INTRODUCTION

Modern software systems are becoming more and more large-scale, complex and unreasonably controlled, resulting in high development cost, low productivity, unmanageable software quality and high risk to move to new technology. Consequently, there is a growing demand of searching for a new, efficient and cost-effective software development paradigm. One of the most promising solutions today is the component-based software development (CBSD) approach. This approach is based on the idea that software systems can be developed...
by selecting appropriate commercial-off-the-shelf (COTS) components and then assemble them to fit a specific architectural style for some application(s) domain. A COTS component can be developed by different developers using different languages and different platforms. In general, a COTS component has three main features: (i) a component is an independent and replaceable part of a system that fulfills a clear function; (ii) a component works in the context of a well-defined architecture; and (iii) a component communicates with other components of the software system through its interfaces (Brown & Wallnau, 1998). In CBSD, the main focus is how to choose the most appropriate and most suited component from COTS components’ market so that it can significantly reduce development cost and time-to-market, and improve maintainability, reliability and overall quality of software system. Several COTS selection methods (Chung, Cooper, & Courtney, 2004; Grau, Carvallo, Franch, & Quer, 2004; Kontonya & Hutchinson, 2004; Leung & Leung, 2002; Rolland, 1999) have been proposed in literature. However, it may be noted that there is no single method which is accepted as a standard COTS selection method. A detailed list of the COTS selection methods has been provided in Mohamed, Ruhe, and Eberlein (2007).


All the previously mentioned optimization models are based on the assumption that in the software development process, the software developer (i.e., decision maker) has complete information. It may be noted that since software development is not an exact science, there are often plenty of indefinite and uncertain factors in the parameter estimation of the COTS selection problem. Hence, the various model parameters are often imprecise or the process of estimation of these input parameters is subjected to uncertainty. In our opinion, the best way to deal with uncertainty in mathematical modelling is to incorporate uncertainty in the model itself. Corresponding to the various possible scenarios of uncertainty in the model parameters, we must explore different possible outcomes and then select the best outcome in a given decision making situation. The optimization models in the software development have benefited greatly from the fuzzy set theory (FST) (Zadeh, 1965) in terms of integrating quantitative and qualitative information, subjective preferences of the decision maker and knowledge of the software experts. Jha, Bali, and Kumar (2011) presented a fuzzy approach for optimal selection of COTS components for modular software system under consensus recovery block scheme incorporating execution time. Gupta, Mehlawat, Mittal, and Verma (2009) formulated fuzzy multiple-objective optimization model for the COTS selection using nonlinear S-shape membership functions describing vague aspiration levels of the decision maker in respect of the weighted quality and cost. Gupta, Verma, and Mehlawat (2011)
Related Content

Positioning Against Other Theories
Lars Taxén (2010). *Using Activity Domain Theory for Managing Complex Systems* (pp. 135-161).
[www.igi-global.com/chapter/positioning-against-other-theories/39676?camid=4v1a](www.igi-global.com/chapter/positioning-against-other-theories/39676?camid=4v1a)

Engineering and Reengineering of Technology Enhanced Learning Scenarios Using Context Awareness Processes
[www.igi-global.com/chapter/engineering-reengineering-technology-enhanced-learning/56071?camid=4v1a](www.igi-global.com/chapter/engineering-reengineering-technology-enhanced-learning/56071?camid=4v1a)

Methods for Developing Flexible Strategic Information Systems: Is the Answer Already Out There?
[www.igi-global.com/chapter/methods-developing-flexible-strategic-information/4225?camid=4v1a](www.igi-global.com/chapter/methods-developing-flexible-strategic-information/4225?camid=4v1a)
Applying Probabilistic Adaptation to Improve the Efficiency of Intra-Query Load Balancing


[www.igi-global.com/article/applying-probabilistic-adaptation-improve-efficiency/75548?camid=4v1a](www.igi-global.com/article/applying-probabilistic-adaptation-improve-efficiency/75548?camid=4v1a)