Driving the selection of cots components on the basis of system requirements

Conference Paper · January 2007

DOI: 10.1145/1321631.1321697 · Source: DBLP

4 authors:

Vittorio Cortellessa
Università degli Studi dell'Aquila

Ivica Crnkovic
Chalmers University of Technology

Fabrizio Marinelli
Università Politecnica delle Marche

Pasqualina Potena
Swedish Institute of Computer Science

Driving the Selection of COTS Components on the Basis of System Requirements

Vittorio Cortellessa
Dipartimento di Informatica
Università dell'Aquila
Via Vetoio, 1, Coppito (AQ),
67010 Italy
cortelle@di.univaq.it

Ivica Crnkovic
Department of Computer
Science and Electronics
Mälardalen University
PO Box 883, SE-721 23
Västerås, Sweden
ivica.crnkovic@mdh.se

Fabrizio Marinelli
Laboratoire d'Informatique de
l'Ecole Polytechnique
Paris(France)
marinelli@lix.polytechnique.fr

Pasqualina Potena
Dipartimento di Scienze
Università "G.D'Annunzio"
Viale Pindaro, 42, Pescara,
65127 Italy
potena@sci.unich.it

ABSTRACT

In a component-based development process the selection of components is an activity that takes place over multiple lifecycle phases that span from requirement specifications through design to implementation-integration. Automated tool support for component selection would be very helpful in each phase. In this paper we introduce a framework that supports the selection of COTS components in the requirements phase. The framework lays on a tool that builds and solves an optimization model, whose solution provides the optimal COTS component selection. The selection criterion is based on cost minimization of the whole system while assuring a certain degree of satisfaction of the system requirements. The output of the model solution indicates the optimal combination of single COTS components and assemblies of COTS that satisfy the requirements while minimizing costs.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications—*tools*.

General Terms

Design.

Keywords

COTS selection, optimization model, software requirements.

ASE'07, November 5–9, 2007, Atlanta, Georgia, USA.

1. INTRODUCTION

In a component-based development (CBD) process, one of the crucial (and specific) activities is component selection. The component selection is an activity in which components are assessed in order to estimate appropriateness of their inclusion into the system. A selection process may result in selecting several candidates or a particular component, or in finding that some desired functionality cannot be found in existing components and the functionality should be implemented. The component selection is not performed only in the implementation phase, in which a selected component is supposed to replace a possible implementation of software resulting from the design phase. If the component selection would be limited to the implementation phase, it would be difficult to find pre-existing components (in particular COTS components which we are interested to) that exactly match the design specifications. For this reason the component selection must be considered earlier in the development process. This activity in practice starts already in the requirements elicitation phase and then being refined in the development process [6].

One of the main objectives of each phase is to select "the best" set of (COTS and in-house) components, but with slight difference of goals. In the requirements phase, the goal is to find appropriate candidate that might be integrated into the system. In the design and implementation phase, particular components are selected and verified/tested in isolation and in combination with other (selected) components. A combination that best fits the goals will be selected. In the maintenance phase there will be requirements to replace a particular component (or set of components) while keeping other components unchanged. In all phases the choice may have significant consequences of the development costs and the final product quality. Therefore the tools that support decisions strictly related to meet functional and non-functional requirements, while keeping the costs within a predicted budget, would be very helpful to the software developers' tasks [5].

For the design phase, in [5] CODER (Cost Optimization under DElivery and Reliability constraints) has been presented, that is a framework based on an optimization model that supports "build-or-buy" decisions in selecting components. Getting as input value the software architecture and the possible scenarios of the system, CODER suggests the best selection for each component of the system, either one of the available COTS products or an in-house developed one. The selection is based on minimizing the costs of construction of the system under constraints on delivery time and reliability of the whole system. In addition, for each in-house developed component CODER suggests the amount of testing to perform in order to achieve the required level of reliability.

All the above considerations can be consistently applied to in-house developed and to COTS components. However, the focus of our work is on COTS components, therefore in the rest of the paper we will refer only to the latter. In particular, we work on the selection of COTS components in the requirements phase, and for this goal we introduce the DEER (DEcision support for componEnt-based softwaRe) framework. Basing on an optimization model, it helps to check if the requirements, that do not depend only on the architecture of the system [18], are satisfied by a COTS component or an assembly of components ([1]).

DEER supports the component selection based on cost and satisfaction of functional and non-functional requirements of the system. We assume that each functional requirement is satisfied by a set of COTS components or assemblies. Basically, the COTS components and the assemblies differ with respect to the cost and eventually to the degree of satisfaction provided for the non-functional requirements. DEER provides a solution that determines the COTS components to choose in order to minimize the costs of construction of the system while satisfying (at a certain extent) the functional and non-functional requirements.

The paper is organized as follows: in Section 2 we provide the formulation of the optimization model that represents the DEER core; in Section 3 we introduce related work and we discuss the novelty of our contribution; conclusions are presented in Section 4.

2. THE OPTIMIZATION MODEL FORMULATION

In this section we introduce the mathematical formulation of the optimization model that represents the core of the DEER tool.

2.1 A premise on the requirement satisfaction

In order to quantify the degree of satisfaction of a requirement by a COTS component it is necessary to adopt a methodology of comparison. In [1], using the outlines of the *goal-oriented requirements engineering* [11] that defines requirements as goals, the authors present a goal-based approach to match the goals of the system and the features of the COTS components. They provide a methodology to derive the *operational goals* that represent the requirements for which it is possible to quantify the degree of satisfaction required. Assuming that an *operational goal* expresses a condition over a quality factor, they define the *satisfaction function* of the *operational goal* g with respect to its quality factor q as follows:

$$Sat_g : M \rightarrow [0, 1] \tag{1}$$

where M is the set of values that the quality factor q of g may take. Besides, they quantify how a COTS component satisfies an *operational goal* in the following way. Let x be an element of the set M and $Sat_g(x)$ be the degree of satisfaction of the goal g whit respect to x. Let C be a COTS component available in the market. Upon determining the value of the quality factor q of g in C, represented by C_q, and using the satisfaction function of the goal g, they define the satisfaction degree that the COTS component C meets with the goal g as $Sat_g(C_q)$.

2.2 The Problem Formulation

Let $G = \{g_1, \ldots, g_n\}$ be a set of n *operational goals*, each of which represents either a functional or a non-functional requirement of the system.

Let $C = \{C_1, \ldots, C_{|C|}\}$ be a set of COTS components available on the market.

Let $\mathcal{A} = \{A_1, \ldots, A_{|\mathcal{A}|}\}$ be a family of assemblies of COTS components. Each element A_j of $\mathcal{A}$ is a set consisting of one or more COTS components which satisfy at least a functional requirements g_k of the system. Notice that although the cardinality of A can grow exponentially in the number of COTS components, the set $\mathcal{A}$ describes the COTS assemblies available in the market and therefore it consists of few elements in practice.

2.2.1 Model Parameters.

Let c_i be the purchase cost, typically provided by the vendors, of the i-th COTS component.

Let $\bar{c}_j$ be the cost of the j-th assembly, i.e., the cost of integration and adaptation needed to build the assembly A_j.

The estimate of the cost $\bar{c}_j$ is outside the scope of this paper, however, the adaptation cost c^{adapt} could be estimated in the following way. Consider the assembly A_j ($1 \leq j \leq |\mathcal{A}|$) and the *operational goal* g_k ($1 \leq k \leq n$). The cost of adaptation of A_j in order to satisfy g_k can be estimated by using the following expression:

$$c^{adapt} = \max(0, Sat_{g_k} - Sat^j_{g_k}) \cdot \bar{A}_j \tag{2}$$

where Sat_{g_k} represents the degree of satisfaction required for the goal g_k, $Sat^j_{g_k}$ represents the satisfaction degree with which the assembly A_j meets with the *operational goal* g_k and $\bar{A}_j$ is an integer number that depend on the features of the assembly A_j with respect, for example, to the values of the development process parameters (e.g. experience and skills of the developing team). To estimate Sat_{g_k} and $Sat^j_{g_k}$ the approach presented in [1] can be adopted.

Besides, the cost of adaptation of A_j could be estimated by considering other factors, such as the cost needed to resolve inconsistencies [16] between the specifications of the COTS components that belong to A_j.

With regard to the integration cost, the components that belong to an assembly must support the style of interaction of the assembly architecture. If a COTS component has a

[1]An assembly is defined as a set of COTS components that interact with each other."Depending on the complexity of the application domain and the scope of available products, it is possible to find different COTS solutions ranging from a single, large package or several specific packages that once integrated will provide the desired capabilities"[1].

different style of interaction, then developers have to introduce glueware to allow correct interactions. Yakimovich et al. in [17] suggest a procedure for estimating the integration cost. They list some architectural styles and outline their features with respect to a set of architectural assumptions. They define a vector of variables, namely the interaction vector, where each variable represents a certain assumption. An interaction vector can be associated either to a single COTS component or to an assembly. To estimate the integration cost they suggest to compare its interaction vector with the software architecture one.

The relationship between the *operational goals* and the assemblies is described by a matrix $SAT(|\mathcal{A}| \times n)$, where the element $SAT[j,k]$ is equal to 1 if the k-th goal is satisfied by the j-th assembly, and 0 otherwise.

For each non-functional requirement, if the satisfaction degree with which an assembly meets the corresponding goal is greater than or equal to the satisfaction degree required for that goal then the entry that corresponds to the goal and the assembly is set to 1.

Since we assume that each element of the assembly set $\mathcal{A}$ satisfies at least a functional requirement (see Section 1), for each row of SAT there exist at least an entry equal to 1. Clearly, in order to obtain a feasible solution, for each functional requirement there must exists at least an assembly satisfying it, i.e., for each column of SAT there must exist at least an entry equal to 1.

2.2.2 *Model Variables.*

The variables of our optimization model are:

$$x_i = \begin{cases} 1 & \text{if the } i\text{-th COTS components available} \\ & \text{on the market is chosen } (1 \leq i \leq |C|) \\ 0 & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if the } j\text{-th assembly is} \\ & \text{chosen } (1 \leq j \leq |\mathcal{A}|) \\ 0 & \text{otherwise} \end{cases}$$

2.2.3 *Objective Function and Constraints.*

The objective function consists in minimizing the total costs, i.e., the sum of the costs of the selected components and the selected assemblies, under the condition that each *operational goal* g_k must be satisfied by at least an assembly. We can formulate the problem as follows:

$$\min \quad \sum_{i=1}^{|C|} c_i x_i + \sum_{j=1}^{|\mathcal{A}|} \bar{c}_j y_j \qquad (3)$$

$$\begin{array}{rcll} \sum_{j=1}^{|\mathcal{A}|} SAT[j,k] y_j & \geq & 1 & \forall k = 1 \dots n \\ y_j & \leq & x_i & \forall j = 1 \dots |\mathcal{A}|, \forall i : C_i \in A_j \\ x_i & \in & \{0,1\} & \forall i = 1 \dots |C| \\ y_j & \in & \{0,1\} & \forall j = 1 \dots |\mathcal{A}| \end{array}$$

Note that the second set of constraints (i.e. $y_j \leq x_i$) is to guarantee that if an assembly is part of the solution then all COTS components making the assembly will be part of it too.

3. RELATED WORK

Several interesting approaches have been introduced for supporting the activity of selection of COTS components (e.g. [15], [4], [9]). A quite extensive list of these approaches can be found in [14]. However, all these approaches basically provide outlines to select the components, and only few of them support automation for this task.

In [7] the DesCOTS (Description, evaluation and selection of COTS components) system is presented. DesCOTS is composed by four tools that collaborate with each other. It is designed for supporting only quality requirements, and the component selection criterion is based on quality models. In [10] the OPAL tool is presented, which supports the definition of the requirements and the selection of the COTS components. Both DesCOTS and OPAL do not consider the possibility that a requirement can be satisfied by a COTS assembly, whereas we introduce this possibility here.

The DEER framework is based on an optimization model. Few other experiences have been based on optimization problems to automatize the development of a component-based system. For example, in our previous work [5] an optimization model is generated and solved by CODER (Cost Optimization under DElivery and Reliability constraints), a framework that supports “build-or-buy” decisions in selecting components. Based on the dynamic monitoring of a system, in [13] an optimization model has been suggested for the best allocation of the components on the hardware hosts while maximizing the availability of the system under some constraints, such as the allocation of two components on the same host. In [8] an optimization model is formulated to allow the best architectural decisions, while maximing the satisfaction of the quality attributes of the system using multi-objective optimization [3] under some constraints, such as budget limitations.

The following major aspects characterize the novelty of our approach:

- DEER could be adopted to automatize whatever approach supporting the activity of selection of COTS components (e.g. [15], [4], [9]).

- DEER is not tied to any particular component-based development process.

- DEER supports the study of the cost of construction of the system while adding, replacing and deleting requirements. The solution of the framework determines the COTS components to choose in order to minimize the cost of construction of the system while assuring the satisfaction of the functional and non-functional requirements, that can be satisfied by single COTS or assemblies of more COTS components.

- DEER allows to analyze how the cost of construction of the system varies while varying the targeted satisfaction degree for each requirement.

- The output of DEER can be an input of the design phase. In fact, the designers of the software architecture, basing on the set of assemblies provided by the framework, can start designing components and, mostly, the necessary gluecode to adapt and assemble them.

4. CONCLUSIONS

We have presented DEER, a framework that supports the selection of COTS components in the requirements phase of a component-based development process. The solution of the framework determines the COTS component to choose in order to minimize the cost of construction of the system while assuring the satisfaction of the requirements, which does not depend only on the architecture of the system ([18]). A requirement can be satisfied by a single COTS component or an assembly of several COTS components.

We are investigating several future directions.

We intend to enhance DEER in order to support the resolution of conflicts between requirements by assigning priorities to requirements and introducing the risk that considers the degree of uncertainty associated with the selection of a certain assembly of COTS components (like in [12]). Besides, we intend to deal with the dependencies that may exist between the requirements. For example, a functional requirements could be tied to a set of non-functional requirements. Therefore, the DEER framework could be constrained to suggest the assembly of COTS that satisfies both the functional and the related non-functional requirements.

We plan to introduce the possibility to develop a component in-house in order to enlarge the choice to the best set of COTS and in-house developed components. In DEER we assume the cost of a COTS component as an input value, whereas for in-house components we intend to introduce a development cost model (e.g. COCOMO [2]).

We also intend to experiment the model and the DEER tool.

Finally, as a long term goal, we work on the automatization of the process of selection of (COTS and in-house) components in all the phases of a component-based development process, as we have illustrated in this paper introduction. In particular, we are designing an integrated tool, based on several optimization model (such as CODER, DEER and others), that may assist software designers during the whole software component lifecycle.

5. REFERENCES

[1] Alves, C., Franch, X., Carvallo, J.P., Finkelstein, A. "Using Goals and Quality Models to Support the Matching Analysis During COTS Selection", *Proc. of ICCBSS 2005*, LNCS 3412, 146-156, 2005.

[2] Boehm, B. "Software Engineering Economics", Prentice-Hall, 1981.

[3] Censor, Y., "Pareto Optimality in Multiobjective Problems", Appl. Math. Optimiz., vol. 4, 41-59, 1977.

[4] Chung, L., Cooper, K., Courtney, S. "COTS-Aware Requirements Engineering: The CARE Process", *Proc. of RECOTS 2004*, September 2004, Kyoto, Japan.

[5] Cortellessa, V., Marinelli, F., Potena, P. "Automated Selection of Software Components Based on Cost/Reliability Tradeoff", *Proc. of EWSA06*, LNCS 4344, 66-81, 2006.

[6] Crnkovic, I., Chaudron, M., Larsson, S."Component-based Development Process and Component Lifecycle", *Proc. of 27th International Conference Information Technology Interfaces (ITI)*, Cavtat, Croatia, June, 2005.

[7] Grau, G., Carvallo, J. P., Franch, X., Carme, Q. "DesCOTS: A Software System for Selecting COTS Components", *Proc. of EUROMICRO 04)*, 2004.

[8] Grunske, L. "Identifying "Good" Architectural Design Alternatives with Multi-Objective Optimization Strategies", *Proc. of ICSEŠ06*, May 20-28, 2006, Shanghai, China.

[9] Kotonya, G., Hutchinson, J. "Viewpoints for Specifying Component-Based Systems", *Proc. of CBSE 2004*, LNCS 3054, Edinburgh, UK, May 24-25, 2004.

[10] Krystkowiak, M., Bucciarelli, B., Dubois, E. "COTS Selection for SMEs: a report on a case study and on a supporting tool", *Proc. of RECOTS03*, 2003.

[11] Lamsweerde, A. "Goal-Oriented Requirements Engineering: A Guided Tour", *Invited mini-tutorial paper, appared in Proc. of RE 2001*, 249-263, Toronto, August 2001.

[12] Mahmood, S., Lai, R. "Analyzing Component Based System Specification", *Proc. of AWRE 2006*, Adelaide, Australia, 2006.

[13] Mikic-Rakic, M., Malek, S., Beckman, N., Medvidovic, N. "A Tailorable Environment for Assessing the Quality of Deployment Architectures in Highly Distributed Settings", *Proc. of 2nd International Working Conference on Component Deployment (CD 2004)*, Edinburgh, UK, May 20-21, 2004.

[14] Navarrete, F., Botella, P., Franch, X. "How Agile COTS Selection Methods are (and can be)?", *Proc. of 31st EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA05)*, 2005.

[15] Ncube, C., Maiden, N. A. M. "PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm", *Proc. of CBSE99*, 1999.

[16] Nuseibeh, B. A., Easterbrook, S. M., Russo, A. "Making Inconsistency Respectable in Software Development", *Journal of Systems and Software*, 58(2), 171-180, 2001.

[17] Yakimovich, D., Bieman, J. M., Basili, V. R."Software architecture classification for estimating the cost of COTS integration.", *Proc. of ICSE 1999*, pages 296-302, June 15-21, 1999.

[18] http://www-128.ibm.com/developerworks/rational/library/4706.html