BALANCING STAKEHOLDER'S PREFERENCES ON MEASURING COTS COMPONENT FUNCTIONAL SUITABILITY

Alejandra Cechich
Departamento de Ciencias de la Computación, Universidad del Comahue, Neuquén, Argentina

Mario Piattini
Escuela Superior de Informática, Universidad de Castilla-La Mancha, Ciudad Real, España

Keywords: Component-Based System Assessment, COTS components, Software Quality.

Abstract: COTS (Commercial Off-The-Shelf) components can be incorporated into other systems to help software developers to produce a new system, so that both artefacts – components and the system – form a single functional entity. In that way, developing software becomes a matter of balancing required and offered functionality between the parties. But required functionality is highly dependent on component's users, i.e. stakeholders of a COTS component selection process. Inputs to this process include discussions with composers, reuse architects, business process coordinators, and so forth. In this paper, we present an approach for balancing stakeholder's preferences, which can be used in the process of measuring functional suitability of COTS candidates. We describe and illustrate the use of our proposal to weight requirements of components and determine suitable COTS candidates for given software.

1 INTRODUCTION

The last decade marked the first real attempt to turn software development into engineering through the concepts of component-based software engineering (CBSE) and commercial Off-The-Shelf (COTS) components. It is clear that CBSE affects software quality in several ways, ranging from introducing new methods for selecting COTS components to defining a wide scope of testing principles and measurements (Cechich et al, 2003).

The idea is to create high-quality parts and put them together. However, joining high quality parts not necessarily produces a high-quality system.

At the same time, defining quality features able to be measured might mitigate the impact of selecting and integrating COTS components. Although measures are not straightforward to take, it might be possible to focus on different aspects of a component, which indirectly – or perhaps directly in some cases – provide metrics on the resulting composition. In that way, metrics might be used to improve the process of selecting and integrating components by reducing risks on decision-making tasks (Sedigh-Ali et al., 2001).

Components are plugged into a software architecture that connects participating components and enforces interaction rules. The architectural options are high-level descriptions of components and their expected interactions. For instance, the model in Alexander and Blackburn (1999), supposes that there is an architectural definition of a system, whose behaviour has been depicted by scenarios or using an architecture description language (ADL). The model explores the evaluation of components using a specification-based testing strategy, and proposes a semantics distance measure that might be used as the basis for selecting a component from a set of candidates.

Our proposal (Cechich and Piattini, 2004), has adapted this model as a basement for quality measurement. We express the semantics distance in terms of functional suitability measures, which provide a better identification of the different COTS functionalities. To do so, a system can be extended or instantiated through the use of some component type. Due several instantiations might occur, an

Cechich A. and Piattini M. (2004).
BALANCING STAKEHOLDER'S PREFERENCES ON MEASURING COTS COMPONENT FUNCTIONAL SUITABILITY.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 115-122
DOI: 10.5220/0002607201150122

assumption is made about what characteristics the actual components must possess from the architecture's perspective. Thus, the specification of the architecture *A* (*SA)* defines a specification S_C for the abstract component type *C* (i.e. $SA \Rightarrow S_C$). Any component K_i, that is a concrete instance of *C*, must conform to the interface and behaviour specified by S_C.

We should remark the importance of determining behavioural incompatibilities through the use of scenario specifications, even thought the scenario *S* is not explicitly included into our measure definitions. This is due to the fact that we consider the definition of metrics as a process included into a broader measurement process, which defines some activities for setting the measurement context – such as defining scenario specifications or identifying stakeholders (Cechich and Piattini, 2003).

Determining the needed quality is difficult when different stakeholders have different needs. One might be tempted to state that the stakeholder requiring the highest component quality should determine the overall level of quality to the CBS. But what if that component use is rather minor and unimportant to the CBS, whereas the major component use does not require anywhere near such a level of quality? Thus it is necessary to balance conflicting requirements for CBS quality.

Weighting requirements for COTS component selection can be problematic. Sometimes these weights are inconsistent and lead to confusion about which are the most essential customer requirements (Maiden and Ncube, 1998). Using more sophisticated methods, such as the AHP method (Saaty, 1990), has received some interest in the application of well-known COTS component selection procedures. However, simpler decision-making techniques can also be appropriate to resolve disagreements promoting a cost-effective use. In any case, clearly defining a way of balancing preferences on requirements is essential to the selection process.

On the other hand, the requirements elicitation techniques have widely used a family of goal-oriented requirements analysis (GORA) methods – I* (I* homepage; Mylopoulos et al., 1999), KAOS (KAOS homepage; Dardenne et al., 1993), and GRL (GRL homepage) – as an approach to refine and decomposing the needs of customers into more concrete goals that should be achieved.

In this paper, we describe a proposal to balance stakeholder's preferences during a COTS component selection process. Our proposal extends a version of a Goal-Oriented Requirements Analysis Method called AGORA (Kaiya et al., 2002) by considering additional features of COTS components. The proposal might be combined with other techniques for weighting preferences such as the Goals-Skills-Preferences Framework (Hui et al., 2003) and the AHP method. Then, the balanced requirements are included into the computation of a compact suite of measures on functional suitability of the COTS component candidates.

In Section 2 of the paper we briefly introduce our measurement approach for COTS component's functional suitability. Section 3 then presents the notion of AGORA graphs as it might be used in COTS component selection to balance stakeholder's preferences. Finally, section 4 introduces our weighting procedure to functional suitability based on measures derived from the graph. We conclude with an overview of research directions and future extensions.

2 MEASUREMENT OF COTS FUNCTIONAL SUITABILITY

In the previous section, we have emphasized the fact that the output from the system should satisfy the user's requirements by using the functionality supplied by at least one COTS component. They are plugged into a software architecture that connects participating components and enforces interaction rules.

Given a specification S_C for an abstract component type *C*, a candidate component *K* to be a concrete instance of *C* must conform to the interface and behaviour specified by S_C.

Although the process of selecting a component *K* consists of evaluating interface and semantic mappings, in our work only semantic mappings are addressed. Mappings in S_C, which represent the different required functionalities, are established between input and output domains. We focus on incompatibilities derived from functional differences between the specification in terms of mappings of a component K_i (S_{Ki}) and the specification in terms of mappings of S_C.

Let's illustrate the measurement procedure by using an E-payment system as an example. We suppose the existence of some scenarios describing the two main stages of the system – *authorisation* and *capture.* Authorisation is the process of checking the customer's credit card. If the request is accepted, the customer's card limit is reduced temporarily by the amount of the transaction. Capture is when the card is actually debited.

The scenarios will provide an abstract specification of the mappings of S_C that might be composed of:

- Input domain: (AID) Auth_IData{#Card, Cardholder_Name, Exp-Date}; (CID) Capture_Idata{Bank_Acc, Amount}.
- Output domain: (AOD) Auth_Odata{ok-Auth}; (COD) Capture_Odata{ok_capture, DB_update}.
- Mapping: {AID $\rightarrow$ AOD};{CID $\rightarrow$ COD}

Suppose we pre-select two components to be evaluated, namely K_1 and K_2 respectively. A typical situation for inconsistency in the functional mappings between S_{K1}, S_{K2} and S_C is illustrated in Figure 1, where dashed lines indicate (required) mappings with respect to S_C, and the solid lines are (offered) mappings with respect to S_{K1} (grey) and S_{K2} (black). Note that the input domain of the component K_1 does not include all the values that the specification S_C requires, i.e. the capture functionality is not provided. Besides, the input domain of the component K_2 includes more values than the required by S_C, although the mapping satisfies the required functionality. We should also note that there is another functionality provided by K_2, i.e. {Taxes $\rightarrow$ Statistics}, which might inject harmful effects to the final composition.

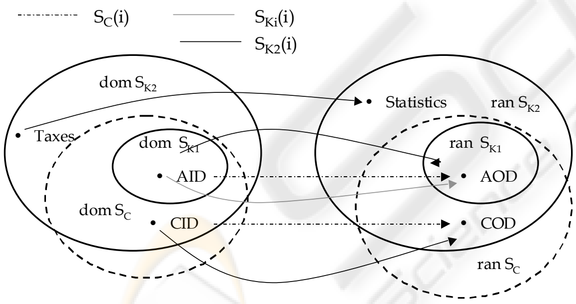

Figure 1: Functional mappings of S_C and S_{K1}/S_{K2}

Our measures of functional suitability have been classified into two different groups: component-level measures and solution-level measures. The first group of measures aims at detecting incompatibilities on a particular component K, which is a candidate to be analysed. However, it could be the case that we need to incorporate more than one component to satisfy the functionality required by the abstract specification S_C. In this case, the second group of measures evaluates the functional suitability of all components that constitute the candidate solution.

To clarify the use of the specification S_C during the measurement procedure, we briefly introduce some metrics. At the component-level, we have defined the following measures (Cechich and Piattini, 2004):

- "Compatible Functionality" (CF_C) as the amount of functional mappings provided by S_K and required by S_C in the scenario S.
- "Missed Functionality" (MF_C) as the amount of functional mappings required by S_C in the scenario S and not provided by S_K.
- "Added Functionality" (AF_C) as the amount of functional mappings not required by S_C in the scenario S and provided by S_K.
- "Component Contribution"(CC_F) as the percentage in which a component contributes to get the functionality required by S_C in the scenario S.

Now, let's calculate the functional suitability measures on K_2 for the E-payment example. Considering the functional mappings provided by K_2 ({AID $\rightarrow$ AOD; CID $\rightarrow$ COD; Taxes $\rightarrow$ Statistics}), the component-level measure results are as follows:

$CF_C(K_2) = 2$; $MF_C(K_2) = 0$, $AF_C(K_2) = 1$;
$CC_F(K_2) = 1$.

These values indicate that the component K_2 is a candidate to be accepted for more evaluation; i.e. the component is completely functionally suitable. But there is one added function that could inject harmful side effects into the final composition. Besides, there are another types of analysis the component should be exposed before being eligible as a solution – such as analysis of non-functional properties (Chung et al., 2000), analysis of vendor viability (Ballurio et al., 2002), and so forth.

Adaptation required by the components should also be quantified. For example, measurement might be defined at three levels: (1) size measures will be basically in terms of the amount of adaptability needed by a component-based solution; (2) complexity of adaptation will be measured in terms of interactions with target components that are identified to determine all potential mismatches; and finally, (3) architectural adaptability might define calculations for measures of changes that affect system's stability (Cechich and Piattini, 2003) in terms of architectural adaptability.

3 STAKEHOLDER'S PREFERENCES ON COTS FUNCTIONALITY

Stakeholders might try to find the best component (or set of components) decomposing and weighting the goals of the abstract specification S_C, as it was presented in the previous section. For example, a *reuse architect* may be interested in identifying and acquiring components promoting the value of reuse and ensuring consistency of design across projects; a *certifier* may be interested in setting component specification standards and ensuring compliance and consistency of components across different teams; or a *business process coordinator* may be interested in demonstrating the value of components with respect to business processes (Allen and Frost, 2001). Hence, functional requirements are affected by different views that should be conciliated.

Generally speaking, goals can be decomposed to calculate a preference value for each stakeholder. The extended version of a Goal-Oriented Requirements Analysis Method called AGORA, (Kaiya et al., 2002), is a top-down approach for refining and decomposing the needs of customers into more concrete goals that should be achieved for satisfying the customer's needs.

An AGORA goal graph, is an attributed version of AND-OR goal graphs, whose parts can be described as follows:

- Attribute values are attached to nodes and edges, in addition to structural characteristics of the graph. There are two types of attributes:
    - A preference matrix is attached to a node, i.e. a goal, and stands for the degree of preference or satisfiability of the goal for each stakeholder; and
    - A contribution value is attached to an edge to express the degree of the contribution of the goal to the achievement of its connected parent goal.
- Rationale can be attached to an attribute as well as a node and an edge. It represents decomposition decisions associated to goal refinement and attribute value definition.

The procedure to construct an AGORA goal graph involves:

- Establishing initial goals as customers' needs.
- Decomposing and refining goals into sub-goals.
- Choosing and adopting the goals from the alternatives of decomposed goals.
- Detecting and resolving conflicts on goals.

The stakeholders attach the value subjectively. However, they can use some systematic techniques, such as the AHP method (Saaty, 1990), to assign more objective values.

The contribution values and preference matrices help to choose suitable sub-goals. Basically, when a sub-goal is connected to an edge having a high contribution, it can be a candidate to be chosen as a successor of his parent goal

Since a preference matrix includes the preference degree for each stakeholder, we can identify the conflicts among them by analysing the variance on the diagonal elements of the matrix.

In the following subsection, we introduce an extension of the AGORA graph to include some necessary elements when evaluating COTS components. Our approach explicitly considers some characteristics of COTS components to balance stakeholder's preferences on functional goals.

3.1 AGORA Graphs in COTS Selection

Initial goals are typically considered as the customer needs and assigned to nodes of the graph. But when incorporating COTS components, goals should be balanced against COTS services. For example, using the main concepts of goal-oriented requirements engineering, the goal acquisition and specification process (Alves, 2003; Alves and Finkelnstein, 2002) includes the necessity of identify goals that help to distinguish between products (called core goals) from those that are provided by most available products (called peripheral goals). Then, our proposal extends the AGORA graph to include a first categorisation of goals into *core* and *peripheral.*

A second categorisation is due to the traditional separation of requirements into functional and non-functional properties. This classification remains relevant due to the different treatments given to the properties when defining quality attributes and measurements. In this work, only functional suitability is considered. Then, we limit the scope of this paper to analysing functional properties.

Initial goals, as considered in AGORA graphs, are the needs of the customers that will be refined and decomposed into sub-goals one after another. It is possible to have more than one sub-goal of a parent goal, and it is also possible to use two types of decomposition corresponding to the logical combination of the sub-goals – one is AND-decomposition and the other is OR-decomposition.

Therefore, with the functional goals of the component specified by mappings in S_C, the next step is to refine the goals considering the perspectives of different stakeholders – reuse architect, certifier, business process coordinator, etc. Then, the computation of stakeholder's preference values for the refined goals will allow us to add preferences to mappings of S_C, distinguishing between core and peripheral functional goals.

In our context of component-based systems, an AGORA graph describes the abstract specification of a required component (S_C) according to the scenario *S*. Figure 2 shows a snapshot of a possible AGORA graph for our E-payment case.

There are two types of conflicts on goals; one is the conflict between goals and the other one is the conflict on a goal between stakeholders. The first type of conflicts appears in Figure 2 between the goals "Prevent unauthorised debits" and "No authorisation", whose edge has a negative contribution value. The second type appears in the figure on the goal "Input AID data". The diagonal elements of the preference matrix show that the reuse architect gave the preference value 8 by himself, while the business process coordinator's preference is –5 given by himself (Figure 3 shows the preference matrix where the stakeholders are identified).

When we find a large variance on the diagonal elements of the preference matrix, there is a possibility of conflict among the stakeholders for the goal. In this case, the relevant stakeholders would be forced to negotiate for the conflict resolution of the goal. Negotiation can be supported by methods such as the WinWin (Boehm et al., 1995).

In Figure 2, final goals are identified to achieve the initial goals, i.e. the sub-goals "Input AID data", "Issue AOD data", "Register formally with the bank", and "Input the user identification by Web page by user".

It seems that information to apply a traditional negotiation process is enough. But, we have already remarked the importance of distinguishing between core and peripheral goals when selecting COTS components. This characterisation would lead to dealing with a third type of conflicts on goals: conflicts between the abstract specification of a component and its possible instantiations. These conflicts should be resolved when the COTS component selection actually take place. Then, it is important to add some extra information on the graph, so the negotiation will be possible.

Then, the AGORA graph in Figure 2 has been extended adding the labels <core>/<peripheral> to facilitate the future negotiation. For example, suppose that we evaluate the components K_1 and K_2 introduced in section 2. We easily note that component K_1 should be withdrawn from analysis because it does not offer one core functionality. We should search for other components or combination of components, such as K_2, to instantiate the three core goals of the graph. On the other hand, there is a peripheral goal ("Input the user identification by Web page by user") on the graph, which would be desirable to have. However, its categorisation as peripheral makes this functionality a candidate to be discharged (or to be added by an adapter), when there are no COTS candidates offering it.

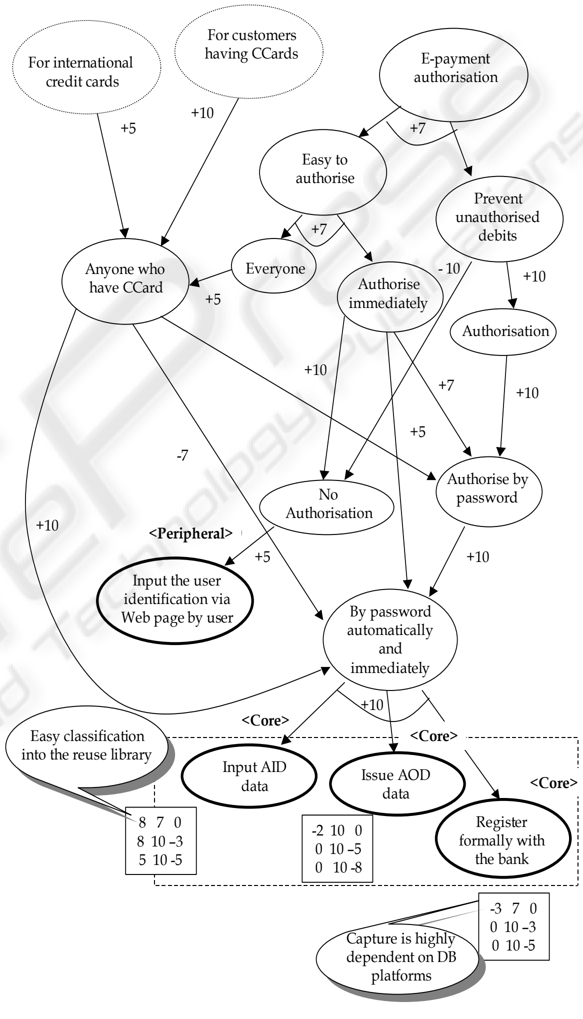

Figure 2: A snapshot of the AGORA graph for the E-payment case

	RA	CE	BC
RA	8	7	0
CE	8	10	–3
BC	5	10	-5

RA: Reuse Architect
CE: Certifier
BC: Business Process Coordinator

Figure 3: An example of a preference matrix

4 MEASURING DESIRABILITY AND MODIFIABILITY OF GOALS

Besides the classification of goals as core and peripheral, the attributes desirability (level of importance for a goal to be met), and modifiability (level in which a goal can be modified) are proposed as attributes for goal description when selecting COTS components (Alves and Filkenstein, 2002).

By using an AGORA graph, we can estimate the quality of several properties of the adopted goals. Particularly, correctness is assumed as a quality factor that represents how many goals in a specification meet stakeholder's needs. Correctness in AGORA is strongly related to contribution values on the path of the adopted goal as well as on its stakeholder's preference value. Particularly, the average stakeholder's preference value of the adopted final goals (*Cup*) is defined by Kaiya et al. (2002) as:

$$Cup = \text{AVE}\left(\bigcup_{f \in FinalGoal,\ s \in Stakeholder,\ m \in Preference} \{m_{s,customer} \mid has(f,m)\}\right)$$

where $m_{s,customer}$ means a stakeholder's preference value evaluated by the stakeholder s in the preference matrix m. The results of the calculation for all the core goals of Figure 2 are as follows:

$Cup(RA) = ((8 + 8 + 5) + (-2 + 0 + 0) + (-3 + 0 + 0)) / (3 + 3 + 3) / 10 = 0.18$

$Cup(CE) = ((7 + 10 + 10) + (10 + 10 + 10) + (7 + 10 + 10)) / (3 + 3 + 3) / 10 = 0.93$

$Cup(BC) = ((0 - 3 - 5) + (0 - 5 - 8) + (0 - 3 - 5)) / (3 + 3 + 3) / 10 = -0.32$

$Cup = (0.18 + 0.93 - 0.32) / 3 = 0.26$

In COTS component selection, this measure might indicate the degree of agreement on stakeholder's preferences, i.e. on the *desirability* of the core goals of the abstract specification S_C. Lower results of *Cup*, such as 26% in our case, show a need of further discussion on the required functionality of the component *C*; i.e. causes of disagreement should be detected. For example, stakeholders have different goals, even their perceptions of reality vary significantly. Then, scenarios may drive the agreement process and establish partial consistency among existing systems – all systems involved in using the COTS component.

On the other hand, *modifiability* is about the degree in which committed goals can be changed when selecting COTS components. Let's briefly clarify the point: suppose there is a strong agreement on a set of goals (*Cup* = 80%), however the search of COTS candidates offering the functionalities shows that there are no candidates available. In this case, evaluators should have agreed on the degree in which the goals (even categorised as core) can be modified. Then, the modifiability of the goals will help to decide on acquiring COTS components with less functionality than required, adding the functionality by means of an adapter (such as a wrapper), or building the missed functionality from scratch.

In (Kaiya et al, 2002), the quality metrics for modifiability include how an AND-OR graph is closed to a tree structure. When there are many incoming edges to a goal, the goal contributes to an achievement of many goals. In consequence, these many goals should be under consideration in case of changing the goal. The quality metric is defined as follows:

$$Tre = \frac{\#\{g \in RefinedGoals \mid \#\{e \mid incoming(g,e)=1\}\}}{\#RefinedGoals}$$

$$RefinedGoals = Goals - Initial\ Goals$$

Calculations for Figure 2 show that there are 3 initial goals and 13 refined goals, from which only 9 have one incoming edge. Then, the result of the calculation of *Tre (modifiability)* for Figure 2 is 9 / 13 = 0.69. In other words, the figure shows four goals whose incoming edges are more than one (13 – 4 = 9), out of 13 refined goals.

We should note that other quality metrics such as unambiguity, completeness, and consistency might be calculated on AGORA graphs. However, desirability and modifiability are the main properties when we apply the analysis on abstract specifications of COTS components aiming at being included into a selection procedure.

4.1 Weighting the functional requirements of *Sc*

Functional mappings of S_C, as introduced in section 2, are associated to one or more refined goals of the graph. By doing so, an agreement among stakeholders might be achieved by calculating the desirability of each group of refined goals representing a particular mapping. For the example in Figure 2, calculations should be split into three groups: one containing the core goals referring to the authorisation functionality (two core refined goals), another containing the capture functionality (one core refined goal), and another containing the peripheral goal. Then, desirability of the three groups should be calculated and decisions should be made based on the following cases, where "agreement-threshold" is a suggested value between 0.6 and 0.7 and "core/peripheral" is the type of refined goal:

- Case 1: desirability(refined goal/s) < agreement-threshold ∧ core ⇒ Try other scenarios to get agreement
- Case 2: desirability(refined goal/s) < agreement-threshold ∧ peripheral ⇒ decision to discharge
- Case 3: desirability(refined goal/s) ≥ agreement-threshold ∧ peripheral ⇒ decision to retain
- Case 4: desirability(refined goal/s) ≥ agreement-threshold ∧ core ⇒ keep for the selection process

Modifiability is calculated to be used during the selection procedure, whether decisions on buying or developing should be made.

Let's consider again the E-payment example. The functional suitability measures introduced in section 2 were calculated for two COTS candidates – K_1 and K_2. Before starting the selection procedure, an abstract specification of the component (S_C) was defined as a reference to be used when comparing candidates, and the component K_2 was indicated as the most functionality suitable. However, the agreement on the functionality required by the abstract specification S_C is not enough (*Cup* indicator is around 26%). It would indicate that we should have tried other scenarios to get agreement (case 1). Hence, the desirability measure might have been used to avoid further investment in searching candidates while required functionality is still not clear, i.e. we should not proceed comparing the functionality offered by candidates until desirability of all core requirements has reached the agreement-threshold.

Of course, actually classifying the requirements as "core" or "peripheral" is another different concern. We assume that the classification is valid and it remains stable during the selection process. However, 69% of modifiability would indicate that there is a good chance of negotiating (and changing) the requirements when balancing between offered services of candidates. But it also could indicate that the classification as "core" should be reviewed. Having higher values on modifiability (around 90%) on a core requirement would indicate that we could potentially resign most of our expectations on this requirement letting offered services prevail. For example, we could keep some of the alternative goals resigning others whether COTS candidates are hard to find or adapt.

Summing up, desirability might reduce search and selection efforts by detecting functionality on which there is no enough agreement; and modifiability might help to predict a space of negotiation and change when constraints from actual candidates are applied.

CONCLUSIONS

Assessing component-based systems, especially systems with COTS components, differs in several ways from the usual situation. Now, stakeholders must be willing to resign some of their requirements trying to adjust their expectations to what actual candidates offer in a marketplace. In this context, balancing requirements among offerings is an outstanding concern when selecting COTS components. In this paper, we have introduced a proposal for calculating a preference value for each functional requirement, and we have weighted the desirability and modifiability of core and peripheral goals. These calculations are some of the inputs required by a broader measurement procedure, which would lead to a more objective evaluation of COTS candidates.

However, an aspect that needs further discussion is the possibility of establishing a set of main stakeholders or roles on a selection process. Furthermore, when an organisation implements the approach, it needs to identify which specific roles and priorities it should address and what does or does not work for that organisation. Preference matrices should be limited to these specific roles, so calculations are kept between practical and meaningful boundaries.

Another aspect that needs more attention is the diverse possibilities of documenting the specification S_C. We have chosen scenarios because of their wide use on evaluating architectures, however other representations might be more suitable depending on specific constraints of the system.

Finally, the classification of requirements as core or peripheral needs to be derived from a previous analysis on factors that traditionally influence software requirements elicitation processes.

ACKNOWLEDGMENTS

This work was partially supported by the CyTED (Ciencia y Tecnología para el Desarrollo) project VII-J-RITOS2, by the UNComa project 04/E048, and by the MAS project supported by the Dirección General de Investigación of the Ministerio de Ciencia y Tecnología (TIC 2003-02737-C02-02).

REFERENCES

Alexander R. and Blackburn M., 1999. Component Assessment Using Specification-Based Analysis and Testing. *Technical Report SPC-98095-CMC, Software Productivity Consortium*.

Allen P. and Frost S., 2001. Planning Team Roles for CBD. In *Component-Based Software Engineering - Putting the Pieces Together*, Addison-Wesley. Edited by G. Heineman and W. Council.

Alves C., 2003. COTS-Based Requirements Engineering. In *Component-Based Software Quality: Methods and Techniques*, Springer-Verlag LNCS 2693.

Alves C. and Filkenstein A., 2002. 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.

Ballurio K., Scalzo B., and Rose L, 2002. Risk Reduction in COTS Software Selection with BASIS. In *Proceedings of the First International Conference on COTS-Based Software Systems*, ICCBSS 2002, Springer-Verlag LNCS 2255 , pp. 31-43.

Boehm B., Bose P., Horowitz E., and Lee M., 1995. Software Requirements Negotiation Aids: A Theory-W Based Spiral Approach. In *Proceedings of the 17th International Conference on Software Engineering*, pp. 243-253.

Cechich A., Piattini M., and Vallecillo A., (eds.) 2003. *Component-Based Software Quality: Methods and Techniques*, Springer-Verlag LNCS 2693.

Cechich A. and Piattini M., 2003. Defining Stability for Component Integration Assessment. In *Proceedings of the 5th International Conference on Enterprise Information Systems*, ICEIS 2003, pp. 251-256.

Cechich A. and Piattini M., 2004. On the Measurement of COTS Functional Suitability. In *Proceedings of the 3rd International Conference on COTS-based Software Systems*, ICCBSS 2004, Springer-Verlag LNCS.

Chung L., Nixon B., Yu E., and Mylopoulos J., 2000. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publisher.

Dardenne A., van Lamsweerde A., and Fickas S, 1993. Goal-directed Requirements Acquisition. *Science of Computer Programming*, Vol. 20, pp. 3-50.

GRL homepage, http://www.cs.toronto.edu/k-m/GRL/

Hui B., Lisakos S., and Mylopoulos J., 2003. Requirements Analysis for Customizable Software: A Goals-Skills-Preferences Framework, In *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pp. 117-126.

I* homepage, http://www.cs.toronto.edu/km/istar

Kaiya H., Horai H., and Saeki M., 2002. AGORA: Attributed Goal-Oriented Requirements Analysis Method. In *Proceedings of the IEEE International Conference on Requirements Engineering*, pp. 13-22.

KAOS homepage, http://www.info.ucl.ac.be/research/projects/AVL/ReqEng.html

Maiden N. and Ncube C., 1998. Acquiring COTS Software Selection Requirements. In *IEEE Software*, Vol. 15(2), pp. 46-56.

Mylopoulos J., Chung L., and Yu E., 1999. From Object-Oriented to Goal-Requirements Analysis, *Communications of the ACM*, *42*(1), pp. 31-37.

Saaty T.L., 1990. *The Analytic Hierarchy Process*. McGraw-Hill.

Sedigh-Ali S., Ghafoor A., and Paul R., 2001. Software Engineering Metrics for COTS-Based Systems. *IEEE Computer Magazine*, pp. 44–50.