Selecting COTS Components: A Comparative Study on E-Payment Systems

Karina Taryano and Alejandra Cechich

Departamento de Ciencias de la Computación – Universidad Nacional del Comahue
Buenos Aires 1400, 8300 Neuquén, Argentina
E-mail: ktaryano@dprneuquen.gov.ar, acechich@uncoma.edu.ar

Abstract

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, component-based software introduces risks, such as unknown quality properties of the components in use, that can inject harmful side effects into the final product. The purpose of a COTS selection is to analyse several candidates to identify the most suitable components to be integrated. This paper presents two of the most representative COTS selection methods, which have been used to select COTS components for an E-Payment system. The role of this case discussion is to offer some guidelines related to the use of the methods for a COTS-based development process.

Key words: COTS-based development – COTS selection – Quality attributes – E-Payment systems.

1 Introduction

The use of commercial off-the-shelf (COTS) products as elements of larger systems is becoming increasingly commonplace. Component-Based Software Development (CBSD) is focused on assembling previously existing components (COTS and other non-developmental items) into larger software systems, and migrating existing systems toward component approaches. CBSD changes the focus of software engineering from one of traditional system specification and construction to one requiring simultaneous consideration of the system context (system characteristics such as requirements, cost, schedule, operating and support environments), capabilities of products in the marketplace, and viable architectures and designs.

Within the last years both researchers and practitioners alike have established COTS quality as an important field to resolving CBS quality problems - problems ranging from COTS definition, measurement, analysis, and improvement to tools, methods and processes [5,6]. Typically, COTS evaluations consist of two phases: (1) COTS searching and screening (CS&S), and (2) COTS analysis (CA). A *COTS search* is a set of activities that attempts to identify and find all potential candidate components for reuse. The search is generally driven by guidelines and criteria previously defined. Some methods propose a separate criteria definition process while others dynamically build a synergy of requirements, goals, and criteria [1,7,8,11,12,15,16]. *COTS screening* is to decide which alternatives should be selected for more detailed evaluation. Decisions are driven by a variety of factors - foremost are several design constraints that help define the range of components. So a balance is struck, depending on the level of abstraction, complexity of the component, goals and criteria, and so forth. Some methods include "qualifying thresholds" for screening, i.e. the criteria and rationale for selecting alternatives for detailed evaluation is defined and documented. Some other methods estimate how much effort will probably be needed to actually apply all evaluation criteria to all COTS software candidates during screening.

As the COTS alternatives have been evaluated, the evaluation data need to be used for making a decision. A *COTS analysis* starts, in general, from a set of ranked COTS software alternatives where the top-ranked alternative is taken and exposed to measurement of a set of final make-or-buy decision criteria. For example, analysis can be done by using an approach called *weighted scoring method*, in which criteria are defined and each criterion is assigned a weight or score.

In this paper, we present a comparative study on selecting COTS for an e-commerce application. In this case, COTS components have been selected to provide the functionality required by the payment system. Two representative methods for selecting COTS have been used – OTSO (Off-The-Shelf Option) [8,9,10] and CAP (COTS Acquisition Process) [13,14,15] – comparing the resulting selection. In Section 2 of the paper we introduce the main characteristics of the selection methods. Section 3 then presents our case study and the selection results. Discussion is provided in Section 4. Conclusion is addressed in the final section of this paper.

2. COTS Selection Methods: OTSO and CAP

2.1 OTSO (Off-The-Shelf Option)

The OTSO method and some experiences of its use are presented in [8,9,10]. 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 the OTSO method, the evaluation criteria are gradually defined as selection process progresses, and criteria are derived from reuse goals and factors that influence these goals. The evaluation criteria definition process 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 [2,3] and Saaty's approach [17].

It is possible to identify four different sub processes in the definition of evaluation criteria: *search criteria definition*, *definition of the baseline*, *detailed evaluation criteria definition*, and *weighting of criteria*. It is enough to define the search criteria broadly so that the search is not unnecessarily limited by too many constraints. The reuse strategy and application requirements are used as the main input in the definition of these criteria. The screening process uses the criteria and determines the "qualifying thresholds", which are in deciding which alternatives are selected for closer evaluation. The definition of the baseline criteria set is essential for cost estimates and for conducting qualitative ranking of alternatives. The search criteria, however, often is not detailed enough to act as a basis for detailed technical evaluation. Therefore, the criteria will need to be refined and formalised before initiating the technical evaluation.

The *search* in the selection process attempts to identify and find all potential candidates for reuse. The search is driven by the guidelines and criteria defined in the criteria definition process. Typical information sources include in-house reuse libraries, Internet and World Wide Web, magazines and journals, vendors, colleagues, experts and consultants, etc. One main challenge in the search is the difficulty of deciding when to stop the search. A simple strategy for ending the search is to use several sources in the search, conduct the search in small increments, and review the frequency of discovering new alternatives at each increment.

The objective of the *screening process* is to decide which alternatives should be selected for more detailed evaluation. Screening is based on the same criteria that were used in the search process. In screening the "qualifying thresholds" are defined. In other words, the criteria and rationale for selecting alternatives for detailed evaluation is defined and documented. The screening process can be initiated as soon as there is at least one relevant alternative to consider. Screening is considered to be complete when evaluation alternatives selected and evaluation tasks have been assigned.

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. The evaluation criteria typically are so comprehensive that all of it may not be covered within the time available for evaluation. Therefore, the ranking of importance of evaluating each criterion should be used as a guideline in evaluation. The evaluation is completed when all alternatives have been evaluated by the defined criteria or required data have been determined not to be available.

The evaluation of alternatives in the OTSO method concentrates on producing consistent data about alternatives. As the COTS alternatives have been evaluated, the evaluation data need to be used for making a decision. A common approach for this is the *weighted scoring method* (WSM), which OTSO applies in the following fashion: criteria are defined and each criterion is assigned a weight or score. In the case of using weights, they may be normalised so that their total is one. If scoring is used, this is done, e.g., by assigning a "weight score" between one and five for each criterion. Then each alternative is given a score on each criterion. The analysis of results also relies on the use of the Analytic Hierarchy Process (AHP) for consolidating the evaluation data for decision-making purposes [17]. The AHP is based on the idea of decomposing a multiple criteria decision-making problem into a criteria hierarchy. At each level of the hierarchy the relative importance of factors is assessed by comparing them in pairs. Finally, the alternatives are compared in pairs with respect to the criteria.

2.2 CAP (COTS Acquisition Process)

The CAP method [13,14,15] consists of three components: the *CAP Initialisation Component* (CAP-IC), the *CAP Execution Component* (CAP-EC), and the *CAP Reuse Component* (CAP-RC). CAP-IC comprises all activities concerned with setting up a decision model for COTS selection and a measurement plan for assessing the COTS software alternatives. CAP-EC provides guidance for performing the assessment of the COTS software product or traditionally developing the respective parts of the software system. CAP-RC enables the storing of knowledge gained about COTS software for reuse in future COTS assessment projects.

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 a 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"). This step is needed to estimate the potential cost of applying CAP. The estimation of measurement effort is either experience-based (from historical data) or by eliciting expert knowledge.

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 straightforwardly 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.

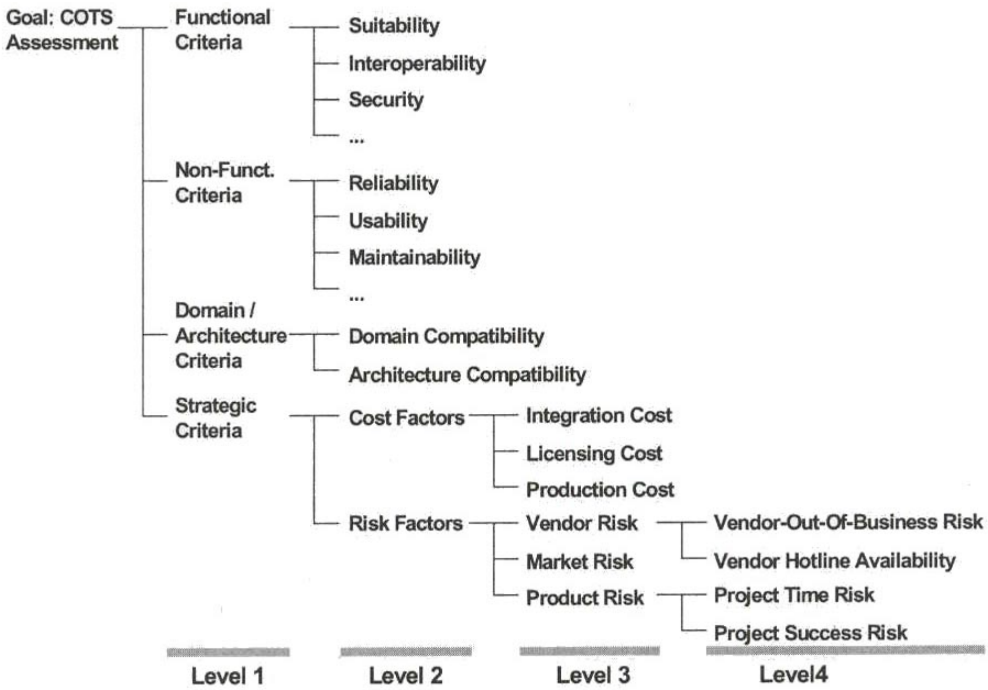


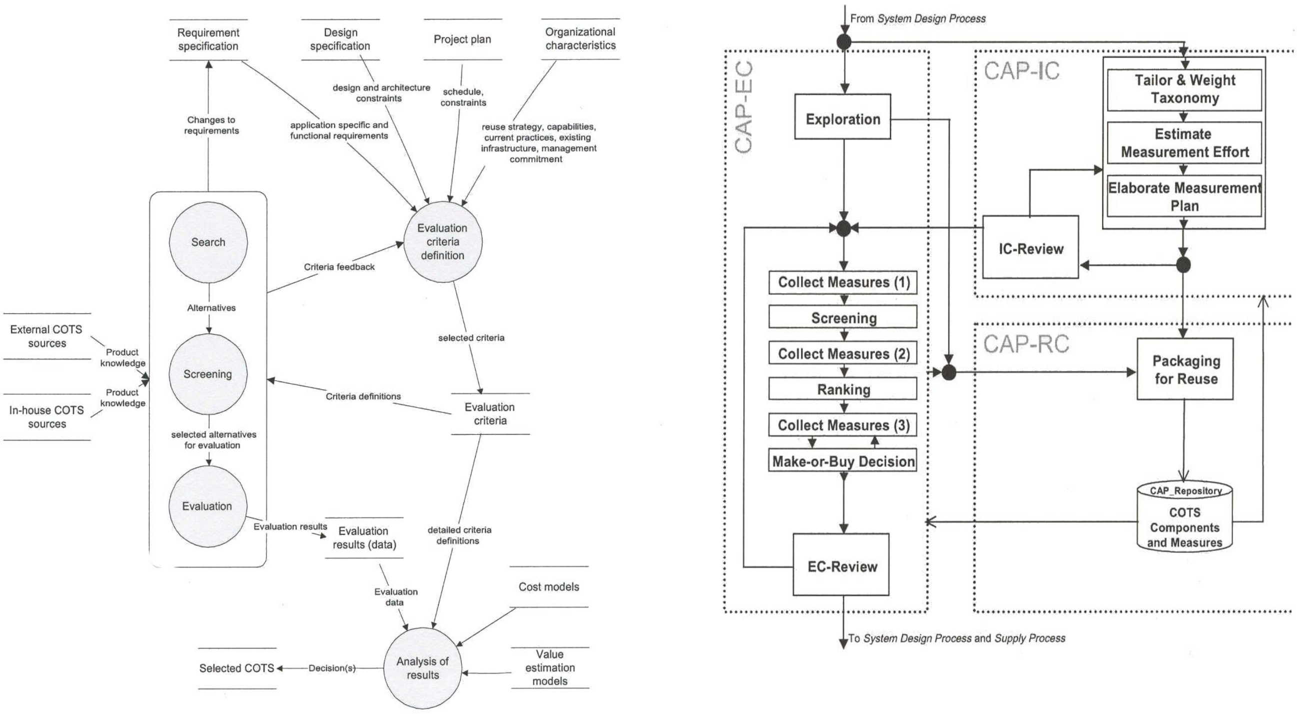
Fig. 1: CAP Taxonomy

CAP-EC basically consist of two elementary CAP activities for exploration of COTS software (CAP activity "Exploration") and for the review of all CAP-EC activities (CAP activity "EC-Review"). During the search activity, data according to the measurement plan is collected on the set of COTS software alternatives. The data gained during this first step are used in the screening activity to eliminate those COTS alternatives that are unacceptable for use. The screening is performed by applying acceptance thresholds to the measures on the COTS software alternatives. Then, data on the remaining COTS software alternatives are collected according to the measurement plan and used to establish a data-driven and priority-based ranking (best to worse) of the COTS software alternatives. For the ranking, the Analytic Hierarchy Process (AHP) is used.

From the set of ranked COTS software alternatives the top-ranked alternative is taken and exposed to measurement of a set of final make-or-buy decision criteria. Finally, the top-ranked COTS software alternative is checked for satisfying the criteria for the make-or-buy decision.

The core part of the selection performed using CAP is the CAP Evaluation Taxonomy, which consists of four levels of criteria as Figure 1 shows. At the lowest level (level 4), the taxonomy is operational by means of metrics. The evaluation taxonomy is tailored and weighted by using the Analytic Hierarchy Process (AHP). The categories and criteria are devised by stepwise subset construction from the initial set of requirements delivered as input to CAP.

Figure 2 below shows the main activities of both methods – OTSO and CAP.



(a) (b)

Fig. 2: Main activities of the OTSO method (a) and the CAP method (b)

3 A Case Study

There are several approaches to perform electronic payments, some of them can co-exist with others and some are mutually exclusive. In our case study, COTS components are limited to provide the functionality associated to credit card payment systems. Authorisation and Capture are the two main stages in the processing of a card payment over the Internet. Authorisation is the process of checking the customer's credit or debit card. The card issuer will respond to the authorisation request with one of three answers:

1. Accepted – the sale can go ahead.
2. Rejected – the card cannot be used for this transaction.
3. Invalid Data – there is some problem with the details provided and these should be checked again with the customer.

If the request is accepted, the customer's card limit is reduced temporarily by the amount of the transaction. This can be done manually or automatically. Capture is when the card is actually debited. This may take place simultaneously with the authorisation request if the retailer can guarantee a specific delivery time. Otherwise the capture will happen when the goods are shipped. Again there are both manual and automatic options available to the retailer.

This section mainly describes the selection processes carried out to find COTS components suitable of providing the main e-payment functionalities, such as authorisation and capture mentioned above. We visited the Web sites of the most popular and widely used software component vendors, including ComponentSource (www.ComponentSource.com), Flashline (www.flashline.com) and JARS (www.jars.com). From these three sites, we pre-select eight components to be evaluated: CCValidation (C1), *by FluidSoftware,Inc.*, CreditCardPack (C2), *by Keyoti,* CardVal (C3), *by GeniusTECH Solutions,* SafeCard (C4), *by Massinissa Software,* ActiveCreditCArd (C5), *Infomentun,* A$Pcharge (C6), *by Blue Squirrel*, CCProcessing (C7), *by Bash Software,* and Echo_Java_Class (C8), *by ECHO,Inc.*

3.1 Selecting COTS using OTSO

The evaluation criteria were hierarchically defined before starting the evaluation of candidates. A set of requirements for the application domain was specified to serve as a basis for establishing the criteria. It were divided into functional features, non-functional features, and managerial features. For brevity reasons, only the analysis based on functional features is presented in this paper.

Hence, Table 1 shows the evaluation data for the COTS components according to the functional criteria definition.

Criteria	C1	C2	C3	C4	C5	C6	C7	C8
Functional								
Cardholder address validation	N/A	Yes	N/A	N/A	Yes	Yes	Yes	Yes
Credit card number validation	Yes	Algorithm MOD10/Lhun	Yes	Algorithm MOD10	Combined validation	Yes	Yes	Yes
Credit card type validation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Authorisation and deposit	N/A	N/A	N/A	N/A	Yes (Gateway and CyberCash)	Different clearing services	Yes (Vital Processing Services)	Yes
Market availability	Yes	Immediately	Immediately	Immediately	Immediately	Immediately	Immediately	Yes

Table 1: Evaluation data for the COTS components according to the OTSO process

The evaluation criteria were also divided into two types: *single criteria*, which are present or absent and are evaluated by a nominal scale, and *composed criteria*, which are measured by an ordinal scale. Both types of criteria should be provided along with a relative weight for the evaluation. The possible values for weighting the criteria were defined as follows: mandatory criteria (5); highly desirable criteria (4); desirable criteria (3); low desirable criteria (2); and no desirable criteria (1). These values were used to set a weight score among the components, as Table 2 shows (column "Weight"). We also classified each component according to the degree in which a particular feature is supported. In order to do that, we define a granularity scale – from 0 to 5 – indicating the full range from "not supported" to "fully supported" features. Table 2 shows the corresponding classification for the analysed COTS components.

The following steps were followed when applying the WSM method:

1. Criteria definition
2. Weight assignment (between 1-5, where 5 corresponds to a full supported criterion)
3. Alternative weighting resulting in weighted scores.
4. Calculation of final points.

The column labelled as “Weight %” in Table 2, shows the weighted scores with respect to the total weight (86 in column “Weight”). For brevity reasons we only shows the functional criteria values, although the final row labelled as “Points” shows the total points for comparison (including all criteria).

Criteria	Weight	Weight %	C1	C2	C3	C4	C5	C6	C7	C8
Functional										
Cardholder address validation	**5**	5.81%	0	3	0	0	4	4	4	3
Credit card number validation	**5**	5.81%	3	4	3	4	4	4	4	3
Credit card type validation	**5**	5.81%	5	5	5	5	5	5	5	5
Authorisation and deposit	**4**	4.65%	0	0	0	0	3	4	4	4
Market availability	**4**	4.65%	4	4	4	4	4	4	4	4
Other criteria not included here	…	…	…	…	…	…	…	…	…	…
Points	**86**		**169**	**146**	**179**	**221**	**279**	**284**	**301**	**137**

Table 2: Calculations after applying the weighting score method

We should remark that the absolute weights only denote the order of the alternatives. In this case, the preference in decreasing order is C7, C6, C5, C4, C3, C1, C2, and C8.

The analysis of results also relies on the use of the Analytic Hierarchy Process (AHP) for consolidating the evaluation data for decision-making purposes.
In this case, the same hierarchically decomposed criteria to characterise functional features was used in both cases – WSM and AHP. An important issue in applying the AHP is that resulting values express a relative order among alternatives. If we want to get a more clear measure of the "goodness" of every component, we could try to average the scores and produce a single figure for comparison. However, this is a dangerous process. For example, we could get a high average score for a component where an assessment of a mandatory feature, such as cardholder address validation, is scored as 2. Therefore, in spite of the average score, this component shouldn't be considered a valid option to be reported for a decision-making process. As a way of minimising the impact of having mandatory features mixed with desirable or highly desirable features, we split the specific functional features proposed in Table 1 into two groups: (F1) Mandatory features, and (F2) Highly desirable features.

Table 3 shows the evaluation data after applying the AHP scoring for components C1, C2, C6 and C7.

After comparing in pairs, we propose that a component to be eligible should be assessed at least as a high score (4) in the F1 group, which must also not contain a score less than 3. Similarly, for F2 group, the average score should be at least 3 without individual score less than 2. Components weighted as 0 are discharged. The grouped average score of the components under consideration are summarised in Table 4.

C1							
Criteria						**Pair Comparison**	
Functional	**Weights**						
Address validation	3						
Number validation	5	3					
Credit card type validation		0	3			**3**	**F1**
Authorisation and deposit			0	3			
Market availability				4		**3**	**F2**
C2							
Criteria						**Pair Comparison**	
Functional	**Weights**						
Address validation	4						
Number validation	5	4					
Credit card type validation Authorisation and deposit		3	3	3		**3**	**F1**
Authorisation and deposit			0				
Market availability			4	4		**4**	**F2**
C6							
Criteria						**Pair Comparison**	
Functional	**Weights**						
Address validation	4						
Number validation	5	4					
Credit card type validation		4	4			**4**	**F1**
Authorisation and deposit			4	4			
Market availability				4		**4**	**F2**
C7							
Criteria						**Pair Comparison**	
Functional	**Weights**						
Address validation	4						
Number validation	5	4					
Credit card type validation		4	4			**4**	**F1**
Authorisation and deposit	4						
Market availability	4	4				**4**	**F2**

Table 3: AHP scoring for components C1, C2, C6, and C7

The components selected by applying the AHP method are (in decreasing order) C6, C7, and C5, which change the preference order suggested in the previous WSM method application.

COMPONENT	Mandatory (F1)	Highly Desirable (F2)	SELECTED
C1	X	X	---
C2	X	X	---
C3	X	X	---
C4	X	X	---
C5	4.33 and all ≥ 3	3.33 and all ≥ 2	✓
C6	**4.33 and all ≥3**	**3.66 and all ≥2**	✓
C7	4.33 and all ≥ 3	3.33 and all ≥2	✓
C8	X	X	---

Table 4: Selected COTS components after applying AHP in OTSO

Note that other approaches can be applied to distribute scores among several groups. For example, we could plot the distribution of the scores on frequency diagram for each feature by choosing the modal (most frequently occurring) score. If you find that scores for the same feature for the same component are distributed evenly across a number of different scale points, you need to review your subjects, your scales, and the specific environment for the reason why the subjects cannot come to a consensus view. Possible reasons include: (1) a lack of understanding of the purpose of the component, (2) misunderstanding of one or more of the evaluators about the definition or scale of a particular feature, (3) evaluations being performed for different underlying requirements, and (4) ambiguous scale points on the judgement scale, or scale points defined in too subjective a way.

3.2 Selecting COTS using CAP

During the *CAP Initialisation Component* (CAP-IC), the definition of the evaluation criteria was carried out similarly to the definition in OTSO, hence in principle, the same criteria hierarchy was used. However, by comparing both criteria hierarchies – the one used in OTSO and the one provided by CAP – we found some similarities and differences. In particular, the domain/domain compatibility characteristic of the CAP taxonomy might be easily incorporated to our taxonomy by defining a degree of satisfaction according to the provided functionality, which was assigned with a qualifying value.

Therefore, the *CAP Execution Component* (CAP-EC) was applied following the criteria taxonomy extended with the domain/domain compatibility criterion previously mentioned. Then, the criteria extension and its weights, as shown in Table 5, were used for calculation. Here, a component is weighted as 5 when the component allows to perform a credit card payment, or weighted as 0 otherwise. In that way, we reduce the amount of alternatives – only C5, C6, C7, and C8 are suitable for comparison.

Now, the absolute weights denote the following order C7, C6, C5, C8. Note that the introduction of a new criterion has particularly changed the ranking of component C8, compared to the OTSO's evaluation. It is interesting to remark that this change mostly depends on the criteria and weight definitions, showing how human-related features, such as expert knowledge and skills, affect evaluation results.

Criteria	**Weight**	**Weight %**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
Domain										
Domain Compatibility	**5**	5.49%	0	0	0	0	5	5	5	5
Total	**91**	100%								
Points			**169**	**146**	**179**	**221**	**284**	**289**	**306**	**142**

Table 5: Criterion definition for Domain/Domain Compatibility

Then, the same eight components were selected to be pondered by the AHP method. Here, just a few combinations of pairs were affected by the added criterion – domain/domain compatibility. Therefore, results are very similar to the OTSO's. The partial resulting calculations are shown in Table 6. The selected components applying the AHP method are (in decreasing order) C6, C7, and C5, which again change the preference order suggested in the previous WSM method application.

COMPONENT	Mandatory (F1)	Highly Desirable (F2)	SELECTED
C1	X	X	---
C2	X	X	---
C3	X	X	---
C4	X	X	---
C5	4.5 and all ≥ 3	3.33 and all ≥ 2	✓
C6	**4.5 and all ≥3**	**3.66 and all ≥2**	✓
C7	4.5 and all ≥ 3	3.33 and all ≥2	✓
C8	X	X	---

Table 6: Selected COTS components after applying AHP in CAP

4. Discussion

Individually, none of the COTS selection methods attempts adequately answer the question of how to identify suitable COTS providing a measure for comparison. Instead, they reflect their authors' views on how a selection should be. However, a closer study of their collections of features and requirements shows that there is a common theme among them, which is used as a guide in formulating quality attributes to be assessed.

Firstly, the analysis on the evaluation procedure reveals some similarities and important differences. Among the similarities, separated procedures to search, screen and analyse components are commonly addressed. However, it's not clearly stated what software elements (or features) are evaluated. For example, OTSO and CAP methods define a searching activity where goals, constraints, and quality features are identified. But, there is no explicit concern on providing a standard specification on those features. Furthermore, the CAP taxonomy involves aspects ranging from quality attributes based on Std. ISO 9126 to underspecified domain attributes, which cover all kind of domains. Besides the inherent complexity, criteria can be characterised by attributes such as their names and their specifications. Qualitative values for the attributes allow mandatory or optional goals to be modelled with various degrees of acceptance. In this case, both methods share an explicit activity for defining quality attributes. The variation comes from the way in which criteria are produced relying on a well-defined attribute decomposition that ends in well-defined metrics. In general, explicit activities on measurement can be found expressed as a measurement plan or a measurement assignment procedure, sometimes including quantification methods.

Secondly, it's clear that the selection is carried out from the users/acquirers point of view in both cases – and from the managers/quality staff point of view when considering decision-making activities. In spite of the actual implementation of the users/acquirers role, both methods reinforce the implications of selecting COTS on organisational roles. That is, users/acquirers might be actually a separated team in charge of assessing and selecting components, or they might be composers in charge of integrating the selected component after evaluation. These changes lead to different roles for reuse architecting, reuse management, etc. that might be involved in the selection process. The different points of view might produce different criteria definitions, which in turn might lead to different results – as we noted for the weighted case in CAP.

Thirdly, both methods focus on selecting COTS components without explicitly mentioning whether the method applies on selecting custom-built components too, which might be provided from a local repository for reuse. However, it seems that CAP might be more easily extended or adapted to cover both types of selection because of its Reuse Component (CAP-RC). The importance of differentiating these selections comes from the fact that COTS selection usually implies a more complex analysis – due to external factors such as vendor viability, requirement volatility due to versioning, etc.

As another concern, unfortunately, there is no consensus for the methods to explicitly include a supplier evaluation activity. In both cases, vendors are identified as sources of COTS components rather than as a main influencing factor when selecting COTS.

Both methods define quality characteristics and attributes, which are specific to the particular nature of COTS components and CBSD. However, the information required to evaluate COTS components using those quality models is not available in the existing commercial software repositories. For example, CCValidation presents a brief functional description along with a sample code and a demo. Supporting platforms are provided as well. However, more detailed information that helps guide evaluation against specific quality attributes – such as reliability, usability, maintainability, etc. – is not provided. In this sense, our analysis also confirms the existing gap between the *required* information by the "theoretical" metrics defined in current component quality models, and the *provided* information actually supplied by software component vendors, as the work in [4] points out.

Summing up, both COTS selection methods – CAP and OTSO – provide a general framework for evaluation, in which a common set of procedures may be generalised. These procedures – searching, screening, and analysis – might be considered as three basic levels that support more complex activities. These activities are mainly concerned with defining COTS quality attributes to be evaluated, i.e. attributes that should agree with goals and requirements previously (or dynamically) stated. 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 a COTS evaluation method 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.

Another remaining issue is trying to define a set of attributes suitable for measurement. Measures are not straightforward to get. We need not only a well-defined set of attributes but also a measuring scope, i.e. the set of attributes to be measured under a particular set of goals and a given point of view. Both methods focus on providing the first part of this statement – a set of attributes. But attributes are very differently considered when analysing COTS functionality, COTS incompatibilities, COTS viability, etc. What about how to define those quality attributes? And which documentation is clear enough, unambiguous, and complete to deal with the different COTS features? All these aspects conform a whole spectrum to select suitable COTS components for use and guidelines should be provided by a selection method to be effective.

Conclusion

The importance of discussing CBS evaluation shows up when considering that component products are developed to be generic, but integrated into a system and used in a specific context with certain dependencies.

In this paper, we have presented a comparative case on applying two well-known selection methods – OTSO and CAP. Our comparison puts the efforts in the same perspective by reviewing them accordingly to their main features. Although some activities were not considered due to constraints on our domain of study, our comparison aims at providing some insights to facilitate

COTS selection understanding. In this sense, similar comparative cases on other widely-spread COTS selection methods such as PORE[12], CEP[16], CRE[1] would be interesting to develop.

On the other hand, the existence of mismatches between the COTS product being integrated and the system is possible due to their different architectural assumptions and functional constraints. Therefore, other important aspects such as how to integrate the COTS product into a system might be also considered. In the next stage of our work, we are comparing different features of COTS selection and COTS integration methods, aiming at offering some guidelines related to the characterisation of both type of methods for a COTS-based development process.

References

1. C. Alves and A. Finkelstein. Challenges in COTS Decision-Making: A Goal-Driven Requirements Engineering Perspective. In *Proceedings of the Fourteenth International Conference on Software Engineering and Knowledge Engineering*, SEKE'02, 2002.
2. V. Basili. Software Modeling and Measurement: The Goal/Question/Metric Paradigm. *Technical Report CS-TR-2956, University of Maryland*, 1992.
3. V. Basili and H. Rombach. Tailoring the Software Process to Project Goals and Environments. In *Proceedings of the 9th International Conference on Software Engineering*, IEEE Computer Society Press, pages 345–357, 1987.
4. B. Bertoa, J. Troya, and A. Vallecillo. A Survey on the Quality Information Provided by Software Component Vendors. In *Proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE),* 2003.
5. A. Cechich and M. Piattini. Issues for Assessing Component-Based Systems. In *Proceedings of VIII Congreso Argentino en Ciencias de la Computación*, CACIC 2002, pages 253–262, 2002.
6. A. Cechich, M. Piattini, and A. Vallecillo, Assessing Component-Based Systems in *Component-Based Software Quality: Methods and Techniques*. Edited by A. Cechich et al., Springer Verlag, LNCS 2693, 2003.
7. W. Hansen. A Generic Process and Terminology for Evaluating COTS Software – The QESTA Process. http://www.sei.cmu.edu/staff/wjh/Qesta.html.
8. J. Kontio. OTSO: A Systematic Process for Reusable Software Component Selection. *Technical Report UMIACS-TR-95-63, University of Maryland*, 1995.
9. J. Kontio. A Case Study in Applying a Systematic Method for COTS Selection. In *Proceedings of the 18th International Conference on Software Engineering*, 1996.
10. J. Kontio, S. Chen, and K. Limperos. A COTS Selection Method and Experiences of its Use. In *Proceedings of the 20th Annual Software Engineering Workshop*, NASA Software Engineering Laboratory, 1995.
11. D. Kunda and L. Brooks. Applying Social-Technical Approach for COTS Selection. In *Proceedings of 4th. UKAIS Conference*, University of York, 1999.
12. N. Maiden, H. Kim, and C. Ncube. Rethinking Process Guidance for Selecting Software Components. In *Proceedings of the First International Conference on COTS-Based Software Systems*, Springer-Verlag, pages 151–164, 2002.
13. M. Ochs, D. Pfahl, G. Chrobok-Diening, and Nothhelfer-Kolb. A COTS Acquisition Process: Definition and Application Experience. *Technical Report IESE-002.00/E, Fraunhofer Institut Experimentelles Software Engineering*, 2000.
14. M. Ochs, D. Pfahl, G. Chrobok-Diening, and Nothhelfer-Kolb. A Method for Efficient Measurement-based COTS Assessment and Selection - Method Description and Evaluation Results. *Technical Report IESE 055.00/E, Fraunhofer Institut Experimentelles Software Engineering*, 2000.
15. M. Ochs, D. Pfahl, G. Chrobok-Diening, and B. Nothhelfer-Kolb. A Method for Efficient Measurement-based COTS Assessment and Selection -Method Description and Evaluation Results. In *Proceedings of the Seventh International Software Metrics Symposium*, pages 285–297, 2001.
16. S. Polen, L. Rose, and B. Phillips. Component Evaluation Process. *Technical Report SPC-98091-CMC, Software Productivity Consortium*, 1999.
17. T.L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, 1990.