient NFR analysis can cause serious problems in systems development [25,28].

To facilitate NFR analysis, this work proposes a Conflict Detector in Non-Functional Requirement Evolution (CDNFRE) mechanism for automatically detecting conflicts during maintenance of non-functional requirements by systems analysts. The CDNFRE tool includes four essential elements: metadata, ontology, cause-and-effect relationship, and conflict detection rule. Metadata can be conceptualized as a blank form, and ontology can be conceptualized as a dictionary that stores concepts and semantic relationships. By applying ontology concepts (vocabulary in the dictionary), the system analyst can provide the metadata (the blank form) needed to model non-functional requirements. Based on cause-and-effect relationships, inference knowledge can be used to derive requirements about side effects from ontology and metadata. In contrast, a conflict detection rule is a rule for using ontology and metadata to infer conflicts as NFRs evolve.

Non-functional requirements consist of software attribute, business value, and restriction [9,22]. Software attribute considers interface, performance, and quality. Business value includes goal, strategy, policy, and control. Restriction represents limitations and assumptions for constraining information systems, such as law, culture, and budge. Business value and restriction, which are related to software attribute, must be considered when determining software attributes. Business value is an important element of information system development in a profit-oriented organization [1,22]. For example, business considerations such as target market and brand position must be considered when designing the interface during electronic commerce websites design.

Ontology is typically offered to support a shared understanding [29]. An ontology includes both concepts and their relationships [4,10]. In the software engineering domain, application of ontology is increasingly common [12]. There are evidences of the benefits of using ontologies in requirement engineering activities both in industry and academy for reducing inconsistencies although the most existing studies addressed functional requirements [5]. One of the nice characteristics of ontologies is their capability of reasoning. The reasoning may help in detecting some type of inconsistencies [14]. Some studies [2,14] apply ontologies in functional requirements. Besides software attributes, other features should be included in NFR, such as business value, restriction, and cause-and-effect relationship which are not considered yet. To

E-mail address: tonyliu@mail.knu.edu.tw.

http://dx.doi.org/10.1016/j.csi.2016.03.002
0920-5489/© 2016 Elsevier B.V. All rights reserved.

fill this research gap, this work uses ontologies which consider various NFR features to detect conflict in non-functional requirements.

The rest of this paper is structured as follows. Section 2 discusses related works in requirement analysis. Section 3 introduces the proposed method of CDNFRE design. Section 4 describes the proposed ontologies for non-functional requirements. Section 5 presents the software architecture and database design of CDNFRE tool. Sections 6 and 7 describes the CDNFRE prototype and the results of a real-world case study to confirm its feasibility and effectiveness. Finally, the results are discussed, and conclusions are given.

2. Related works

The requirement metadata provides a requirement data format. Most of the existing works [8,11,15,16,21] focus on functional requirement metadata, and only Fletcher, Liu, and Tang [7]) have considered NFR metadata. Sadana and Liu [31]) proposed that NFR metadata can be described in terms of three main characteristics: quality attribute, functionality, and specific characteristic. In the metadata, a quality attribute of functionality can be presented as a specific characteristic. For example, the following requirement can be presented based on the above metadata: sound quality of music player should be excellent. Egyed and Grünbacher [6]) revealed that quality attributes often needs adjustment. One software attribute may influence another. Therefore, cause-and-effect relationships must be considered for an effective non-functional requirement analysis.

To compare existing requirement conflict analysis methods, several features including purpose, considering NFR, automatic tool, and configurable rule support are chosen based on the research goals of this work. Table 1 shows the conventional requirement conflict analysis methods, most of which provide tools for automatically detecting conflict. Three of the studies [7,15,16] listed in the table partially consider NFR, and no one discusses how configuration capability affects conflict detection rules. To fill the gap on low variety of NFR features and the lack of configurable rule support, this work focuses on conflict detection in various NFR features and how to provide configuration capability.

The NFR metadata proposed in this study are based on the works summarized in Table 2. In other words, seven features presented in Table 2 are revealed by the related works. Lee and Xue [19] and Lee et al. [18] integrated unified modeling language (UML) with goals, *i.e.*, non-functional requirements. Examples of goals are maintaining max number of users, optimizing performance, and optimizing flexibility.

In a study of software quality, Egyed and Grünbacher [6] argued that quality can be adjusted and that one quality can influence another. Robertson and Robertson [30]) defined the elements of a requirement as identification number, NFR, type, use case, description, rationale, originator, fit criterion, customer satisfaction and dissatisfaction, priority, conflicts, supporting material, and history. Customer satisfaction revealed by Robertson and Robertson [30]) is not included in this CDNFRE metadata because customer satisfaction can be represented by NFR concretely. Glinz [9]) defines a non-functional requirement as system attribute or system constraint.

Sadana and Liu [31]) proposed that NFR metadata can be defined in terms of quality attribute, module, and specific characteristic. Mylopoulos, Chung, and Nixon [26]) and Supakkul, Hill, Chung, Tun and Prado Leite [33]) showed that modules for increasing and decreasing NFR goals can achieve a sufficient satisfaction level. They also showed that side effects should be considered when formulating NFRs. This CDNFRE metadata provides plentiful NFR features comparing to the existing NFR metadata to fill the research gap.

3. Ontology-based NFR conflict analysis method

The CDNFRE system was designed to support ontology-based non-functional requirement conflict analysis (ONFRCA) method [22], which analyzes ontology-based non-functional requirement conflict in four steps: modeling prior knowledge, modeling new NFR, determining side effect, and detecting conflicts. Fig. 1 depicts the four steps of ONFRCA.

Constructivism theory is the theoretical foundation of the ONFRCA method. Constructivism indicates that the existing knowledge is the basis in interpretation. The new data should be considered and interpreted based on the existing knowledge to reason new information for adapting the external world. Therefore, modeling prior knowledge is the first step in the ONFRCA process. Modeling new NFR in step 2 is to consider new data. The prior knowledge and new NFR are used to reason new information, such as side effects and conflicts. The four steps of the ONFRCA process are summarized as follows.

(1) Modeling prior knowledge: this phase models the minimum prior knowledge needed to describe existing information systems. Prior knowledge comprising adopted NFR, cause-and-effect relationships, and ontologies that is manually proposed by employees and then approved by an organization according to the consensus of a management-level committee. Adopted NFR describes existing information systems. Cause-and-effect relationships must be identified to understand side effects. Ontologies contain the terms needed to describe NFR and cause-and-effect relationships. For example, an adopted NFR is “the atmosphere of the electronic commerce website should be relaxing”. Atmosphere is a kind of interface attribute, electronic commerce website is a module, and relaxing is a specific value for describing the atmosphere in the ontologies.
(2) Modeling new NFR: a major task in this phase is referencing prior knowledge by browsing the NFRs adopted so far. Ontologies include terms that describe new NFRs manually. The ONFRCA process adds new terms to the ontologies of prior knowledge by returning to the modeling prior knowledge phase. For example,

Table 1
Existing requirement conflict analysis methods.

Method	Purpose	Considering NFR	Automatic Tool	Configurable rule support
Hausmann, Heckel, and Taentzer [11])	Detecting conflicts in UML use case	Not disclosed	Yes	Not disclosed
Kim, Park, Sugumaran, and Yang [16])	Detecting conflicts in goal and scenario	Partially yes, goal is considered.	Yes	Not disclosed
Gervasi and Zowghi [8])	Detecting inconsistencies in if…then statement	Not disclosed	Yes	Not disclosed
Kaiya and Saeki [15])	Analyzing inconsistencies and incompleteness in requirement document	Partially yes, software attribute and restriction are considered.	Not disclosed	Not disclosed
Liu [21])	Detecting conflicts in UML activity diagram	Not disclosed	Yes	No, rules are coded in C#.
Fletcher, Liu, and Tang [7])	Using trade-off strategies to select services with conflicting non-functional attributes	Partially yes, software attributes are considered.	Yes	Not disclosed
CDNFRE	Detecting conflicts in non-functional requirement	Yes, software attribute, business value, restriction, and evaluation are considered.	Yes	Yes, the rule editor module and rule base are provided.