A Method for Compatible COTS Component Selection

Jesal Bhuta and Barry Boehm

Center for Software Engineering, University of Southern California,
Los Angeles, California 90089, USA
{jesal, boehm}@cse.usc.edu

Abstract. Software projects involving integration of multiple commercial as well as in-house components, often confront interoperability problems. This is a result of the component selection process being limited to piecewise evaluation of system capabilities while neglecting a more thorough evaluation of interoperability between candidate components. Such problems often lead to increased costs and schedule overruns. Based on empirical data gathered from five years of developing e-services applications at USC-CSE, we have developed and applied a method for component selection that focuses on piecewise evaluation, as well as the interoperability between the candidate components. In this paper we describe the method and present a real-world example showing how it operates within the spiral process model generator.

1 Introduction

Projects that require COTS, legacy, reusable and custom component integration often confront interoperability problems, where one component may not function well with the other. Such problems if detected late during the project development life cycle can result in increased costs and overrun schedules. Many development projects succumb to such issues because they do not consider the interoperability between components as a criterion when selecting components for system development. The selection instead is primarily based on piecewise evaluations of the system capabilities each component satisfies.

From our experiences in e-services projects we have observed that projects that have mitigated the risk of component interoperability earlier in the project development cycle have been more successful during the integration phases. However, those that neglected the component interoperability issues during their component selection cycle had to do a considerable amount of additional work in developing complex glueware to integrate the components, or re-work to replace certain incompatible components.

In this paper we propose a method that will help development teams reduce the risk of component incompatibilities during component assessment cycles. Section 2 of this paper describes the background and related work. Section 3 provides the definitions of the terms commonly used in this paper. Section 4 describes the method as a process element within the USC COTS process framework [14], while section 5 explains the elements that will facilitate this process. Section 6 illustrates the use of the method with a project example.

X. Franch and D. Port (Eds.): ICCBSS 2005, LNCS 3412, pp. 132 – 143, 2005.

2 Background and Related Work

In [1] and [21] researchers argue that interoperability problems amongst components occur due to multitude of reasons including functional mismatch, non-functional mismatch, architectural mismatch, component conflicts and interface conflicts [19][20][33]. For example, Garlan's Aesop [21] project found that the architectural mismatches among four COTS components caused a factor of four overrun in schedule (6 months to 2 years) and a factor of five increase in cost (1 to 5 person-years). Most contributions to reducing the problems have been product oriented [1][17][19][20][24][30][31]. Some COTS-based development papers have addressed the interoperability issue at a high level [3][4][14][15][22][26][28][29]; this paper provides a more specific method for selecting interoperable components.

In [5] researchers, based on empirical results, argue that the effort per line of glue code averages about three times the effort per line of developed applications code. Additionally they assert that the development and post deployment efforts can scale as high as the square of the number of independently developed COTS products targeted for integration. This empirical evidence is a compelling reason to provide CBA developers with better process for mitigating the integration risks, and reducing component integration effort and cost.

In [14] we have provided a recursive and re-entrant USC Composable COTS Based Application (CBA) process elements framework, which helps the developers design a life-cycle development plan using the spiral model to evaluate, tailor and integrate components in a CBA project. This paper is provides an extension to the framework, which focuses on activities for identification of a compatible set of COTS components to build the CBA.

3 Definitions

We adopt the SEI COTS-based System Initiative's definition [15] of a COTS product: A product that is:

- Sold, leased or licensed to the general public,
- Offered by a vendor trying to profit from it,
- Supported and evolved by the vendor, who retains the intellectual property rights,
- Available in multiple identical copies,
- Used without source code modification.

In [15], SEI also describes a COTS-based System (CBS) very generally as "any system, which includes one or more COTS component." The definition above for COTS-based Systems includes most current systems, including those that depend upon operating systems, or similar relatively stable frameworks. However, COTS considerations do not significantly affect the development cycle for such systems. Alternately we define a COTS Based Application (CBA) as a system for which 30% of the end-user functionality is provided by COTS components, and at least 10% of the development effort is devoted to COTS considerations. The numbers 30% for end-user functionality and 10% for development effort are approximate behavioral CBA