INCO Project – Internal Report

Selected Literature on COTS products

Marco Torchiano
Rev. 6 - 9/3/2001 18:12

This is a partial and biased list of literature resources on COTS. It is incomplete because (save a few exceptions) I've read or at least browsed all the literature listed here. It is biased because it somewhat reflect my personal discovery path in this broad field.

1 Definitions and classifications

1.1 COTS Definitions

Source	Description
Federal Acquisition Regulations [1]	The acronym "COTS" stands for Commercial-Off-The-Shelf, so firstly we must define what is 'commercial', and what is 'off-the-shelf'. The official definition of the term "commercial" is given in the Federal Acquisition Regulations (FARs). A commercial item is: 1. Property customarily used for non governmental purposes and has been sold, leased, or licensed (or offered for sale, lease or license) to the general public; 2. Any item evolved from an item in (1) through advances in technology and is not yet available commercially but will be available in time to satisfy the requirement; 3. Any item that would satisfy (1) or (2) but for modifications customarily available in the commercial marketplace or minor modifications made to meet Federal Government requirements; 4. Any combination of items meeting (1) - (3) above; 5. Services for installation, maintenance, repair, training, etc. if such services are procured for support of an item in (1), (2), or (3) above, as offered to the public or provided by the same work force as supports the general public; or other services sold competitively in the marketplace;

	6. A non-developmental item developed exclusively at private expense and sold competitively to multiple state and local governments. As for the term "off-the-shelf", it can mean that the item is not to be developed by the user but is already existing. Such software can be used as • development tools (e.g., compilers); • integral parts of the new system (e.g., libraries); • both development tools and parts of the new system (e.g., DBMS, compilers with run-time libraries, OS with APIs).
SEI [18]	A COTS product is • sold, leased, or licensed to the general public; • offered by a vendor trying to profit from it; • supported and evolved by the vendor, who retains the intellectual property rights; • available in multiple, identical copies; and • used without source code modification.
COTS top-10 list [20]	Recently Basili and Boehm [Basili 2001] proposed another definition of COTS, which is different from the previous one. According to their definition, COTS software has the following characteristics: • The buyer has no access to the source code; • the vendor controls its development; and • it has a nontrivial installed base (that is, more than one customer; more than a few copies). This definition is more restrictive and does not take into account some types of software products like software products developed for special purposes and not widely deployed, special version of commercial software products and open source software.
Carney&Long [21]	A characterization of COTS products proposed by Carney and Long 2000, considers origin and modifiability of COTS and reports some examples. COTS products are classified according two attributes and placed into a two dimensional space. **Origin**'s possible values are: Independent Commercial Item, Special Version of Commercial Item, Component Produced by Contract, Existing Components from External Sources, Component Produced In-house. 'Special Version of Commercial Item' refers to a product developed by a commercial vendor and slightly modified for a client, where the modification may or may not be included in the next commercial release of the product. 'Component Produced by Contract' refers to subcontracting. 'Existing Components from External Sources' refers to components that are not developed internally, and usually not paid for either. **Required Modification** has five possible values: Extensive Reworking of Code,Internal Code Revision,Necessary Tailoring and Customization, Simple Parameterization, Very Little or no Modification. Two of them assume access to code (extensive

	reworking, internal code revision), two (necessary tailoring, parameterization) imply some mechanism built into the COTS to modify its functionality.

1.2 COTS-Based System

COTS are usually parts used to build larger systems. Here we change point of view, and we consider the system instead of the part. A COTS-system is a computer-based application that integrates one or more COTS.

Source	Description
Carney [22]	Carney identifies three types of COTS systems in function of the number of COTS used and their influence on the final system. • *Turnkey systems* are built around a (suite of) commercial products, such as Microsoft Office or Netscape Navigator. Only one COTS is used, and customization does not change the nature of the initial COTS. • *Intermediate systems* are built around one COTS (ex., Oracle) but integrate other components, commercial or developed in house. The central COTS is the main part of the system, but integration of other components is key. • *Integrated systems* are built by integrating several COTS, all on the same level of importance. The final system is not dominated by any single COTS, integration is the key to building the system.
Wallnau [23]	This classification of COTS-based systems identifies two different kind of systems: 1. COTS-solution systems: one substantial product (suite) is tailored to provide a "turnkey" solution. Main characteristics are: • Generic solutions, very tightly coupled to business process • Tailoring, parameterization focus • Vendor maintain • Take the vendor infrastructure • You adapt the tool 2. COTS-intensive systems: many integrated products provide system functionality. Main characteristics are: • Probably more flexible to business process change/creation • Integration, engineering focus • You maintain • Your own infrastructure • More complex to maintain

1.3 Architectural classification

Many works addressed the integration problem of COTS products, in particular the work by Yakimovich et al. [15] proposes a set of criteria used for classifying software architectures in order to estimate the integration effort.

The architectural properties adopted in such work are:

- component packaging
- type of control
- information flow
- synchronization
- component binding

Such characteristics are used to classify both the components and the system they are used in. The result is limited list of software system types classified according to these principles.
[Not Complete]

2 Selection

Summary of COTS selection approaches. The following table is an extension of the one found in [1].

PORE[4] Procurement Oriented Requirements Engineering	Elicitation of Features of Existing COTS software and requirements engineering are conducted in parallel Eventually a COTS software is selected that almost exactly fits the requirements
OTSO[5][6] Off-The-Shelf Option	Starting from a set of requirements specifying the system component, a decision taxonomy using AHP[7] and a set of measures is defined to select the most suitable COTS component in a given requirements context. The phases are screening on the full set of measures, ranking, detailed evaluation, cost and value estimation, and then the buy decision for a specific COTS software
CAP[1][3] COTS Acquisition Process	Process made up of three parts: Initialization, Execution and Reuse. The first part deals with the acquisition process planning and its cost estimation. Second part provides guidance for performing the COTS assessment (based on the AHP[7]) and taking the make-or-buy decision. The third part is responsible for storing all the information gathered by the other parts in order to decrease the cost of future COTS acquisition processes.
IusWare[8]	The methodology is based is based on the multicriteria decision aid approach and consists of two main phases: design of an evaluation model, application of the model. The design phases can be broken into: identification of relevant actor, identification of evaluation type, definition of a hierarchy of attributes, definition of the measures, choice of an aggregation technique.
[11] Scenario Based COTS Selection	An impact analysis of COTS is carried on considering system scenarios, they are modified under the hypothesis of using different COTS candidates, a new scenario set is produced together with a list of issued encountered during COTS adaptation
RCPEP [9]	This process consists of two phases: trade study, aimed at

Requirements-driven COTS product evaluation process	screening initial candidate products, and hand-on evaluation, which consists of an in-depth evaluation resulting in one (or more) recommended products.
[10] Risk Management Metrics	Some (mostly generic) metrics are proposed and their relation to the Cost of Software Quality (CoSQ) and to CMM maturity levels

3 Architectural mismatches

The idea of architectural mismatch was first used by Garlan et al. in a milestone paper in software architecture literature[12].
Yakimovich[15] propose a classification of architectural features that enable the evaluation of integration efforts.
Gacek[14] and Medvidovic[13] propose a methodology for evaluating the architectural impact of software components.
Such a method allows the selection of both the components and of a suitable architectural style. The key point is the identification of architectural mismatches.
In [13] a synthesis and analysis approach is proposed in order to keep the decision taken during the development process consistent with the models used during the initial architectural assessment phase.
A sort of unification of the ideas presented in [15] and [14] is proposed in [16].

4 Development process

The development process should be changed in presence of COTS components.

Source	**Description**
Morisio et al. [17]	Report about adopted COTS based processes and proposal of a new COTS based process
SEI [18]	Basic activities and their classification under the perspective of defining a COTS based process

5 Cost models

A discussion of the economical issues related to COTS adoption is presented in [28].
One model currently under development and calibration: COCOTS[19]

6 Other

A short overview of problems using COTS can be found in [26].
Another overview of problems is presented in [29].

The problem of assessment and certification of suitability of COTS is described in [27].

The COTS products addressed by such definition present some specific non-technical problems, related to the quick turnaround (every 8-9 month)[20] of products releases. In addition marketplace consideration add further variability: in the COTS products market there are no standardized measures [24] mainly due to marketing strategies aimed at obtaining vendor lock-in. Variability and marketing strategies suggests that there will never be a single unified marketplace of standardized COTS products [25], thus becomes more and more important to have a clear understanding of multifaceted world of COTS.

7 References

[1] Oberndorf, T., COTS and Open Systems - An Overview, 1997, available at http://www.sei.cmu.edu/str/descriptions/cots.html#ndi

[2] Ochs, M.A.; Pfahl, D.; Chrobok-Diening, G.; Nothhelfer-Kolb, B. “A Method for Efficient Measurement-based COTS Assessment ad Selection – Method Description and Evaluation Results” in Proceedings IEEE 7th International Software Metrics Symposium, London, England , 4-6 April 2001. pp. 285-296

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

[4] N.Maiden, C.Ncube, “Acquiring COTS Software Selection Requirements”, IEEE Software March/April 1998, pp 46-56.

[5] Jyrki Kontio: "OTSO: A Systematic Process for Reusable Software Component Selection", University of Maryland, Technical report, December 1995.∈

[6] J. Kontio “A Case Study in Applying a Systematic Method for COTS Selection”, Proc. 18th ICSE, IEEE, 1996.

[7] T.Saaty. “The Analytic Hierarchy Process”, McGraw-Hill, New York, 1990.

[8] Morisio M., Tsoukiàs A., IusWare: A methodology for the evaluation and selection of software products, IEE Proceedings Software Engineering, June 1997, pp. 162-174.

[9] P.Lawlis, K.Mark, D.Thomas, T.Courtheyn. "A Formal Process for Evaluating COTS Software Products". IEEE Computer, Vol. 34, No. 5, May 2001.

[10] S.Sedigh-Ali, A.Ghafoor, R.Paul. "Software Engineering Metrics for COTS-Based Systems", IEEE Computer, Vol. 34, No. 5, May 2001, pp. 44-50

[11] M. D. Feblowitz, S. J. Greenspan "Scenario-Based Analysis of COTS Acquisition Impacts" In Requirements Eng 3(3/4), 1998, pp 182-201

[12] D. Garlan, R. Allen, and J. Ockerbloom: "Architectural Mismatch or Why it's hard to build systems out of existing parts", Proc. International Conference on Software Engineering (ICSE'95), Seattle, WA, USA, 1995, p. 179-185

[13] Egyed, A.; Medvidovic, N.; Gacek, C. "Component-based perspective on software mismatch detection" In IEE Proceedings-Software, Volume: 147 Issue: 6, December 2000

[14] Gacek, C., “Detecting Architectural Mismatches During Systems Composition”, USC Technical report, USC/CSE-97-TR-506, July 1997.

[15] D.Yakimovich, J.Bieman, V.Basili: "Software architecture classification for estimating the cost of COTS integration", Proc. 21st International Conference on Software Engineering, Los Angeles, USA, 1999, p. 296-302.

[16] D.Yakimovich: "A comprehensive reuse model for COTS software products", Ph.D. Thesis Proposal, Computer Science Department, Univ. Maryland, 2001

[17] M. Morisio, C. Seaman, A. Parra, V. Basili, S. Condon, and S. Kraft, "Investigating and Improving a COTS-Based Software Development Process", Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000) , Limerick, Ireland, June 2000.

[18] L.Brownsword, T.Oberndorf, C.Sledge. "Developing New Processes for COTS-Based Systems", IEEE Software July/August 2000, pp. 48-55

[19] USC-CSE-2000-501. “COCOTS: A COTS Software Integration Lifecycle Cost Model - Model Overview and Preliminary Data Collection Findings”, Computer Science Department, USC Center for Software Engineering

[20] V.Basili, B.Boehm. "COTS-Based Systems Top 10 List" IEEE Computer, Vol. 34, No. 5, May 2001

[21] David Carney and Fred Long: “What Do You Mean by COTS? Finally a Useful Answer”, IEEE Software March/April 2000, p. 83-86.

[22] Carney, D. Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities. SEI Monographs on Use of Commercial Software in Government Systems, Software Engineering Institute, Pittsburgh, USA, June 1997.

[23] K.Wallnau, D.Carney, B.Pollak. “How COTS Software Affects the Design of COTS-Intensive Systems”. In SEI Interactive, 6/98, 1998. Available online at http://interactive.sei.cmu.edu/Features/1998 /June/cots_software/Cots_Software.htm

[24] J.Voas. "Faster, better, cheaper", IEEE Software, May/June 2001pp. 96-97

[25] K.Wallnau. "On Software Components and Commercial ('COTS') Software". In Proceedings of 1999 International Workshop on Component-Based Software Engineering, Los Angeles, CA, USA, May 17-18, 1999

[26] J.Voas. "The Challenges of Using COTS Software in Component-Based Development", IEEE Computer, Vol. 31, No. 6; JUNE 1998, pp. 44-45

[27] J.Voas. "Certifying Off-the-Shelf Software Components", IEEE Computer, Vol. 31, No. 6; JUNE 1998, pp. 53-59

[28] J.Voas. "COTS Software: The Economical Choice?", IEEE Software Vol. 15, No. 2; MARCH/APRIL 1998, pp. 16-19

[29] B.Boehm, C.Abts. "COTS Integration: Plug and Pray?", IEEE Computer, Vol. 32, No. 1; JANUARY 1999, pp.135-138