

Defining a Scope for COTS Selection Methods

Fredy Navarrete, Pere Botella and Xavier Franch
Universitat Politècnica de Catalunya
{fnavarrete, botella, franch}@lsi.upc.edu
http://www.lsi.upc.edu/~gessi

Abstract. The specification of a methodology defines a set of procedures and techniques that are associated to a specific domain. As part of this specification, it is advisable to establish a scope that allows identifying the set of roles and activities that should be covered to develop a life-cycle for a specific domain. If such a scope is not clearly defined in a methodology, some problems may arise, e.g., the set of roles in charge of carrying out the processes may lack of coordination, cooperation, and communication during the development of the life-cycle for the domain. In the Commercial Off-The-Shelf (*COTS*) components selection domain, there are currently different methodologies which define procedures and techniques to select or to license a *COTS* component from the marketplace. The application of these *COTS* selection methods results in processes that are different from usual development ones, yielding to new activities and responsibilities that should be defined in a scope which covers the interactions of specialized roles. However it may be observed that these methods do not put emphasis neither on the identification of these roles, nor on their subsequent interactions, nor on their combination to form a selection team. Furthermore, activities differ from one method to another. The contribution of this work is to define a scope for *COTS* selection processes, identifying and defining the undertaken activities, the roles that take place, their interactions and their responsibilities, and to organize a life-cycle around them. We use a goal-oriented approach, the i^* notation, and a framework to model the engineering process, the OPEN Process Framework (*OPF*), with the purpose of issuing a well-defined work team that can adapt itself to the internal processes of a particular organization.

1 Introduction

Commercial Off-The-Shelf (*COTS*) components are software components that may be purchased or licensed from the marketplace [13]. *COTS*-based systems require some specific software activities, and among them *COTS* components selection play a crucial role [14]. In some previous work [1, 2], we studied some of the most widespread *COTS* selection methods (*CARE* [3], *SCARLET* [4], *OTSO* [5], *EPIC*

Please use the following format when citing this chapter:

Navarrete, F., Botella, P., Franch, X., 2007, in IFIP International Federation for Information Processing, Volume 244, Situational Method Engineering: Fundamentals and Experiences, eds. Ralyté, J., Brinkkemper, S., Henderson-Sellers B., (Boston Springer), pp. 298-312.

[6], STACE [7], PECA [8]) with the purpose of analyzing if the agile principles and values briefed in the agile manifesto [9] influence them. We observed in this analysis that neither the human factor, nor the conformation of a selection team, were clearly defined within the processes suggested by those methods. But in fact, even a more relevant observation not specifically bound to this agile perspective, was concluding that COTS selection projects need specific roles and new activities to support a successful selection [10] but that in fact we may say that current methods either do not address this issue because they emphasise the analysis of the artifacts generated during the process, or they just outline some general recommendations but do not provide a comprehensive framework.

To tackle this point, we may recall that a methodology specifies a set of procedures, techniques, rules, and postulates employed by a discipline [11] which influences the development of a specific domain. In [12], Cockburn proposes to use structural terms (*Process, Milestones, Quality, Activities, Teams, Products, Techniques, Roles, Standards, Tools, Skills, Personality and Team Values*) to embrace a methodology to be applied to any team endeavour. Some of these structural elements are defined by a specific *Scope*. Cockburn defines the scope of a methodology as: *...consists of the range of roles and activities that it attempts to cover* [12]. Then, he characterizes the scope of a methodology along three axes:

- *Role coverage*: describes the set of roles that fall into the coverage of a methodology.
- *Activity coverage*: defines which activities of a specific project fall into the coverage of the roles in a methodology.
- *Life-cycle coverage*: mainly, specifies the coverage of a methodology over a life-cycle in a specific project.

Therefore, our attempt to clarify which are the roles, activities and life-cycle specificities in the domain of COTS components selection, may be rephrased as identifying and defining a scope for COTS selections methods, and this is the goal of our paper. Building such a scope requires taking into account the concrete roles and the main activities that must be assigned to cover a life-cycle of COTS selection processes. To define formally a scope for COTS selection methods, we have started by modelling COTS selection processes in a high level, to identify their main activities. We have used a process-focussed OO methodology, the OPEN Process Framework (*OPF*) [15], to formalize these activities. Next, we have identified the roles proposed by current COTS selection methods, describing some activities that they do not cover. Next, we have used a widespread goal-oriented notation, the *i** framework [16], to put together the roles conforming a selection team, and to state the interactions among these roles, and also among the selection team and its environment, obtaining a scope from a highly strategic perspective. Finally, we have considered the life-cycle perspective in order to complete the scope.

2 The Activity Dimension in the COTS Selection Scope

In some recent work [2] we have identified the most relevant processes that appear during COTS selection methods, we have decomposed these processes into

tasks, and finally we grouped these tasks into five categories. We summarize the result in this section.

2.1 Market Exploration

The COTS marketplace is composed of different kinds of technology segments from which COTS components are acquired or licensed. Currently, we may find a significant quantity of COTS information from the marketplace. Nevertheless, the market dynamics and continuous updates of COTS components, makes this information obsolete quickly. For this reason, we need selecting the necessary information produced in the market exploration process taking into account the provider information, the COTS component features themselves, and other aspects that we can reuse. In Table 1, we summarize the main activities undertaken in the market exploration process.

Table 1. MARKET EXPLORATION ACTIVITIES

Activities	Description
Candidate Component Identification	It identifies the candidate components from the market, classifying them into suitable categories and domains that fit to main requirements of the problem at hand.
Candidate Component Solution Identification	It identifies ensembles of related COTS components from the marketplace that conform to the required type of system.
Candidate Vendor Analysis	This task analyzes the main aspects of vendors from the marketplace.
Component Vendor Monitoring	Monitoring COTS vendors helps to establish relationships with mutual benefit in which users and vendors work together with each other.

2.2 Requirements Engineering

Requirements engineering applied to COTS projects heavily depends on the dynamics and evolution of the components available from the marketplace. The processes and technical tools that we use to steer the elicitation and specification of requirements must try to adapt user needs to the real state of marketplace. In table 2, we describe the main tasks involved in COTS requirements engineering.

2.3 System Architecture Analysis

Before integrating a component from the marketplace into a specific information system, we must consider the constraints, restrictions and composition of the system architecture. For this reason, we need a system description to know the main features over which we integrate the new components. In Table 3, we describe the five main activities that take part of the system architecture process.

Table 2. REQUIREMENT ENGINEERING ACTIVITIES

Activities	Description
Requirements Elicitation	This task comprises the set of activities necessary to carry out the requirements formulation. Therefore, activities such as to identify properly the user requirements, to hold a continuous requirements analysis, and to negotiate the requirements with user representatives, are essential activities in COTS projects to maintain the system architecture integrity, to understand the user needs, and to seek suitable components from the marketplace.
Business Analysis	It specifies the set of tasks necessary to compare the user requirements against organizational goals. For this reason, the market analysis is performed, identifying the suitable providers, analyzing the specific market technology segment where the organization technology is developed. Furthermore, user goals are analyzed throughout the COTS project to preserve the user vision.
Requirements Management	It includes the set of activities needed to manage properly user requirements; for example, activities to negotiate, to store and to control requirements are proposed.
Vision	This requirements engineering task produces and documents the vision of user representatives about a required component.

Table 3. SYSTEM ARCHITECTURE ACTIVITIES

Activities	Description
Architecture Reuse	It is focused on seeking reusable elements and artefacts within the system architecture. In order to carry out this task, we need an architecture description timely updated.
Architecture Prototyping	Considering this task can help us to support and validate the decisions that can impact over the system architecture
Architecture Production	This task identifies the set of features that compose the system architecture, determining their advantages, restrictions, and constraints against the user requirements.
Architecture Documentation	The purpose of this task is gathering the architecture information that we must store to obtain the component that we can reuse, and the added functionality of the system components
Architecture Integrity Assurance	This task preserves the architecture integrity to be not violated when we select or we integrate a new component in the system architecture

2.4 Candidate Component Evaluation

As suggested in previous categories, we can find different components in the marketplace that adjust to user requirements. The evaluation process must take into account techniques and tools that help to discriminate between the different options. The team that steers component evaluation must have either knowledge or experience in the component domain under evaluation. Therefore, the team members must be able to handle technical tools; besides, they must have a good understanding

of the users’ needs to evaluate components according to them. In Table 4, we describe the main tasks performed in candidate component evaluation.

Table 4. CANDIDATE COMPONENT EVALUATION ACTIVITIES

Activities	Description
Candidate Component Evaluation	Its responsibility is evaluating the COTS candidate component features with respect to user requirements.
Candidate Solution Component Evaluation	Ensembles of COTS candidate components which may be part of the possible final solution are evaluated.
Business Impact Analysis	This task analyzes the impact of component candidates over objectives of the business.
Candidate Vendor Analysis	This represents the main aspects of possible vendors, which influence component evaluation criterion.

2.5 Component Selection

We need considering different criteria to choose a suitable component, because neither the most expensive component nor the cheapest one are necessarily the most advisable components to integrate into the information system. There are a lot of aspects that play a crucial role when selecting a candidate component, such as the contract, the component aggregated functionality, the verification of the functionality offered by the COTS vendor and the integration ability, among other factors. In Table 5, we describe briefly the main tasks that are part of selection processes.

Table 5. COMPONENT SELECTION ACTIVITIES

Activities	Description
Component Vendor Monitoring	It represents the set of tasks to control and monitor the vendor representative in the selection processes and component integration.
Component Monitoring	It undertakes the tasks to control the component performance during the evaluation processes and selection. Besides, it proposes the control of future versions and releases of selected components.
Business Process Modification	Its responsibility is measuring the impact of selected components over the business goals.
Component Selection	It comprises the set of tasks for discarding or selecting the candidate components that are part of the possible final solution
Update Definition of System Architecture	This definition is relevant to maintain updated the changes of the system architecture when the COTS components are selected for integration.

3 The Role Dimension in the COTS Selection Scope

After addressing the activity dimension of the scope, we cope with roles. The situation is fairly different. Nowadays, the most used COTS selection methods have their own set of practices and suggested processes and activities. But as we have

pointed out in previous work [1], these methods do not fully succeed in considering individual motivations, as well as in defining the human factor, within their suggested processes. In fact, building a work team is considered a secondary aspect in conventional methodologies [19]. Although a set of roles is included in some methods, they are more focused on the artifacts produced by the roles than on the formalization of the specific scope and the composition of the team in which all the roles will develop their activities. In table 6 we summarize the most relevant statements about COTS selection roles found in these approaches [3, 6, 8, 20], complementing them with some roles specified in the RUP-for-COTS proposal., for lack of space we have highlighted some roles that take part in the selection processes and that not overlap with other roles [21].

Table 6. DEFINITION OF ROLES IN SEVERAL COTS SELECTION METHODS

COTS Method	Roles Suggested
CARE [3]	“Requirements engineer: complete and correct description of users needs with a technical background”
	“Software architect: defining the outline for the software architecture and defining the baseline software architecture”
	“Engineer component: maintaining the component repository”
	“Component vendor: complete and correct information about the component products”
IEEE 1062 Software Acquisition Standard [20]	“Acquirer: a person or organization that acquires or procures a system or software product (which may be part of a system) from a supplier”
	“Developer: a person or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the life cycle process”
	“Supplier: a person or organization that enters into a contract with the acquirer for the supply of a software product (which may be part of a system) under the terms of the contract”
EPIC [6, 21]	“Acquirer: identify relevant COTS packages and vendors”
	“System architect: oversees the entire system and all factors that might affect its development”
	“Designer: specify the COTS package configuration requirements”
	“Data designer: ...mapping between the data sources and the target databases...”
PECA [8]	“Evaluator: should have technical experience”
	“Charter: defines the scope and constrains of the evaluation”
	“Evaluation stakeholders: are those individuals or groups with vested interest in the results of a COTS evaluation...”

The roles of this set are mainly concerned on their work products, rather than defining the interactions that take an important part at the moment of selecting a COTS component. For example: in activities as “candidate component identification”, “business process modification”, “update definition of system

architecture”, and “architecture production” (among others), the specific contribution of all team members to develop these activities is not defined.

In the rest of the section, we propose a set of roles primarily based on Table 6 but complemented with some missing roles covering the activities not included in current COTS selection methods. As a result, we have identified nine specific roles for COTS selection, which are identified in Table 7 with the following capital letters:

- A** *System Architect*: Defines the structure of the information system, identifying constraints and technological specifications that compose it.
- B** *Market Watcher*: Explores the marketplace segments involved in the undertaken selection process to find the candidate COTS components which are to be evaluated and assessed with respect to user specifications.
- C** *COTS Component Evaluator*: Evaluates candidate COTS components which are assessed with respect to user requirements using the appropriate techniques. Experience in the component domain under evaluation is required.
- D** *Requirements Engineer*: Guides the elicitation, negotiation and validation of user requirements. To do so, he or she needs a minimum technical background and socialization ability.
- E** *COTS Vendor Interface*: Communicates with a particular COTS component provider company, trying to involve it inside the project, looking for mutual benefits of both parts.
- F** *Stakeholder Representative*: Someone who has an interest on the system-to-be and, as a consequence, has an interest on the success of the selection process.
- G** *COTS Data Expert*: Evaluates and stores the information that is produced during the process, part of which may be used in future selection projects taking place in the same or similar domains.
- H** *COTS Lawyer*: Protects the company interests at the moment of acquiring or licensing a component, collaborating in the writing and review of contracts.
- I** *COTS Provider*: for providing detailed information and demos of components during detailed analysis.

In Table 7 we show the correspondence of those proposed roles with the ones identified in Table 6 (each column stands for a role using the capital letters introduced above). Besides this set of roles specific for COTS selection, other transversal ones can be incorporated in the selection team. Among them, we consider at least the Project Manager [22] and the COTS Quality Engineer. The existence of these roles will be made explicit in the next section.

Once the roles that compose a COTS selection team have been identified, we address the different interactions that may occur among them. With this purpose, we use the i^* framework [16] basically because of two reasons: 1) it includes roles as part of its model elements; 2) it is possible to declare both high-level and low-level interactions, using the same model element (dependencies). For this reason, we use Strategic Dependency (*SD*) models to identify the *Strategic Dependencies* that arise inside a selection team. We use the *RiSD* methodology to construct this *SD* model [23], because *RiSD* suggests a construction guide and specific syntax for each constructor of an *SD* model.

Table 7. MAPPING THE PROPOSED ROLES WITH EXISTING ONES

COTS Method	Roles Suggested	A	B	C	D	E	F	G	H	I
CARE [3]	Requirement engineer				X					
	Software architect	X								
	Engineer component							X		
	Component vendor					X			X	X
IEEE 1062 [20]	Acquirer						X			
	Developer	X		X	X					
	Supplier					X			X	
EPIC [6]	Acquirer		X							
	System architect	X								
	Designer				X					
	Data designer							X		
PECA [8]	Evaluator			X						
	Charter			X						
	Evaluation stakeholders						X			

In our scope, *i** models consists of a set of nodes that represent roles and a set of dependencies that represent the relationships among them, expressing that an actor (*dependor*) depends on some other (*dependee*) in order to achieve some objective (*dependum*). The *dependum* is an intentional element that can be a: *resource* (a physical or informational entity), *task* (particular way of doing something), *goal* (condition or state of affairs in the world that the actor would like to achieve) or *softgoal* (a condition in the world which the actor would like to achieve, but the criteria for the condition being achieved is not sharply defined a priori, and is subject to interpretation) [16, 23] (see Figure 1 for a legend).

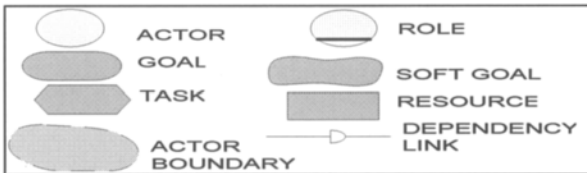


Fig. 1. Graphical representation of *i** constructs.

In Figure 2 we can observe the SD model that identifies the interactions among the members of a selection team. In the model we may distinguish the selection team (whose boundary is drawn in green) that contains the different roles defined in previous sections. Furthermore, some external actors appear, which represent the environment in which the selection team operates: the *Organization* interested in the selection, the *Information System* under construction, the *COTS Marketplace* and the *Vendor Representative* company. Dependencies among these actors and the roles inside the selection team are also included in the model.

We explain next, the most important interactions that appear in the model (we use the capital letters to identify the abbreviations of each role):

- *Stakeholder Representative (UR)*: depends on *Requirements Engineer* to validate his/her requirements, because the *Requirements Engineer* must negotiate and steer the user needs.
- *System Architect (SA)*: depends on *Market Watcher* to compare the candidate components with the system architecture, for this reason the *Market Watcher* has to explore the marketplace to find components that will be evaluated.
- *Requirements Engineer (RE)*: depends on *Stakeholder Representative* to negotiate user requirements, because the *Stakeholder Representative* has to adapt his/her requirements to the market.
- *COTS Component Evaluator (CE)*: depends on *Market Watcher* to evaluate candidate components, because the *Market Watcher* must explore the marketplace to find components to be evaluated.
- *Market Watcher (MW)*: depends on *Requirements Engineer* to locate the candidate components, since the *Requirements Engineer* must define user requirements with the purpose of driving the component search in the market.
- *Vendor Representative (VR)*: depends on *COTS Vendor Interface* to answer to the organization needs, since the *COTS Vendor Interface* is the communication bridge between the organization and the provider.

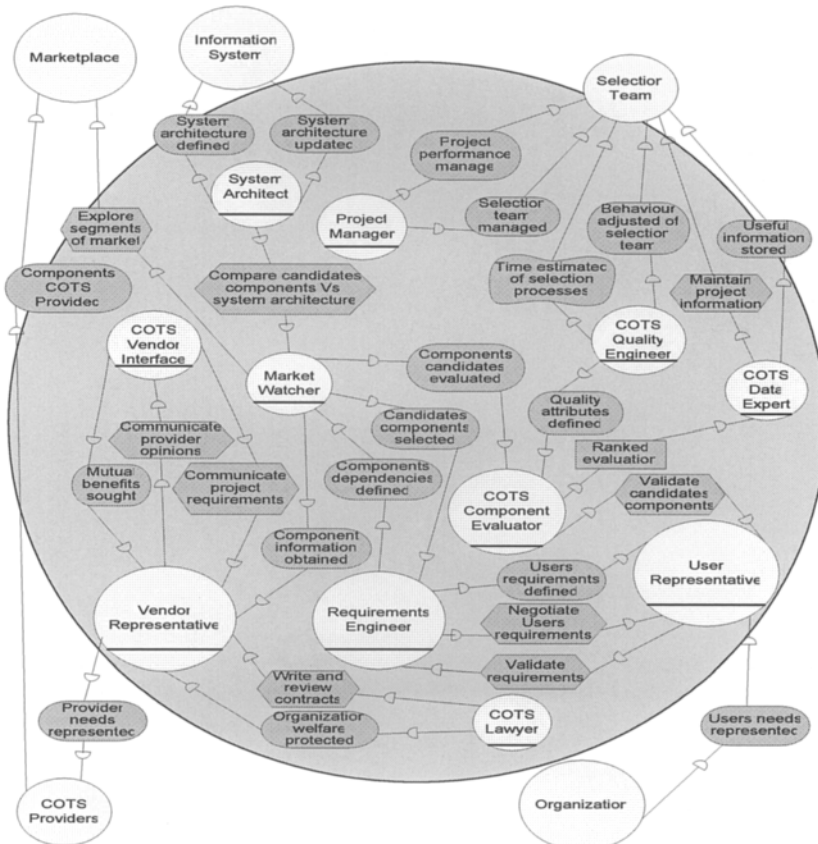


Fig. 2. Interaction of roles in a strategic *i** model.

- *COTS Vendor Interface (VI)*: depends on *Vendor Representative* to communicate him/her the project requirements, because the *Vendor Representative* can provide information to select a suitable component.
- *COTS Lawyer (LW)*: depends on *Vendor Representative* to write and review the acquisition contracts since the *Vendor Representative* is the owner of components that the organization wants to acquire or license.
- *COTS Quality Engineer (QE)*: depends on the whole selection team to estimate the time for each process, because the selection team has the knowledge for doing this task.
- *COTS Data Expert (DE)*: depends on the whole selection team to store useful information, since the selection team must take advantage of this information.
- *Project Manager (MN)*: depends on the whole selection team, because only with its help, the project manager can achieve his objectives.

4 Defining a scope formally

In this section we aim at formalising the scope for COTS methods, defining the roles and the activities to cover the main selection processes. To do so, we use OPEN, a framework consolidated in the field of process modelling. OPEN was created by a group of methodologists, researchers, tool vendors and practitioners [17], which includes concepts bound to business modelling, business decision making, maintenance, and application development. Our main purpose is to take the OPEN processes repository defined in [18] (OPF, the Open Process Framework) making stress in the activities that should be undertaken to carry out COTS selection processes, which belong to the five categories mentioned in section 2. On the other hand, in OPF, the roles compose teams and these teams are part of organizations; for lack of space, we focus on the roles hierarchy without specifying what kind of organizations or what kind of teams the roles compose.

In Figure 3, we present our COTS selection role hierarchy. We identify which roles are taken from the OPF (shaded boxes) and which are specific COTS selection roles (thick-lined boxes). These roles are classified according to two kinds of OPF roles, *Internal Role* (“it is a producer internal”) and *External Role* (“it is a producer external, outside of the work product to be developed but it is relevant to the development process”). As a class of *External Role*, OPF proposes the *Representative* abstract class, which corresponds to a person that represents a specific type of organization or group of people that have common interests. Some of the roles identified in the previous section are defined as concrete classes that inherit directly from *Representative*:

- o *Vendor Representative* (OPF), is a representative of the COTS provider company, with the purpose of providing detailed information and demos of components, among others benefices;

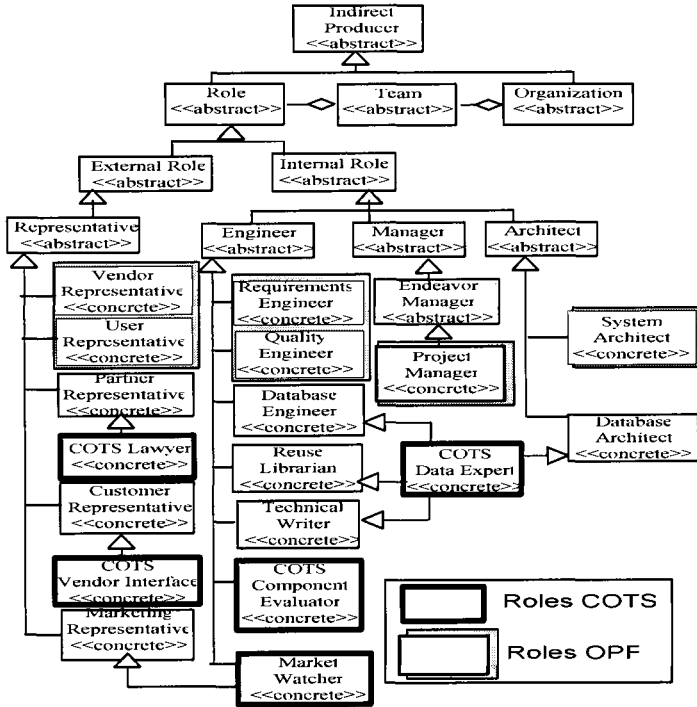


Fig. 3. Formalization of COTS selection roles using the OPF.

- *User Representative* (OPF), representing the needs of the stakeholders in the selection team;
 - *COTS Lawyer* (COTS), this class inherits from *Partner Representative* (OPF) the knowledge about the contracts that are carried out with the *Vendor Representative* (OPF);
 - *COTS Vendor Interface* (COTS), this class makes part of organization that requests a component, thus it inherits from *Customer Representative* (OPF);
 - *Market Watcher* (COTS), this class inherits from two OPF classes: as a representative, it inherits from *Marketing Representative* (OPF), because it uses the knowledge about the marketplace where the organization technology is developed; as an *Engineer* (OPF), it needs technical skills to perform marketplace exploration, e.g. classifying technological segments therein.
- On the other hand, among the *Internal Roles*, we find in OPF three abstracts classes of interest for our work. The first of them is *Engineer*, and as descendants we define:
- *Requirements Engineer* (OPF);
 - *COTS Quality Engineer* (COTS), helps in the definition of quality attributes of COTS components. This class inherits from *Quality Engineer* (OPF) the skills to provide improvement and time estimates of each selection activity;
 - *COTS Data Expert* (COTS), this role inherits from *Database Engineer* (OPF), *Technical Writer* (OPF) and *Database Architect* (OPF) because is the role for a person who creates, maintains and plans the data structure that will support the

information that somebody can reuse inside the project or in future projects (for the project it is very important storing the information that someone can use without documenting each process excessively);

- **COTS Component Evaluator (COTS)** and **Market Watcher (COTS)** are very-specific COTS roles that we define them as direct heir of *Engineer*.

The next abstract class that inherits from *Internal Role* is *Manager* (OPF), which makes reference to the administration activities carried out by a person. We find a class that inherits from *Manager*:

- **Project Manager (OPF)**, which corresponds to the person in charge of representing the selection team at the organization. A person playing this role drives the work team through the selection process. This class inherits from other abstract class *Endeavor Manager* (OPF), because this class has the necessity of carrying out the project goals.

The final abstract class that inherits of *Internal Role* is *Architect*, which makes reference to the person that produces a specific architecture. We can find a concrete class in this hierarchy:

- **System Architect (OPF)**, because this class has to describe the structure of information system.

With the model depicted in Figure 4, we are able to identify the abstract activities that roles must undertake during specific projects, where concrete activities have a set of task to make a work (*Work Unit*) during COTS process, and where we can classify them to be reused in future projects.

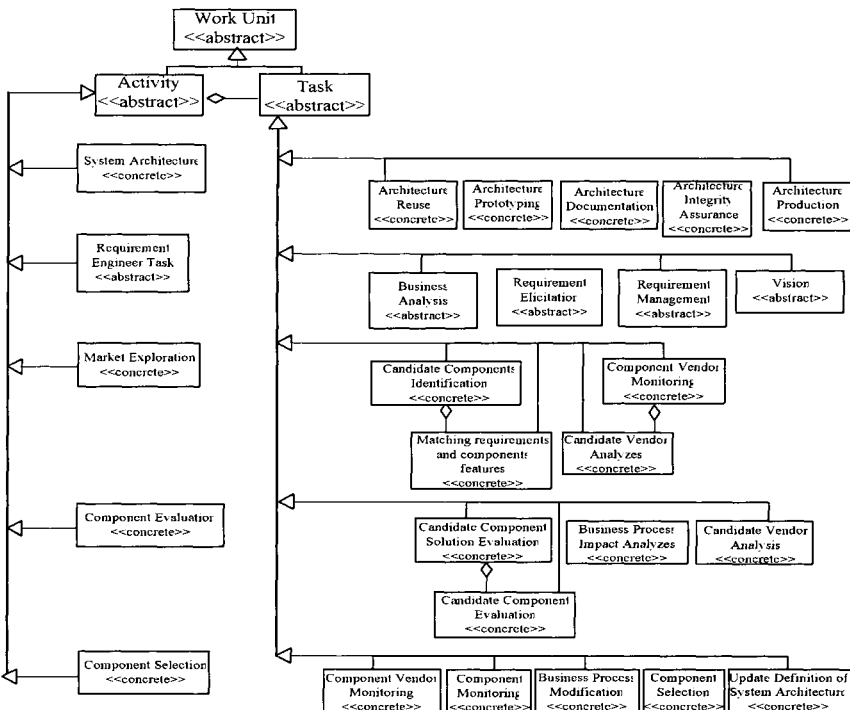


Fig. 4. Formalization of activities using the OPF.

6 Life-cycle coverage

COTS selection processes are continuous processes, and usually they arrive at their end when the development of the information system under study does not evolve across the time. For that reason, the interaction of the roles during the life-cycle of an information system is constant. Therefore, we want in this section to represent the continuous movement of COTS process as a kind of orbital system, with activities gravitating around the project at the center, where capital letters *SA* represent software architecture, *RE* requirement engineering, *ME* market exploration, *CE* component evaluation, and *CS* component selection (see Figure 5).

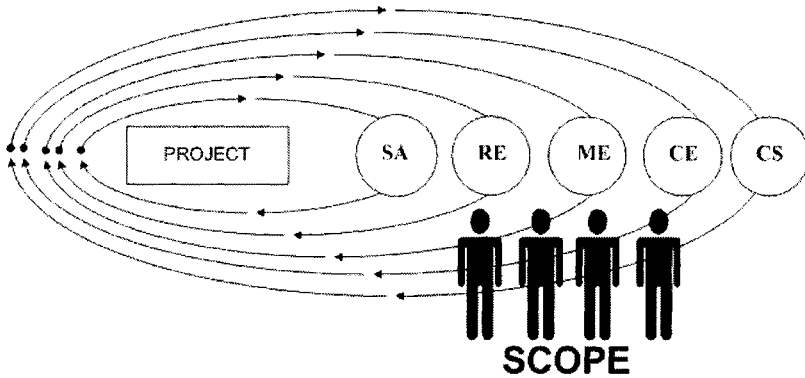


Fig. 5. Life-cycle coverage and roles coverage.

In COTS projects it is common that we can select more of one COTS component from the marketplace. Therefore, the scope defined within a specific project (*role coverage*, *activity coverage*, and *life-cycle coverage*) helps us to know in an explicit time instant T , the set of roles and the activities that are adaptable to the needs of project. For instance, in a time $T1$ within a COTS project X, the priority can be centered in the component evaluation, and the second level of importance could be the definition of the system architecture to obtain a successful selection, because the market exploration and the requirement engineering have been performed and they have been controlled for a time $T1$. If we have defined previously a scope for our project, we can obtain for a time $T1$ a life-cycle coverage, role coverage, and activity coverage to develop our COTS project, and doing so we gain in knowledge and learning over the process carried out.

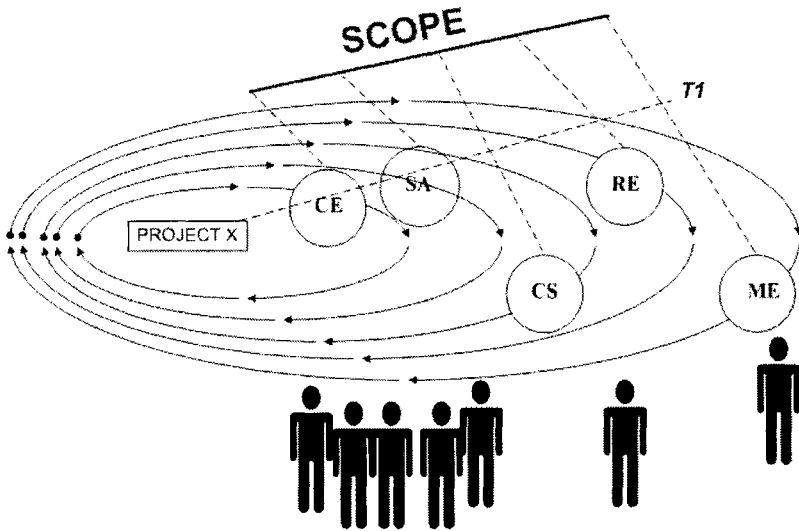


Fig. 6. Identifying a scope for a specific time $T1$

7 Conclusions

There are several COTS selection methods available, but they do not define formally the interactions among the roles and the activities needed to cover the life-cycle in the COTS projects. For this reason, we have proposed the definition of a scope to improve the process engineering perspective of those methods. Defining and identifying the roles and the activities that may be needed in COTS selection projects provides an improvement in the maturity of COTS processes and helps to identify the endeavors that are needed during the COTS project development. The use of formal or at least rigorous frameworks such as OPF and i^* has been of great help to identify the roles and the activities involved in selection processes, and as a result, we have defined the roles and the activities identified in the previous section by contextualizing them in the OPF.

Our research agenda primary includes the definition of a COTS selection method built upon our presented proposal, based in a model able to consider what scope is necessary for a whole COTS-based development life-cycle.

References

1. Navarrete, F., Botella, P., Franch, X. "How Agile COTS Selection Methods are (and can be)?" in Proc. Euromicro 2005. Porto, Portugal.

2. Navarrete, F., Botella, P., Franch, X. "Reconciling Agility and Discipline in COTS Selection Processes" in Proc. Commercial-off-the-Shelf (COTS)-Based Software Systems, 2007. ICCBSS '07.
3. Chung, L. Cooper, K. Courtney, S. "COTS-Aware Requirements Engineering and Software Architecting" in Proc. SERP 2004.
4. Maiden, N. Kim, H. Ncube, C. "Rethinking Process Guidance for Selecting Software Components" in Proc. 1st ICCBSS, LNCS 2255, 2002.
5. Kontio, J. "A Case Study in Applying a Systematic Method for COTS Selection" in Proc 18th Intl' ICSE, 1996.
6. Albert, C. Brownsword, L. "Evolutionary Process for Integrating COTS-Based System (EPIC): An Overview". Carnegie Mellon University, Software Engineering Institute CMU/SEI-2002-TR-099 ESC-TR-2002-009, July 2002.
7. Kunda, D. "STACE: Social Technical Approach to COTS Software Evaluation" in Proc. Component-Based Software Quality - Methods and Techniques, LNCS 2693, 2003.
8. Dorda, C. Dean, C. Morris, E. Oberndorf, P. "A Process for COTS Software Product Evaluation." in Proc. 1st ICCBSS, LNCS 2255, 2002.
9. Beck, K., et al. Manifesto for Agile Software Development. Available at: <http://www.agilemanifesto.org>
10. Ncube, C. Maiden, N. "PORE: Procurement Oriented Requirements Engineering Method for a Component-Based System Engineering Development Paradigm." in Procs. 2nd International Workshop on Component-Based Software Engineering (CBSE), 1999
11. Merriam-Webster, On-Line Dictionary <http://www.m-w.com/dictionary/methodology>. Last Update April 2007.
12. Cockburn, A. "Agile Software Development". Addison Wesley 2000-2001.
13. Meyers, B., Oberndorf, P., Managing Software Acquisition: Open Systems and COTS Products, Addison-Wesley, 2001.
14. Finkelstein, A., Spanoudakis, G., Ryan, M., "Software Package Requirements and Procurement", in Procs. 8th IEEE IWSSD, 1998.
15. Firesmith, D. Henderson-Sellers, B. Graham, I. "OPEN Modeling Language (OML) Reference Manual". Cambridge Univ. Press, New York, 1998.
16. Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" in Proc 3rd IEEE Int. Symp. on Requirements Engineering, RE'97. Washington, USA.
17. Henderson-Sellers, B. "The OPEN framework for enhancing productivity" Software, IEEE 17(2), March-April 2000 Page(s): 53 – 58
18. Firesmith, D. Henderson-Sellers B. Graham, I. OPEN Process Framework (OPF). Available: <http://www.opfro.org/>. Last update April 2007.
19. McBreen, P. "Questioning Extreme Programming". Addison Wesley, 2003.
20. IEEE recommended practice for software acquisition, IEEE Standard 1062, 1998.
21. Pannone, R. Peraire. C. "The IBM Rational Unified Process for COTS-based projects: An introduction". Available at: <http://www-128.ibm.com/developerworks/rational/library/aug05/peraire-pannone/>
22. Albert, C. Brownsword, L. "Evolutionary Process for Integrating COTS-Based System (EPIC): An Overview". Carnegie Mellon University, Software Engineering Institute CMU/SEI-2002-TR-099 ESC-TR-2002-009, July 2002.
23. Grau, G., Franch, X., Mayol, E., Ayala, C.P., Cares, C., Haya, M., Navarrete, F., Botella, P., Quer, C.. "RiSD: A Methodology for Building i* Strategic Dependency Models". In Proc 17th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2005.