Methods for Measurement-Based COTS Assessments and Selection *

Alejandra Cechich[1] and Mario Piattini[2]

[1]*Departamento de Informática y Estadística – Universidad Nacional del Comahue*
Buenos Aires 1400, Neuquén, Argentina
phone: (54) 299 - 4490312, fax: (54) 299 4490313
E-mail: acechich@uncoma.edu.ar

[2]*Escuela Superior de Informática - Universidad de Castilla-La Mancha*
Paseo de la Universidad 4, Ciudad Real, España
phone: (34) 926 295300, fax: (34) 926 295354
E-mail: Mario.Piattini@uclm.es

Abstract. During recent years, new software engineering paradigms like component-based software engineering and COTS-based development have emerged. Both paradigms are concerned with reuse and customisation of existing components. The use of COTS software has become more and more important in state-of-the-art and state-of-the-practice software and system development. Using COTS software promises faster time-to-market, which can yield substantial advantages over competitors with regards to earlier placement of a new product on a market. At the same time, COTS software introduces risks such as unknown quality properties of the COTS software in use that can inject harmful side effects into the final product. In this paper, we present a brief comparison of well-known COTS evaluation methods. Our work is focused on identifying the main goals and features of the methods in order to extend their uses on evaluating COTS products as well as COTS development processes.

1. Introduction

Typically, COTS-based development - in parallel to traditional development process models, e.g. waterfall, spiral - consist of four phases: (1) COTS assessment and selection, (2) COTS tailoring, (3) COTS integration, and (4) maintenance of COTS and non-COTS parts of the system. The first phase — COTS assessment and selection — is the most crucial phase in the COTS-based cycle. Here long-term decisions on which COTS will be used in a software system are made. Non-optimal COTS software used in the development of a system can become extremely costly for a software organisation. Apart from the general reuse problems (selection, integration, maintenance, etc.), COTS products have by their own specific problems [1]:

- Incompatibility: COTS components may not have the exact functionality required; moreover, a COTS product may not be compatible with in-house software or other COTS products.
- Inflexibility: usually the source code of COTS software is not provided, so it cannot be modified.
- Complexity: COTS products can be too complex to learn and to use imposing significant additional effort.
- Maintainability: different versions of the same COTS product may not be compatible, causing more problems for developers.

* This work is partially supported by the CYTED (Ciencia y Tecnología para el Desarrollo) project VII.J-RITOS2 (Red Iberoamericana de Tecnologías de Software para la década del 2000).

Therefore, repeatable and systematic methods to assess and select COTS software are an important issue in COTS-based software engineering. Currently, there exist several solutions for COTS selection and the following section gives a brief summary of existing methods to COTS assessment and selection.

2. Methods for COTS Assessment and Selection

The **OTSO (Off-The-Shelf-Option)** method and some experiences of its use are presented in [2, 3]. The method supports the search, evaluation and selection of reusable software and provides specific techniques for defining evaluation criteria and comparing the costs and benefits of product alternatives. In OTSO method, the evaluation criteria are gradually defined as selection process progresses. The evaluation criteria are derived from reuse goals and factors that influence these goals [4]. The evaluation criteria definition process in OTSO essentially decomposes the requirements for the COTS software into a hierarchical criteria set. Each branch in this hierarchy ends in an *evaluation attribute*: a well-defined measurement or a piece of information that will be determined during evaluation. This hierarchical decomposition principle has been derived from Basili's GQM [5, 6] and Saaty's approach [7].
The *search* in the selection process attempts to identify and find all potential candidates for reuse. The objective of the *screening* process is to decide which alternatives should be selected for more detailed evaluation. The objective of the *evaluation* process is to evaluate the selected alternatives by the evaluation criteria and document evaluation results. Evaluation produces data on how well each alternative meets the criteria defined. As the COTS alternatives have been evaluated the evaluation data needs to be used for making a decision. The analysis of results relies on the use of the Analytic Hierarchy Process (AHP) for consolidating the evaluation data for decision-making purposes [7].

The **CAP (COTS Acquisition Process)** method [9, 10] consists of three components: the CAP Initialisation Component (CAP-IC), the CAP Execution Component (CAP-EC), and the CAP Reuse Component (CAP-RC).
The first step in CAP-IC is the identification of criteria against which candidate COTS software alternatives must be evaluated (CAP activity "Tailor & Weight Taxonomy"). In this activity the requirements are translated into taxonomy of evaluation criteria and prioritised (or weighted) according to the Analytic Hierarchy Process (AHP) under incorporation of multiple stakeholder interests.
The second step is to estimate how much effort will probably be needed to actually apply all evaluation criteria to all COTS software candidates (CAP activity "Estimate Measurement Effort").
The third step is to set up the measurement plan according to which all evaluation activities will be conducted (CAP activity "Elaborate Measurement Plan"). The measurement plan is either designed straightforward from the taxonomy of evaluation criteria - in the case the measurement effort estimates for measurement satisfy the budget and resource constraints. Alternatively, the measurement plan is constructed by employing optimisation algorithms with the objective to maximise priority coverage in the measurement plan. Finally, a review step certifies that all CAP-IC activities have been conducted correctly (CAP activity "IC-Review").

PORE (Procurement-Oriented Requirements Engineering) [11] uses an iterative process of requirements acquisition and product evaluation/selection as its main approach for Component-based Systems Engineering (CBSE). The basic PORE life-cycle process model has six generic processes. They are defined at 3 levels according to Humphrey's process model; hence, the heart of PORE is an iterative and parallel process of requirements acquisition and product evaluation and

selection. Within this iterative process, the PORE method integrates different methods, techniques and tools for requirements acquisition and product identification and evaluation with process guidance for choosing and using each technique such as feature analysis techniques, MCDM (Multi-Criteria Decision Making) techniques, argumentation techniques, requirements acquisition techniques, requirements engineering methods, etc. As well as integrating these techniques, PORE also provides guidelines for designing product evaluation test cases and for organising evaluation sessions.
The PORE method uses the requirement model to both acquire and elaborate the requirements statements and to check requirement-product compliance during the product evaluation and selection process. In the CBSE development process, this view is even taken further to include information about product suppliers such as their technical capabilities, application domain experience, ISO standard certification, CMM level, etc. and legal issues involved in product procurement such as negotiation contract terms and conditions, licensing arrangements, etc.

CEP (Component Evaluation Process) [12,13] defines a tailorable process and associated methods to evaluate components for inclusion into a system. At the start of a project, the Component Capability Database is searched for any components that already exist in the system, components that have been dictated or preselected for use in the system, or known candidate components for the system. There are five primary activities in the Component Evaluation Process, a scope of the activities to be performed, a search and screening of components that are candidates for inclusion into the system, the definition of the evaluation criteria, the evaluation itself, and, the analysis of the evaluation results. Decision support methods have been adapted for use with the Component Evaluation Process and may be used in activities ranging from search and screening of candidate components to analysing the evaluation results.
The Evaluate Component activity in the decomposition contains the following activities:

- *Scope Evaluation Efforts*, determines the scope for the activities involved in the Component Evaluation Process.
- *Search and Screen Candidates*, conducts a search of component sources, both internal and external, for all potential candidate components to be considered for inclusion in the system. This includes components that already exist in the system, have been preselected for the system, or dictated for use in the system.
- *Define Evaluation Criteria*, uses the initial search criteria, including several characteristics to produce detailed evaluation criteria necessary to conduct the evaluations of the component alternatives.
- *Evaluate Component Alternatives*, uses the detailed evaluation criteria to perform the evaluation on the selected component alternatives.
- *Analyse Evaluation Results*, performs the analysis of the evaluation results for the evaluations performed on the selected component alternatives.

3. Major issues

The OTSO method starts from the same input as CAP but appeared too much consuming effort. Furthermore, OTSO does not allow for a smooth and dedicated change of COTS evaluation depths. The OTSO method supports systematic evaluation of COTS alternatives and considers both financial and qualitative aspects of the selection process. The experiences in several cases [3, 8] indicate that the method is feasible in operational context although more formal experiments are required to validate the method.
PORE method focuses on requirements engineering and COTS procurement at the same time, thus its scope is broader than the other methods. One of the problems of the PORE method is that the

iterative process of requirements acquisition and product evaluation/selection is very complex. At any point, a large number of possible situations can arise, or a large number of processes and techniques to use in a single situation can be recommended. To handle this scale of complexity, a prototype tool known as PORE Process Advisor is being developed. The main components of the tool are a process engine which analyses the current set of goals to be achieved, model properties (inferred by the situation inference engine) and instructions from the requirements engineering team to recommend process advise.
CEP is an advancement of OTSO. There are several enhancements, e.g. planning facility for evaluating the available component alternatives, an activity for developing, executing, analysing evaluation scenarios, and the use of an additional mathematical method for evaluation data analysis. The Component Evaluation Process is a standalone process; however, it also may be used in conjunction with a higher-level development process.
CAP is also based on OTSO, but strictly Measurement-oriented, allowing for optimisation of the evaluation towards cost-efficiency, systematic changes of evaluation depth, and spiral model-like enactment. CAP defines a procedure for evaluating and selecting COTS software. The starting point for this decision procedure is an initial set of COTS software alternatives and a measurement plan that defines which criteria have to be measured and evaluated in order to identify the most suitable COTS software. However, simply measuring all applicable criteria on all COTS software alternatives can be expensive since (i) many COTS software alternatives might be available, (ii) the set of evaluation criteria could be quite large, and (iii) some of the criteria might be very difficult or expensive to measure, e.g. reliability. The effectiveness of CAP depends on the expressiveness of the criteria selected for evaluation. A trade-off between the effectiveness of the evaluation criteria and the cost, time, and resource allocation of the criteria while measuring must be reached.
All methods form a more or less waterfall-like process for reusable component evaluation and selection. It is based in all cases on general-purpose criteria without going into deeper characterisations.

Conclusions and Future Work

We have presented brief comparison of existing methods for assessment and selection of COTS components. We know that a late recognition that non-optimal COTS software was used in the development of the system can become extremely costly for a software organization. However, existing methods are specially focused on general properties of components instead of domain-oriented properties. A domain-oriented focusing could produce a more accurate evaluation since properties of components could be more precisely defined. In the next stage of our work, we are adapting traditional COTS evaluation methods, such as the OTSO method or the CAP method, to include characteristics of products in e-commerce applications. Our extension will consider aspects according to ISO standard 9126 on Product Quality [14] applied to COTS [15] including detailed metrics as well as other specific proposals for e-commerce.

References

[1] Buschmann R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-Oriented Software Architecture - A system of Patterns, John Wiley & Sons, 1996
[2] Kontio Jyrki.: OTSO: A Systematic Process for Reusable Software Component Selection, University of Maryland, Technical Report UMIACS-TR-95-63, 1995.
[3] Kontio J., Chen S., and Limperos K.: A COTS Selection Method and Experiences of its Use, 20th Annual Software Engineering Workshop, NASA Software Engineering Laboratory, Greenbelt, MD, 1995.
[4] Kontio J., Caldiera G., and Basili V.: Defining Factors, Goals and Criteria for Reusable Component Evaluation, Proceedings of CASCON'96 Conference, Toronto, Canada, 1996.

[5] Basili V.: Software Modelling and Measurement: The Goal/Question/Metric Paradigm, University of Maryland, Technical Report CS-TR-2956, 1992.
[6] Basili V. and Rombach H.: Tailoring the Software Process to Project Goals and Environments, Proceedings of the 9th International Conference on Software Engineering, IEEE Computer Society Press, 1987, pp. 345-357.
[7] Saaty T. L.: The Analytic Hierarchy Process, McGraw-Hill, 1990.
[8] Kontio Jyrki: A Case Study in Applying a Systematic Method for COTS Selection, Proceedings of the 18th International Conference on Software Engineering, Germany, 1996.
[9] Ochs M., Pfahl D., Chrobok-Diening G., and Nothhelfer-Kolb: A Method for Efficient Measurement-based COTS Assessment and Selection - Method Description and Evaluation Results, Fraunhofer Institut Experimentelles Software Engineering, IESE-Report No. 055.00/E, Version 1.0, 2000.
[10] Ochs M., Pfahl D., Chrobok-Diening G., and Nothhelfer-Kolb: A COTS Acquisition Process: Definition and Application Experience, Fraunhofer Institut Experimentelles Software Engineering, IESE-Report No. 002.00/E, 2000.
[11] Ncube C. and Maiden A.: PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm, http://www.sei.cmu.edu/cbs/icse99/papers/11/11.htm
[12] Polen S., Rosen L., and Phillips B.: Component Evaluation Process, Software Productivity Consortium, Report SPC-98091-CMC, Version 01.00.02, May 1999.
[13] Polen S., Rosen L., and Phillips B.: Component Evaluation Process, Appendix G: CEP in ETVX Notation, Software Productivity Consortium, Report SPC-98091-CMC, Version 01.00.02, March 2000.
[14] ISO/IEC 9126 Standard, Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their Use, International Organization for Standardization, (Ed.) Geneva, 1991
[15] Bertoa M. and Vallecillo A., Atributos de Calidad para Componentes COTS, V Workshop Iberoamericano de Requisitos y Ambientes de Software, La Habana, Cuba, 23-26 April 2002.