Advances in Engineering Software 54 (2012) 1–16

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

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

A graph based requirements clustering approach for component selection

Muhammad Ali Khan [a], Sajjad Mahmood [b,*]

[a] College of Applied and Supporting Studies, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[b] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

ARTICLE INFO

Article history:
Received 7 October 2011
Received in revised form 5 July 2012
Accepted 10 August 2012
Available online 12 September 2012

Keywords:
Component based software system
Software component selection
Signed graph clustering
Matching index
Signed graph
Requirements engineering
Usage dependency
Non-functional dependency
Threat dependency

ABSTRACT

In recent years, Component Based System (CBS) development has found widespread application, particularly in the domain of engineering software where it is often necessary to adapt and integrate existing software tools to handle new problems. The ability to identify suitable components that match system requirements is fundamental to CBS success. To date, CBS selection techniques often make an ideal assumption that there is one-to-one mapping between requirements and components. In reality, components are usually designed for general purposes and provide a range of features that can be adapted to meet the needs of a CBS. This implies that a component can potentially match more than one functionality of a CBS. On the other hand, system requirements are usually not independent of each other and a component selection process needs to consider the dependencies between system goals. Furthermore, the existing component selection methodologies usually provide a single solution for the component selection problem, thus limiting the options available to a CBS developer. In this paper, we present a component selection process that uses a signed graph to model interdependencies of CBS-to-be needs and groups related goals into clusters, based on the usage, non-functional and threat dependencies. Subsequently, the matching index of each cluster of related goals is used as a criterion to identify a portfolio of candidate components for a CBS, providing multiple solutions for the component selection problem whenever possible. The component selection process helps a developer to elicit stakeholder needs, analyze their interdependencies and select components for a CBS. We also present application of our approach to a Meeting Scheduling System (MSS) and a Construction Management System (CMS).

1. Introduction

Over the last couple of decades, the widespread use of software has placed new expectations on the software industry [1] and there is an ever growing push towards software reuse. CBS development is an approach that aims to move the software industry away from developing each system from scratch. It focuses on integrating existing components to build a software system, with the potential benefits of reducing development time and delivering quality system by using quality components. Due to these advantages, CBS development has been utilized in a variety of engineering applications such as computer-aided design and manufacturing [2], biomedical modeling [3], finite element software [4] and the list keeps growing. The success of a CBS project requires a collaborative process in which both system stakeholders and candidate components balance the conflicting interests between what is needed and what is available [5,6]. This collaborative process needs to focus on negotiating individual interests during the component selection phase of a CBS. In this paper, we adopt Szyperski's definition of a component [7]: '*A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties*'.

Recent research [8,9] shows that CBS success depends on the ability to identify suitable components. Inappropriate component selection can lead to adverse affects such as shortlisting components that barely fulfill the needed functionality or introduction of extra costs in integration and maintenance phases [10]. The component selection process is further complicated by the fact that individual components usually provide fixed capabilities and it may not be possible to satisfy all system needs by the available components. On the other hand, components can sometimes provide additional features that may not be needed (or even desirable) in a given system. Furthermore, component selection is usually carried out by dealing with unstructured information on the web [11,12], which makes detailed evaluation of all possible candidate components highly costly and impractical.

In line with traditional systems, stakeholder requirements for a CBS are usually not independent from each other and more than

* Corresponding author. Tel.: +966 38607698.
E-mail addresses: malikhan@kfupm.edu.sa (M.A. Khan), smahmood@kfupm.edu.sa (S. Mahmood).

0965-9978/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2012.08.002

one requirement can collaborate to achieve certain features of a CBS. Hence, the component selection process cannot assume that there will always be one-to-one mapping between requirements and candidate components. Although consolidation of requirements into clusters, based on their interdependencies, is essential to understand such a process, it is not sufficient on its own as it may be impossible to find a suitable component to match a consolidated requirement. An assessment of the availability of matching components is therefore essential to validate the practicality of requirements clustering. We believe that the CBS selection process needs to focus on how to analyze interdependent requirements, consolidate related requirements in a meaningful way and facilitate shortlisting of components that match these consolidated requirements. Additionally, whenever possible the selection process should provide multiple solutions for the component selection problem to offer greater flexibility to a CBS developer. This implies that component selection should not be opportunistic; rather it should be carefully planned, based on requirements dependencies and availability of components.

In this paper, we present a component selection process that provides guidelines for CBS developers to consolidate related requirements into clusters and shortlist components that best match the required functionalities of a CBS. The component selection process developed in this paper consists of three phases: (i) *goal-oriented specification* which elicits CBS-to-be needs; (ii) *dependency analysis* that focuses on analyzing the dependencies between CBS-to-be needs; and (iii) *cluster analysis* that organizes interdependent CBS-to-be needs into clusters and uses the *matching index* to identify candidate components for a CBS.

We adopt the goal elicitation and refinement rules presented in [13,14] for phase (i). In phase (ii), we introduce three types of goal dependencies, namely *usage*, *non-functional* and *threat* dependencies based on the types of interaction between CBS goals. In phase (iii), we define the *goal dependency graph*, which is a signed graph, to represent the relationship between CBS goals and modify the local optimization signed graph clustering algorithm [15] to cluster interdependent goals together while also taking into account the availability of suitable components. Signed graphs have often been used to capture positive and negative relationships in networks [15–18]. Thus the signed graph representation lends itself to CBS goal analysis, as goals can interact positively or negatively with each other (see Section 5). In addition, the proposed representation and clustering method yields a list of candidate components that can be further evaluated for selection. We also present applications of our approach to Meeting Scheduling System (MSS) and Construction Management System (CMS) case studies. The main contributions of our work include:

- developing a dependency analysis technique for CBS goals,
- introducing the concept of goal dependency graph,
- adapting signed graph clustering for consolidating CBS goals,
- introducing the concept of matching index to select components for a CBS, and
- providing multiple candidate components for a CBS, whenever possible.

The remainder of this paper is organized as follows. Section 2 reviews related literature. In Section 3, we give an outline of our approach. Section 4 describes the goal-oriented specification while Section 5 presents the goal dependency analysis. In Section 6 we describe our clustering algorithm. Case studies are presented in Sections 7 and 8. In Sections 9 and 10, we discuss some observations based on the findings of our case studies and propose directions for future work.

2. Related work

2.1. Component based systems overview

CBS development is an integration centric methodology with emphases on using pre-fabricated components to build a software application. Qureshi and Hussain [19] argued that the traditional software process models are not adequate for CBS development and pointed out the drawbacks of the existing CBS development models. They proposed a CBS development model by modifying the object oriented process model [20]. In addition to traditional waterfall model phases [21], the CBS development needs additional phases of component selection, engineering and testing; and evaluation [19]. Fig. 1 gives an overview of the CBS development life cycle, adapted from [19].

The CBS development process can be divided into five main stages, namely, communication, planning, analysis and component selection, engineering and testing, and evaluation phases. The first phase, communication, gathers the overall objectives of the CBS-to-be. The communication phase is carried out at the start of a project to collect the basic requirements and initial use cases of the CBS-to-be. The planning phase focuses on preparing the project specification document. The project specification document is used for the feasibility and risk assessment of a CBS project.

The third phase, analysis and component selection, consist of two sub steps namely, 'analysis' and 'component selection'. The 'analysis' sub step helps in gathering detailed requirements of a CBS. The dependencies between requirements are also analyzed in this phase of the CBS development life cycle. The 'component selection' sub step uses the detailed requirements to select suitable components from the components repository. The fourth phase deals with integration and testing process by focusing on adapting the selected components and writing glue-code to assemble them into a CBS. Finally, the fifth phase, evaluation, deals with customer evaluation and verification of a CBS with respect to stakeholder requirements. In this paper, we are only concerned with the 'analysis and component selection' phase.

2.2. Requirements analysis and component selection

Requirements analysis and component selection are recognized as entwined processes that play a central role in the overall

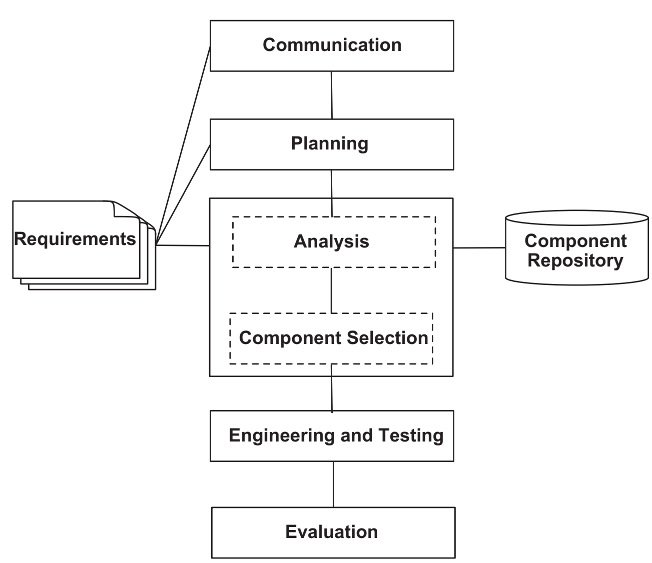

Fig. 1. The CBS development phases (adapted from [19]).