Computers & Industrial Engineering 115 (2018) 278–289

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A multi-objective optimization model of component selection in enterprise information system integration

Lifeng Mu[a,*], C.K. Kwong[b]

[a] SHU-UTS SILC Business School, Shanghai University, 20 Chengzhong Road, JiaDing District, Shanghai 201899, PR China
[b] Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, The Hong Kong Polytechnic University Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Keywords:
OR in software engineering
Flexible IT architecture
Enterprise information technologies
Component selection
Multiple objective programming

ABSTRACT

Integrating legacy IT assets and new commercial software components together into a flexible IT architecture is one of open challenges facing modern enterprises today. Most of previous studies focused on using re-engineering to improve the flexibility of IT architectures, rather than employing optimization theory in architecture design problem, especially the problem of component selection and re-allocation in IT architecture. Moreover, a scant amount of literature is available on considering the architectural flexibility and integration cost simultaneously. To fill in this gap, based on a modified quantitative method of measuring the relationship between couplings and cohesions in architecture, we devise a nonlinear multi-objective binary integer programming to select components from legacy candidates and commercial candidates, and to group them into services under the service-oriented architecture (SOA) environment. The customized SPEA2 algorithm is further used to solve the problem, and some managerial insights are provided based on experiments and sensitivity analysis with the model.

1. Introduction

After years of changes in business logics, various information submodules, implemented by using different platforms and technologies, have been introduced into Enterprise Information System (EIS). Consequently, EIS tends to be too heterogeneous and complex to react agilely to market changes and business process reengineering, and a number of legacy assets are now becoming obsolete. Recently, the booming market of commercial component gives enterprises an opportunity to introducing the state of the art. Whereas, owing to long periods of existence of legacy assets, they have often undertaken business-critical applications and cannot be replaced inexpensively. Therefore, the cost-effective integration of legacy systems with new technology becomes an inevitably challenge to IT executives. To cope with the complexity of EIS integration, researchers in industry and academia have made numerous attempts, including electronic data interchange (EDI), abstract data types, structured programming, object orientation, design patterns, and modeling languages, etc. Over the last few years, a service-oriented approach, named as service-oriented architecture (SOA), has become increasingly popular. As an effective technology of information system integration, SOA is based on the concept of a service, which is normally composed of databases, new software components, legacy modules, and other IT assets (for convenience, the term "component" is used to represent all these assets inside services). Services can be deployed geographically, and can be integrated with each other to achieve a loosely coupled architecture, through a wide variety of platform independent service interfaces.

However, after years of SOA participation, CIOs (Chief Information Officers) have realized that the technique is not the only issue of the system integration – some managerial decisions also play important roles (Şen, Baracli, Şen, & Basligil, 2009), which include the following:

(1) The enterprise has made extensive investments in IT assets over the course of many years. So, is it wise to replace outdated components with new commercial components? Or keep leveraging them in spite of their limited functionality and performance?
(2) Which legacy components should be encapsulated and reused into the final EIS, and which ones should be replaced by commercial substitutes for introducing new functionalities and business logics?
(3) In addition to economic motivation, the flexibility of EIS architecture also needs to be deliberated in strategic decision. In order to achieve a flexible architecture with acceptable cost, how to select, group and deploy candidate components (including legacy candidates and new commercial candidates) into services?

* Corresponding author.
E-mail addresses: mulifeng@shu.edu.cn (L. Mu), c.k.kwong@polyu.edu.hk (C.K. Kwong).

https://doi.org/10.1016/j.cie.2017.11.013
Received 16 April 2015; Received in revised form 26 October 2017; Accepted 12 November 2017
Available online 21 November 2017

To the best of our knowledge, there is scant literature on addressing above-mentioned managerial issues. In this paper, we propose, for the first time, the mathematical model of component selection which considers the architectural flexibility and integration cost at the same time. To be more specific, given the information of a new architecture (including the number of services, required components to be implemented), the set of candidate components (including legacy components and commercial components) and relationships among candidate components, how to select components from candidate components to implement required components in order to design a flexible architecture with minimal integration cost.

The remainder of this article is organized as follows. Although no previous studies have focused on all the aforementioned issues simultaneously, separate issue has been considered in many literatures, and these literatures are reviewed briefly in Section 2. The component selection problem under SOA environment is described in Section 3. In Section 4, We introduce a new quantitative measure of architectural flexibility. Corresponding mathematical description and a multiple objective optimization model are provided in Section 5. A customized SPEA2 algorithm is designed to solve the multi-objective optimization problems in Section 6. We also provide an illustrated example in Section 7, and discuss managerial implications in Section 8. Finally, we conclude this research and suggest future research opportunities in Section 9.

2. Literature review

2.1. Reengineering and optimization of legacy EIS

Improving a software system's quality to easily accommodate future business changes is the primary goal of similar studies. To achieve this goal, previous studies mainly adopted two methods – reengineering and optimization. In the field of reengineering, Sarkar et al. (2009) proposed a modularization approach of reengineering to avoid deviating from the intended architecture and deteriorating into unmanageable monoliths. Based on a specific reengineering tool, Kienle and Muller (2010) discussed software reverse engineering, exploration, visualization, and documentation. In order to extend the lifetime of legacy systems, Canfora, Fasolino, Frattolillo, and Tramontana (2008) employed a wrapping approach to migrate legacy system interactive functionalities to SOA. To achieve integrity and consistency, and effective conformance in architectures of large-scale systems, Jung and Hatcliff (2010) presented a type-centric framework for specifying heterogeneous component-oriented architectures.

In addition to reengineering approaches, optimization methods also are employed to address issues of architecture reconstruction. For example, Walker et al. (2013) used an automatic architectural optimization to assist designers in exploring many different options and evaluating them according to specific criteria. To seek cost-effective solutions to the rapid obsolescence of software and the increasing demand for new functionality, Fuentesfernandez, Pavon, and Garijo (2012) proposed a model-driven process for the modernization of component-based systems. Likewise, Tran et al. (2012) proposed a model-driven and view-based optimization approach to achieve compliance in service-oriented architectures. Umar and Zordan (2009) introduced a decision model for SOA reengineering projects that combines strategic and technical factors with cost benefit analysis for integration versus migration decisions.

2.2. Make-or-buy and component selection

Previous studies have discussed the topic of component selection mainly from four aspects: factors, identification, evaluation and selection.

A significant breakthrough in the area of factor evaluation is the work of Daneshgar, Low, and Worasinchai (2013). They systematically investigated factors that affect "build vs. buy" decision for software acquisition. They conducted rounds of data collection and analysis: first, a pilot study for deriving additional factors that may be specific to SMEs, and a follow up main set of semi-structured interviews in 8 SMEs for further confirmation. It was found that some of the factors that affect the decision process for large organizations also apply to the SMEs whereas some factors do not. More specifically, the study found that the following factors affect both SMEs and large organizations: requirements fit, cost, scale and complexity, commoditization/flexibility, time, in-house experts, support structure, and operational factors. Factors mainly applying to large organizations were strategic role of the software, intellectual property concerns, and risk, Factors particularly relevant to SMEs. Based on and inspire from their work, aiming to address software selection problem for EIS of large organizations, we adopt two primary factors (cost and architectural flexibility) as optimization targets.

In addition, in order to identify suitable components, Hasheminejad and Jalili (2014) introduced a novel method to select components from legacy systems. For the same purpose, Sanchez, Oliveira, Barbosa, and Henriques (2015) proposed an approach to identify concrete patterns of interaction from legacy code at a higher level of abstraction. Cui and Chae (2011) applied an agglomerative hierarchical clustering algorithm for component identification in legacy systems.

In terms of evaluation, Palviainen, Evesti, and Ovaska (2011) developed an approach to evaluate, predicate and measure software components, and Hasheminejad and Jalili (2014) considered cohesion and coupling metrics to evaluate legacy components from an information system.

Depending on research methodologies adopted, previous studies can be further classified in terms of framework, optimization model, and algorithm, respectively.

In aspect of framework, Lizcano, Alonso, Soriano, and Lopez (2014) presented a visual framework to offer end-to-end composition to empower end users to develop their own applications, and Mudiam, Gannod, and Lindquist (2006) described an architecture-based approach to the synthesis of services from legacy components and their subsequent integration with service-requesting client applications.

In the matter of mathematical model, researches focused on how to build rational mathematical models to reflect real-world problems. For instance, in order to address the problem of undertaking multiple application tasks in a cost-efficient manner, Tang, Mu, Kwong, and Luo (2011) introduced an optimization model to select appropriate software components. Şen et al. (2009) proposed an integrated decision model dealing with qualitative and quantitative objectives for software component selection.

In terms of algorithm, in order to select components that best fit requirements, Clark et al. (2004) discussed search techniques in selecting components in large commercial off-the-shelf (COTS) repositories. Naseem, Maqbool, and Muhammad (2013) designed a cooperative clustering algorithm for software modularization. Liao, Liu, Zhu, and Wang (2014) introduced a sub-swarms particle swarm optimization algorithm for service composition. Fan, Fang, and Jiang (2011) developed a new co-evolution algorithm to solve the problem of web service selection and composition.

Previous studies have made great contributions to addressing the issues of component selection in EIS integration. However, the separate consideration of the integration of legacy components and the introduction of new commercial components usually cannot yield an effective and operative decision-making method. Unlike conventional researches which were concerned with reengineering legacy assets (Canfora et al., 2008; Cui & Chae, 2011; Fuentesfernandez et al., 2012; Hasheminejad & Jalili, 2014; Kienle & Muller, 2010; Sanchez et al., 2015; Sarkar et al., 2009; Tran et al., 2012; Umar & Zordan, 2009) and the construction of a new information system (Daneshgar et al., 2013; Palviainen et al., 2011; Tang et al., 2011; Walker et al., 2013; Şen et al., 2009) separately, the inheritance of legacy assets and the introduction