Integrating Architecture and Familiarization in CBD

Letizia Jaccheri
Department of Computer and Information Science
Norwegian University of Science and Technology
Norway
letizia@idi.ntnu.no

Marco Torchiano
Dipartimento di Automatica e Informatica
Politecnico di Torino
Italy
marco.torchiano@polito.it

Abstract

There are several attempts to model and provide guidelines for the development of COTS-based software systems. The literature offers a wide spread in terms of both viewpoints and focus. We present here our attempt to model the CBD process using a software process modeling language (E3).

Our work proposes a unified view in the form of a software process model which provides a formal abstraction of existing COTS-based development processes. In addition, our work proposes a process model for the acquisition phase, which recognizes the importance of architecture and gives room to a learning phase.

1 Introduction

As technology evolves, new processes for developing and maintaining software system continuously emerge and become used. COTS (Commercial-Off-The-Shelf) based development is one example of a class of software processes which have been recently proposed and studied by the software engineering community [10] [3] [8]. The COTS-based development process involves both the vendor organization that produces, sells, or distributes COTS products, the integrator organization that acquires and assembles these products, and the customer organization that purchases and pays for the final software system. In [3] the main feature addressed is the idea that the COTS-based development is a continuous trade-off between several factors: the requirements for the CBS, the COTS products offered by the marketplace, and the architecture of the CBS.

In [10] there is a description of the relationships between the software development organization, which integrates COTS products, the COTS vendor organizations, and the customer organization. In addition it defines the need for a phase at the beginning of the development where identification, evaluation, and selection of COTS products take place.

The COTS-based development processes proposed in the literature are described by informal notations stressing different aspects. The goal of this work is to provide a synthesis of the state of the art of the existing COTS-based development processes and propose some directions for improvement with special emphasis on the issue of learning about COTS. Our general idea is that process modeling can be used as a tool to model existing COTS based development processes. One of the goals of the software process modeling discipline is process understanding and analysis by synthesis.

The paper is organized as follows. First, in section 2, we summarize the state-of-the-art of COTS based development process [5] and we model the "standard" CBD process using a process modeling language. Then in section 3 we discuss the main issues of such a process in terms of improvements and discrepancies with the state-of-the-practice. In section 4 we present an enhanced CBD process that better addresses learning issues. Finally we draw our conclusions in 5.

2 Background

We formally describe an archetypical COTS-based development process by means of a Process Modeling Language (PML). Both the process model itself and the modeling activity allow us to reason about the process to identify drawbacks and propose changes. As we have experience with the E3 PML [6] and its associated tool, we adopt this environment to describe processes.

The process model we developed tries to integrate the aspects described by the different models found in the literature. Here we give some fragments of the process models we have developed. For each fragment, hints for discussion are given.

Figure 1 presents a simplified process model providing an overview and a context of the COTS-based development process. We use a subset of the E3 notation: rectangles denote tasks; books denote artifacts; hats denote roles; arrows between tasks and artifacts denote input and output relationships; and connections between tasks and roles denote responsibility.

The model is based on those found in the literature and consists of a limited set of concepts. This model defines the context to reasoning about COTS-based development. In our model, vendors produce and/or distribute COTS

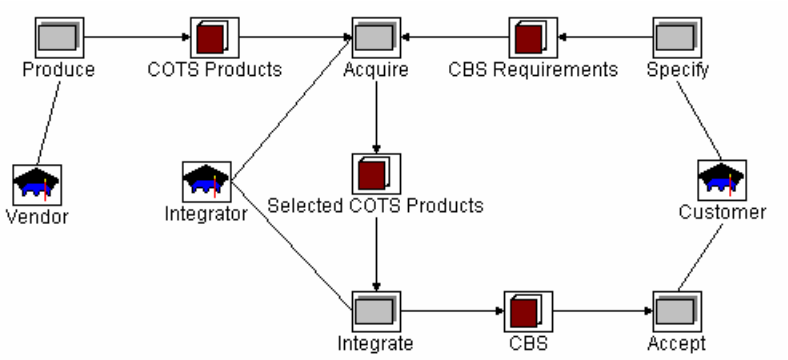

Figure 1: COTS-based development: a simplified view.

products and the integrator acquires one or more COTS products and builds them into a COTS-based system (CBS). Finally, the customer purchases the CBS.

This model fragment identifies three main roles. First, the *Vendor* is the organization that produces, sells, or distributes the COTS products. The Vendor is the source for both the product itself and its documentation. Often it provides some kind of support services which may be free or sold separately. Second, the *Integrator* is the organization that builds the COTS-based system assembling COTS products, possibly together with internally developed pieces of software (often called glueware) and components. Finally, the *Customer* is the organization that purchases and pays for the CBS. In this simplified view, requirements for the CBS come directly from the customer.

The proposed high-level model is in agreement with the one proposed in [10] about the actors that interact in the CBD process.

The CBD process model proposed in Figure 1 provides a context for the Acquire task. Figure 2 depicts the structure of the Acquire task. This is also based on acquisition processes found in the literature (e.g. [9] [11]); in particular we find the same activities (i.e. identification, evaluation, and selection) described in the model presented in [10].

3 Issues

The two process model fragments presented so far describe a picture of the current state of the art in the literature. Based on our experience, both interacting with the industry and teaching to students, in the development of COTS –based systems we identified some drawbacks and several areas for improvement.

By analyzing the acquisition process model, we observe at least three directions for improvement. First, the identification phase of the acquisition process must be extended to support COTS products discovery and familiarization. Then, the qualitative data that is available about COTS products must be managed and acquired by the integrator organization. Finally, the relationship between the architecture of the final system and the selected COTS products must be managed.

These directions are confirmed by the theses presented in [14]. One of the finding from a set of interviews is that formal selection procedures are seldom used. Familiarity either with the product or with the generic architecture is the key factor in selection.

Often the basic approach for COTS acquisition consists in searching the web using a search engine. Unfortunately this approach produces what is known as information overload, from which it is hard to derive useful information. Thus it is important to devote some

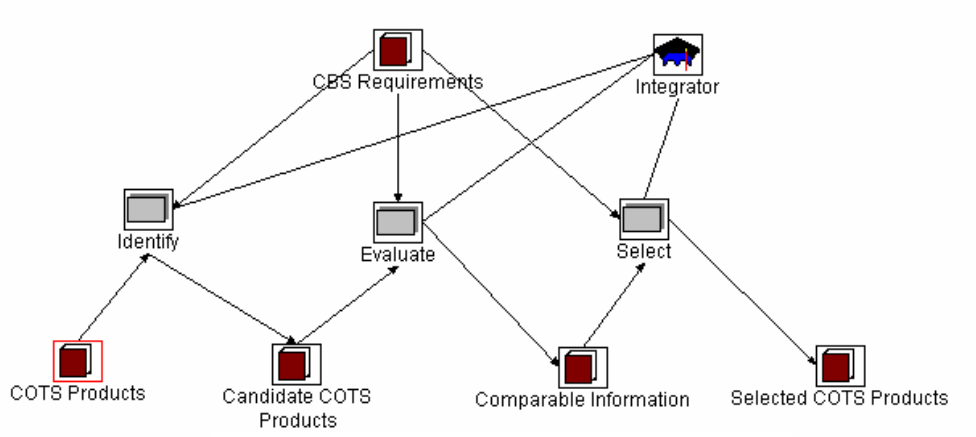

Figure 2: The structure of the Acquire task.

effort in putting order in this large amount of information. Moreover, it is difficult to find complete and trusted information about COTS products due to the bias of vendor in providing data about the products.

Concerning software architecture issues, in some cases the architecture is decided before detailing the requirements. Once the architecture is chosen, it places heavy constraints on the products selected. Some researchers propose a COTS products selection process largely based on requirements and their negotiation (e.g. [8] and [2]). While acknowledging that requirements are important, in [14] it is noticed that sometimes the real driver of the selection process and the fulcrum of all trade-off activities is the architecture.

4 Proposed extensions

We propose an extended COTS based development process in the form of a software process model. This process has been empirically validated in several occasions [7, 12, 13].

The process model fragment presented in Figure 3 proposes an improved Acquire task. With respect to the process model in Figure 2 there is a new task, called Learn task. The learning phase produces a set of COTS products that can be used as a starting point for the identification of potentially useful products. In addition, a preliminary software architecture is defined which can be used to drive subsequent identification and evaluation tasks. We have proposed a process model for this Learn task. We will also provide a discussion around the process model proposed in Figure 3. This model in fact can be added by important details which express the trade-off between the CBS Requirements and the Integration phase.

Acquisition approaches proposed in the literature are attribute-based, intended to measure product suitability with a set of stated requirements. Actually the relationship between the requirements for selecting a COTS product and the requirements for the whole CBS depends on the type of system. For turnkey systems the two requirements are essentially the same (i.e. what I would expect from the system is what I would expect from the COTS product). In the case of integrated systems the transition between COTS and CBS requirements should be mediated by software architecture. In this work we choose to adopt the definition of software architecture provided in [1]: “The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components and the relationships between them”.

The architectural components represent placeholders for COTS products, and the architecture as a whole dictates product requirements. In order to design such architecture, it is important to know which components can be used. Otherwise we may end up with an unfeasible architecture. Therefore, the integrator must have a thorough understanding of candidate COTS products before he or she can make an appropriate selection. Such knowledge may come from different sources, depending on the integrator organization's experience with CBS domains and application areas. Experienced developers may possess all the knowledge required to design a feasible architecture. Whenever such experience is not available, however, we need a means to procure the required knowledge.

The approach we suggest for acquiring knowledge about the COTS product is called learning process. It is made up of two main activities: classification and characterization.

Classification aims at classifying the products into equivalence classes, which group similar components that can be evaluated in similar ways. The classes can be similar to the domains described in [4].

Characterization evaluates in detail the products with the purpose of assessing the suitability for the project. This activity can be performed using for instance quality models as in [4].

The process involves two main roles: Facilitator and

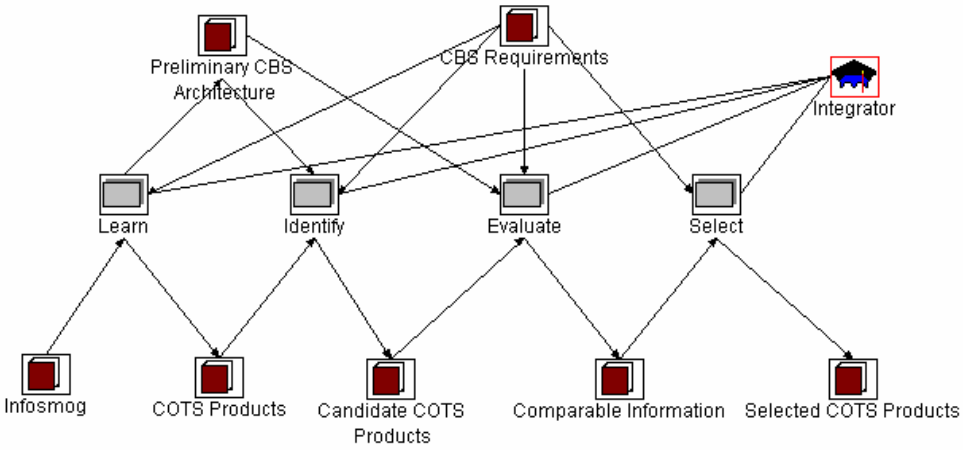

Figure 3: A proposal for an enhanced Acquire task.

Integrator. The Facilitator acts as a supervisor who, by defining the attributes to be measured, determines the details of the process. He can be either internal or external to the integration team, but in the latter case he must know the application domain thoroughly. The team that designs and develops the CBS serves in the role of Integrator.

We have validated our proposed process model in academic settings. Our investigation has been replicated twice and the forms which make our process models operational are available for reuse at http://www.idi.ntnu.no/~letizia/cots.html.

5 Conclusions

In summary, several COTS-based development processes have been proposed in the literature. These are different from the traditional water-fall like software processes since they are characterized by simultaneous definition of requirements, architecture and design, and trade-off between requirements and available COTS on the market place. In our work we have studied the existing COTS-based processes and provide software process models. These models facilitate the discussion about COTS-based development and give directions for improvement. In this position paper, we have provided an example of a general acquisition process and its improvement. The message is that although COTS-based development processes are neither linear nor simple, the application of process modeling techniques leads to models which can be analyzed, compared, discussed upon, and improved.

There are several directions for continuation of this work. The first is to continue to use the process in academic settings, to refine the forms provided at http://www.idi.ntnu.no/~letizia/cots.html and to tune the process especially to improve the part related to architecture definition. The second direction is to relate our proposed process to industrial settings. While from information conversations with industrial actors, our process seems to get agreement for adoption in industrial settings, we are still in the process of discussing questions around “why” and “how” we should try out our model in industry. Moreover, since we propose a model which aims at improving the way people learn about COTS, a validation of our proposal should rely on a model for measuring how much people have actually learned.

Due to the settings where we devised our processes there are features of the CBD process that are not covered. Mainly the negotiation of requirements based on the actual COTS products available on the marketplace.

As emerged also in the panel “Do We Need Requirements in COTS-Based Software Development?” at ICCBSS 2004, software architecture plays a key role in requirements negotiation. In this sense the preliminary architecture can support requirement negotiation.

6 References

[1] L. Bass, Clements, P., Kazman, R., *Software Architecture in Practice*: Addison-Wesley, 1998.

[2] B. Boehm, "Requirements that handle IKIWISI, COTS, and rapid change" *IEEE Computer*, 33 (7): 99-102, July 2000.

[3] L. Brownsword, T. Oberndorf, and C. A. Sledge, "Developing New Processes for COTS-Based Systems" *IEEE Software*, 17 (4): 48-55, July/August 2000.

[4] J. P. Carvallo, X. Franch, C. Quer, and M. Torchiano, "Characterization of a Taxonomy for Business Applications and the Relationships among them" Proc. of Int. Conf. on COTS Based Software Systems (ICCBSS 2004), Redondo Beach (CA), USA, 1-4 February, 2004

[5] R. Conradi, L. Jaccheri, and M. Torchiano, "Using software process modeling to analyze the COTS based development process" Proc. of Prosim'03, 2003

[6] L. Jaccheri, P. Lago, and G. P. Picco, "Eliciting Software Process Models with the E3 Language" *ACM Transactions on Software Engineering and Methodology*, 7 (4): 368-410, October 1988.

[7] L. Jaccheri and M. Torchiano, "Classifying COTS products" Proc. of 7th European Conference on Software Quality (ECSQ 2002), Helsinki, Finland, June 9-12, 2002, pp. 246-255.

[8] P. Lawlis, K. Mark, D. Thomas, and T. Courtheyn, "A Formal Process for Evaluating COTS Software Products" *IEEE Computer*, 34 (5): 58-63, May 2001.

[9] N. Maiden and C. Ncube, "Acquiring COTS Software Selection Requirements" *IEEE Software*, 15 (2): 46-56, March/April 1998.

[10] M. Morisio, C. Seaman, A. Parra, V. Basili, S. Condon, and S. Kraft, "Investigating and Improving a COTS-Based Software Development Process" Proc. of 22nd International Conference on Software Engineering, Limerick, Ireland, June, 2000, pp. 32-41.

[11] M. A. Ochs, D. Pfahl, G. Chrobok-Diening, and B. Nothhelfer-Kolb, "A COTS Acquisition Process: Definition and Application Experience" Proc. of 11th ESCOM Conference, Shaker, Maastricht, 2000, pp. 335-343.

[12] M. Torchiano, L. Jaccheri, C.-F. Sørensens, and A. I. Wang, "COTS Products Characterization" Proc. of Conference on Software Engineering and Knowledge Engineering (SEKE'02), Ischia, Italy, July 15-19, 2002, pp. 335-338.

[13] M. Torchiano and L. Jaccheri, "Assessment of Reusable COTS Attributes" Proc. of 2nd Int. Conference on COTS Based Software Systems, Ottawa, Canada, February 10-12, 2003

[14] M. Torchiano and M. Morisio, "Overlooked Aspects of COTS-Based Development" *IEEE Software*, 21 (2): 88-93, March/April 2004.