



HAL
open science

IS-COTS: a help to COTS Products Integration

Philippe Roose

► **To cite this version:**

Philippe Roose. IS-COTS: a help to COTS Products Integration. IOTSDM - Third International Off-The-Shelf-Based Development Methods Workshop (IOTSDM '08) - Position Paper - Held in conjunction with the Seventh International Conference on COTS-Based Software Systems (ICCBSS '08), Feb 2008, Madrid, Spain. hal-00346774

HAL Id: hal-00346774

<https://hal.science/hal-00346774v1>

Submitted on 17 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IS-COTS: a help to COTS Products Integration

Philippe Roose

LIUPPA/IUT de Bayonne

2, Allée du Parc montauray

64600 Anglet / FRANCE

Mail: Philippe.Roose@iutbayonne.univ-pau.fr

Abstract

This presentation is about an Information System that will help selecting COTS Product. First we explain what is a COTS Product and its identity card in order to help the description a one COTS. Then we present an information system and the requirements for an efficient integration. We propose a set of environments and scenarios in order to help the information retrieval and the detection of problems of non-compatibility. We will end by presenting a tool implementing the IS and helping developers to check the selection and to develop prototypes.

Keywords

COTS Product, Information System, Prototype, Integration

1. Introduction

The notion of commercial-of-the-shelf (COTS) becomes classical. More and more organization tend to acquire COTS instead of developing more or less from scratch the application. The integration of COTS tools into software application becomes an important and desirable task [7].

The idea of reusing pre-existing products is not new, but good specific skills and tools processing such softwares are hard to find. How can we select them, how do we evaluate them? How can we integrate them? And how to replace them ?

In order to narrow these limitations, we favor the use of an environment that will make easy the development of COTS Products implemented prototypes.

This article proposes an information system to facility the integration of COTS products.

Our presentation offers a solution based on an Information System whose goal is to make the collecting of information easier, not only on each pre-selected COTS, but also on integration constraints: it measures the difficulties, how hard it will be to realize the project, and shows the incompatibilities.

The paper is organized as follow: in the first part will give a detailed explanation of what is a COTS Product. The second part will present our analysis based on an

Information System, followed by a prototype providing an evaluation. The third and last part presents the IS and the prototype based on it that will help the selection.

2. COTS

Everybody agrees on the following COTS (*Commercial Off The Shelf*) definition: a ready made component available for sale to the general public (www.webopedia.com). This very general definition raises many expectations.

First of all, a COTS is a software product that can be found in various versions whose source code can be open or not, sold, rented or given free by a salesman. Regular updates and new functions are available.

With such a wide open definition, many softwares can be considered COTS. Therefore, let's draw a more restrictive but more appropriate definition that will include what is required from a COTS Product:

A COTS Product is a product offering one or several services required for the execution of another application.

This general term of COTS is incomplete. It covers five aspects, each distinct from each other: *COTS System*, *COTS Based Product*, *COTS Software*, **COTS Products**, (our topic in the paper) and *COTS Component*.

The words *COTS System* [1][5] covers COTS as a set of softwares. As such, it won't be the product itself that it will be studied, but the system as a whole.

The COTS Software helps identifying the set of products. Using this term, many articles will analyze the selection and compare the different products. *COTS Software* is more restrictive than *COTS System*. *COTS System* offers a global vision of the set of COTS composing the application, whereas *COTS Software* focuses only on COTS, apart from the application to implement.

Another restriction in COTS will be found in the third function of *COTS Product*, which points out the definition of a product and the problems linked to its integration inside the application. In this framework, the relations, comparisons, and constraints between products are not seen

as before. Only functionalities or technical matters regarding integration are then considered.

We can conclude that *COTS Components* will rather focus on software components' technique. It is a specific view of COTS Product with components designed to be integrated.

3. Using a Prototype

The use of COTS (ie COTS PRODUCTS) in a software development process implies a good client/provider communication, in order to target each COTS specificity that will be useful for the customer.

According to ISO/IEC 9127 [6], there are close to 600 risky conditions or input parameter combinations. That explains why getting information is hard. Moreover, these numerous risky conditions and the weak information structuring, lead to a scattering and a lack of accessibility. COTS based developers often say that they need to prototype their application to integrate one or several COTS in order to:

- ✚ Experiment and test the veracity of explanations given for the COTS according to its production/services.
- ✚ Collecting qualitative and quantitative information
- ✚ Collecting information on compatibility, execution environment, etc.
- ✚ Validate information and ensure the processes' validity with correct test beds.

The collected information will help creating a correct COTS evaluation, and will be used to facilitate the selection steps. Nevertheless, obstacles can appear on the way to realizing

prototypes: The first one is to obtain the information necessary to integrate COTS. They won't always be easy to retrieve. While implementing a prototype it will become possible to know the means to integrate a COTS, as well as each COTS' the specific method.

A good communication between the client and the provider is essential for an efficient collecting of information. This way, the developer will get an easy access to the necessary data to implement his prototype.

Collecting these information will also allow using COTS apart from its integration, and will give as well the necessary source code needed for its integration. The application will also function with the whole set of COTS.

This is a three step phase:

✚ *Using COTS*

We first identify the set of softwares necessary to execute the COTS and its services. Recursively, A COTS may require another one and so on. More generally, with the set

of COTS, we will get a dependence graph [8]. This step makes possible the identification of all the softwares that are not 100% optimum, and of the unavoidable compatibility problems.

✚ *COTS Integration*

At this point we identify the set of softwares that are needed to implement the integration of the COTS into the application. These softwares are: development tools, API (Application Programming Interfaces), etc. We'll also identify implementation languages to work with, and a list of source code samples making integration easier.

On a second level, this step will show configurations that are not suitable for the integration of some COTS.

✚ *Certification of Information*

This last step clearly identifies the author of the information, certainly the best source to get reliable information.

By the way, the information system that we propose will provide an easy access to information, a good communication between clients and COTS providers, as well as an automatic checking of the information's reliability. In order to avoid all ambiguity, we shall stress that we do not offer an evaluation nor an estimation of the risks and the cost process. Neither do we propose a system that helps the selection of COTS. We begin once the choice of COTS has been made.

4. Information System

In order to build the information system we need some additional information:

- ✚ A unique identifier for each COTS (as MAC address for network cards for example) This identifier could be the international ID of the company providing the COTS plus the name and version of the COTS.
- ✚ A reference to other COTS or other applications (for example pre and/or post requirements or alleged non compatibilities)
- ✚ Information on COTS use and integration.

This information is put together in one entity called ICOTS (*Identity Card for COTS*) [3]. It provides information's summary that will help identifying the COTS.

4.1 Environment and scenarios

We propose the notion of environment and scenario. A set of available scenarios will always surround an integration environment, but some of them won't be valid.

Each scenario is a particular case to evaluate. To describe a prototype or a COTS, we will identify three environments: execution, integration and implementation environments (see Figure 1).

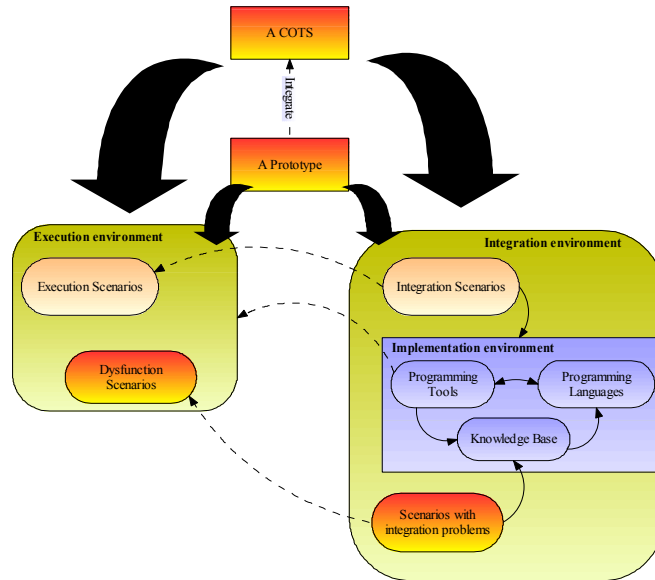


Figure 1 : Three environments

The information is provided by COTS designers and providers. They are completed by actors called clients during the COTS-based application's development steps. They all have to be linked to the products ie the ICOTS ID cards. From this ICOTS we deliver information regarding integration and execution environments. Therefore, when COTS providers will need to describe an environment for their product, they can refer to another ICOTS that will describe its own environment.

With successive references, it is possible to get the list of all software elements needed for a specific environment and therefore to detect incompatibility problems. Lets take an example: we need COTS "A" & "B" for a prototype. "A" needs the "Y" product in order to be executed, "Y" needs "Z" which is non-compatible with "X". "B" needs the "V" product needing "X". In this example, we can conclude that the execution of A & B simultaneously is not possible. This is a non-compatibility problem that the developer will have to solve. He can choose to use either one of the two COTS and develop the services that haven't been provided by the missing COTS. Or he can choose to replace one of them, if a similar one is available.

A set of reports will help the developer to control the numerous informations collected from the products. Some will help him when realizing integration (cf. integration scenario). Others will warn him on execution/implementation configurations that have raised

problems to other COTS users. The use of evaluation prototypes greatly helps to qualify and estimate risks and cost.

Report on execution environment:

- ✚ Set of softwares needed to the execution of each COTS.
- ✚ List of incompatible softwares.
- ✚ List of compatible operating systems common to all COTS.

Report on integration environment:

- ✚ List of softwares needed for integration. Services provided by selected COTS and those remaining to develop tools for integration.
- ✚ Documents linked to each COTS service.

Report on failures:

- ✚ List of non-compatibilities for each COTS.
- ✚ Identification of the COTS responsible for non-compatibility in an assemblage.

4.2 IS actors

The COTS provider and designer (called FourCo)

Within a classical distribution circuit, one must dissociate the manufacturer of a product from the distribution circuit or the vendor. Most of the time, they represent two distinct entities. But in the case of the relation we present today, it is not relevant to know who provides the product and who gets the information needed by the client. The only difference should be the quality of the service provided by the various distribution circuits. This criterion is important for the selection of the COTS vendor, but has no impact on the quality of the service nor the choice of the product. This is the same product, whoever the vendor. So we do

not need to identify two distinct actors. We focus on information needed by clients in order to select and integrate COTS. We particularly want to help them implement a prototype, and not the vendor selection. That is why we will consider only one actor, called FourCo. His role is to provide information for his clients. He alone will be in charge of all information asked by the client. FourCo's first requirement is to initialize a good client/provider relation and to spread information on the product that they are setting up for their client.

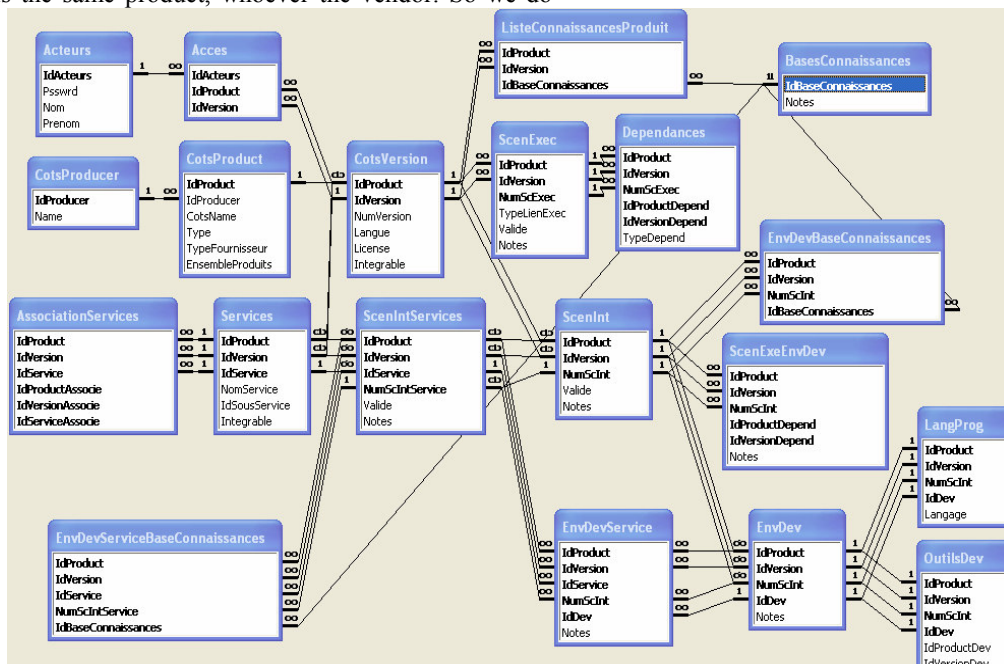


Figure 2: Entity/Association Schema for ICOTS

This diagram (see Figure 2) represents the IS database, and shows various information available on providers' web sites (for each product, which service is available)

COTS User

He has to fish for information, some offered by COTS providers. For those that are not easily retrieved, he will have to search for them with evaluation and tests realized with prototypes. This means that the "COTS User" himself will start the first implementation process activity (technological survey). He will initialize the communication process with FourCo and will go on communicating with it during all the application's development process. He can get some help from a new actor called "COTS Community".

COTS Community

It represents the set of developers with at least one COTS based application development experience. They acquired

information during the development steps and particularly during qualification, adaptation and implementation steps. The history of their development will help to compare estimates with real costs and risks of the COTS users managed project.

The COTS community turns out to be an information provider, just like the designers. Such actors play an important part, by enriching the reports associated with each COTS. All these information will be useful to other users, when they integrate the product.

Various actors can enrich the IS; first of all, the provider/consumer. Nevertheless, producers will be identified separately, whether they are COTS providers (with a high level of trust) or COTS Community.

Each actor can play the part of a provider or a consumer. If he participates to the production and enrich information, he can play the part of COTS community. He will capitalize information.

5. Environment to capture information

Getting information is essential for an interesting COTS based application development process. [2] prefers implementing prototypes essential for several development process steps and validation of information. With the help of such prototype, the studying of integration of one or several COTS together will become easier. Developers will get and validate with this study essential information for each step of the COTS based application development process. Such a use of prototypes in experimentation is an approach that will guarantee access to information that are either missing, or difficult to find.

Experimentation is often the only way to ensure the validity of collected information.

The following applications will show the benefits of such a method.

Presentation

The IS-COTS application will record the ID Card with information linked to the COTS product and to the COTS based application. This tool will also turn to be a big help to the client/provider relation: the ICOTS becomes available, and is open to new ICOTS downloading. IS-COTS enables the identification of all COTS Products'

services that are needed for the COTS based application. Particularly the interoperable services (those that are not interoperable are not interesting) Moreover, for each COTS and service, it provides an identification for execution and integration environments, in the purpose of helping the prototype's developer. I will also help him to input pre-requisites, development tools, code samples, and identified failures. Finally, applied to processing and information's production, IS-COTS will help targeting the set of softwares needed for the development, the execution of the prototype as well as possible failures in the prototype execution. The full date model is available at [4]. As a demonstration, will use two well known COTS Products: MS Outlook and MS Project in order to use respectively their contacts and their task management services.

The use of IS-COTS is divided in four parts: The first one (see Figure 3) is carried out by the COTS provider: it is the ICOTS pin, in our case Ms Outlook). The three following steps will be carried out by the user himself.

They are the ICOTS pin's creation within the prototype, and the association between services and production reports (cf. Figure 4).

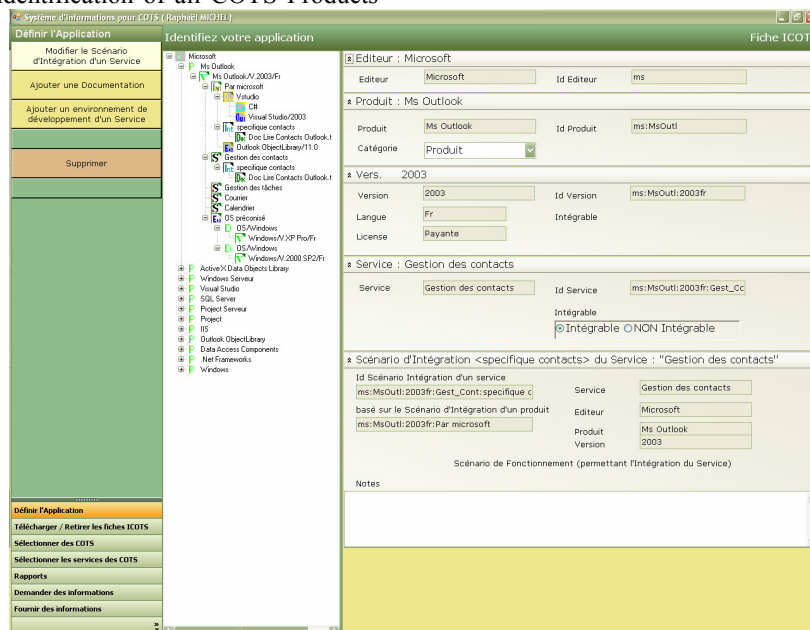


Figure 3: ICOTS (provider)

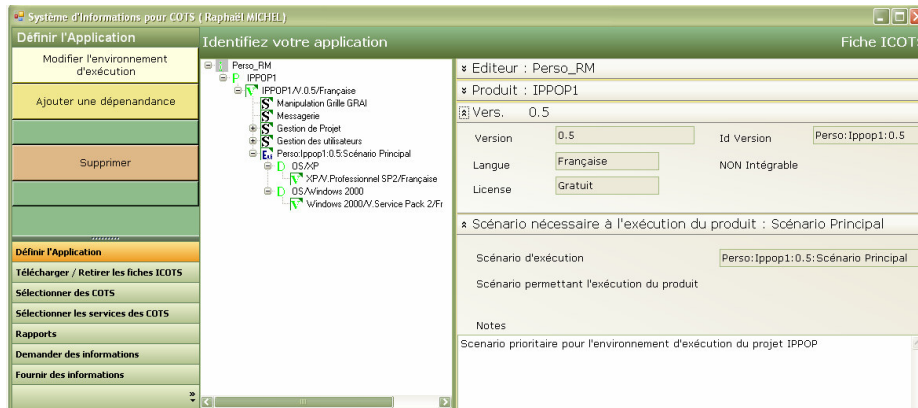


Figure 4: ICOTS (user)

6. Conclusion

The prototype has been implemented with .NET Technology for Windows. It is downloadable at the following URL (<http://www.iutbayonne.univ-pau.fr/~roose/pub/recherche/SICOTS>). It allows improving the collect of information, facilitates and accelerates the access to information for the integration. Moreover, it assists evaluations thanks to the reports and allows information capitalization by the COTS Products User Community. The information System implemented helps the development of prototypes for COTS Products based applications. This IS centered approach provide an ID Card for COTS, the ICOTS. It also provides information to help the integration of COTS Products.

Access to information is rare and precious. Therefore, it is important to capitalize the experience increased with developments. The application allows this task. Nevertheless, our approach has limits. The first one deals with interoperability problems for ICOTS. It needs a normalization of exchanges and data. Moreover, we have only addressed COTS Products with an integration point of view. Of course, this does not cover the wide spectrum of selection, evaluation, qualification, risks and costs steps.

7. Bibliography

[1] David J. Carney, Patricia A. Oberndorf, Patrick R.H. Plac - A Basis for an Assembly Process for COTS-Based Systems

(APCS) - Technical Report ; CMU/SEI-2003-TR-010, ESC-TR-2003-010 ; *May 2003*.

[2] Karl R.P.H. Leung, Hareton K.N. Leung - On the efficiency of domain-based COTS product selection method: Information and Software Technology pp 703-715; 44 (2002).

[3] Michel Raphaël, Roose Philippe, Barbier Franck - *Information System for Evaluation of COTS* - 3rd International Conference on Software Engineering Research, Management and Applications (SERA2005) - Central Michigan University, Mount. Pleasant, Michigan, USA - IEEE Computer Society 2005, ISBN 0-7695-2297-1, pp.64-69 - August 11 – 13 - 2005

[4] Michel Raphael – Information System for the Evaluation of COTS Products – PhD Thesis, UPPA, France, December 2006.

[5] Maurizio Morisio, Nancy Sunderhaft - Commercial-Off-The-Shelf (COTS): A Survey A DACS State-of-the-Art Report ; December 2000.

[6] ISO, 2004 Software engineering -- Product quality -- Part 4: Quality in use metrics - SO/IEC TR 9126-4:2004

[7] Egyed, A. Balzer, R. - Unfriendly COTS integration - instrumentation and interfaces for improved plugability - Automated Software Engineering, 2001. (ASE 2001), pp. 223-231, Marina del Rey, CA, USA, ISSN: 1527-1366 ISBN: 0-7695-1426-X.

[8] K. Vijayalakshmi, N. Ramaraj, R. Amuthakannan, S.M. Kannan - A new algorithm in assembly for component-based software using dependency chart- International Journal of Information Systems and Change Management (IJISCM), Vol. 2, No. 3, 2007.