COTS Selection: Past, Present, and Future

Abdallah Mohamed[1], Guenther Ruhe[1], Armin Eberlein[2]
[1]University of Calgary, 2500 University Drive, NW, Calgary, AB, T2N1N4, Canada
[2]American University of Sharjah, Sharjah, P.O. Box 26666, UAE
{asamoham, ruhe, eberlein}@ucalgary.ca

Abstract

Commercial Off-The-Shelf (COTS) products are increasingly being used in software development. In COTS-based development, selecting appropriate COTS is the most crucial phase. This paper explores the evolution of COTS selection practices, and surveys eighteen of the most significant COTS selection approaches. The paper traces how each approach contributed to the improvement of current COTS selection practices, and then compares them .The paper also highlights some open research issues relevant to the selection process, and concludes with a discussion of possible future directions to address these issues.

1. Introduction

Most of today's software systems include one or more COTS (commercial off-the-shelf) products. COTS are pieces of software that can be reused by software projects to build new systems [1, 2]. Such COTS include word processors, email packages, etc [3].

Performing a good COTS selection plays a critical role in the success of the final system [4]. COTS selection is the process of determining the fitness-of-use of COTS products in a new context, and then selecting one or more products with the highest fitness [5].

COTS selection involves many challenges such as the high complexity of the process [6]. To overcome these challenges, several efforts were made during the last decade to model the COTS selection process. However, none of these methods can be considered as the silver-bullet to solving the COTS selection problem. Different approaches are of different effectiveness and might be suitable for different contexts.

In this paper, we show how different COTS selection approaches contributed to the advance of COTS selection practices. The paper reviews current COTS selection approaches, discusses their contribution, compares them, shows their pros and cons, and briefly explains their main activities. Next, the paper highlights those issues that were not sufficiently addressed by existing approaches, and shows possible future directions that can be used to solve these issues. To the best of our knowledge, there is no previous such thorough surveys.

This paper is structured as follows: Section 2 describes the COTS selection process. The section starts by describing the so-called "*general COTS selection process*" which highlights the main activities that most existing approaches use. Then, we discuss the evaluation process which is the core of any COTS selection approach. Section 3 presents COTS selection approaches and summarizes their contribution to the evolution of COTS selection practices. Section 4 presents some of the 'present' open issues relevant to the COTS selection process. Then, in section 5 some 'future' directions hat can be taken to address these issues are discussed. Finally, conclusions are given in section 6. The paper includes one appendix which describes the approaches listed in section 3.

2. The COTS selection process

2.1. The general COTS selection process

Although there is no commonly accepted method for COTS selection [6], all methods share some key steps that can be iterative and overlapping. These steps formulate what we refer to as the *General COTS Selection (GCS)* process which is described as follows:

Step$_1$: Define the evaluation criteria based on stakeholders' requirements and constraints.

Step$_2$: Search for COTS products.

Step$_3$: Filter the search results based on a set of 'must-have' requirements. This results in defining a short list of most promising COTS candidates which are to be evaluated in more detail.

Step$_4$: Evaluate COTS candidates on the short list

Step$_5$: Analyze the evaluation data (i.e. the output of *Step*$_4$) and select the COTS product that has the best fitness with the criteria. Usually, decision making techniques, e.g. analytic hierarchy process (AHP) [7], are used for making the selection.

After *Step*$_5$, the selected COTS product is usually customized (aka tailored) as needed in order to reduce the mismatches it has with the requirements. A COTS product can be customized in different ways, such as using add-ons, adjusting parameters, etc.

2.2. COTS evaluation

COTS evaluation is the core of the COTS selection process as it is the activity in which the fitness-of-use

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

of COTS products is determined. COTS evaluation provides information necessary for the decision maker to select the best COTS product from a set of competing alternatives [8, 9].

COTS are evaluated against a set of criteria that represents the stakeholders' requirements and system constraints. Kontio et al. [10] suggest to hierarchically define the evaluation criteria, where a set of high-level goals are gradually refined based on such factors as the application requirements and architecture, the COTS capabilities, etc. Maiden et al. [4] agrees with Kontio and further suggest defining the evaluation criteria while at the same time evaluating existing COTS. The technical literature also includes efforts to explain how to define the evaluation criteria based on quality models. For example, Franch et al. [11] and Carvallo et al [12] propose a six-step method to build a structured quality model for the purpose of COTS evaluation. They rely on the ISO/IEC 9126 quality model [13] and further explain activities to define a set of metrics that can be used during the evaluation.

Generally, there are three strategies which can be applied to evaluate COTS products [14, 15]:

1. *Progressive filtering,* which starts with a large number of COTS, and then progressively defines discriminating criteria through successive iterations of product evaluation cycles, where in each cycle 'less fit' products are eliminated. This strategy requires running steps 1 to 4 in the GCS process iteratively until a small number of most promising COTS products is identified from which one or more can be selected for integration into the system.
2. *Puzzle assembly*, which assumes that a COTS-based system requires fitting various components together like pieces of a puzzle. This implies that a product that 'fits' in isolation might not be acceptable when combined with other products. Therefore, this strategy suggests considering the requirements of each product while simultaneously remembering the requirements of other products in the puzzle.
3. *Keystone identification*, which starts by identifying a key requirement (e.g. vendor location or type of technology), and then searching for products that satisfy this keystone requirement. This allows quick elimination of a large number of products that do not satisfy the key requirement.

More than one of the above strategies may be employed in the same project [15]. For example, a developer might use 'keystone identification' first and then later 'progressive filtering'.

3. COTS selection approaches

This section surveys and compares current COTS selection approaches. The section tries to trace how each approach contributed to the improvement of current COTS selection practices. Each approach is discussed in more detail in Appendix 1.

3.1. The evolution of COTS selection practices

Figure 1 traces the improvements made to the COTS selection process over the last decade. One of the first proposals was given by Kontio et. al. [16] who proposed the OTSO (Off-The-Shelf Option) approach for COTS selection in 1995. OTSO is considered an important milestone in the evolution of COTS selection practices as it served as a basis for other approaches. OTSO defined the basic structure of the COTS selection process. This structure is very similar to the GCS process described in section 2.1.

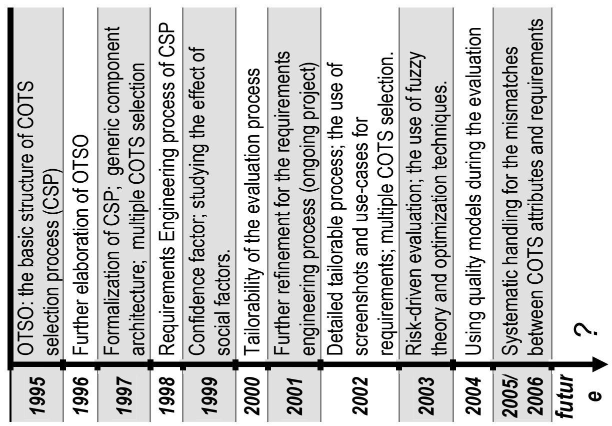

Figure 1 Evolution of COTS selection practices

In 1996, Kontio published several follow-up papers to elaborate OTSO (e.g. [10]) in which the process of defining the evaluation criteria is described in detail.
In 1997, several approaches were proposed:
(i) The IusWare (IUStitia softWARis) [17] approach tried to formalize the selection process, and to address quality requirements during the evaluation process.
(ii) The PRISM (Portable, Reusable, Integrated, Software Modules) [18] approach proposed a generic component architecture that can be used during the COTS evaluation process.
(iii) The CISD (COTS-based Integrated Systems Development) [19] approach proposed a model that can be used to select multiple homogeneous COTS products.

However, it was not until 1998 that another important milestone was reached with the PORE approach [4]. The importance of PORE is that it proposed a requirements engineering process for COTS-based development. PORE suggested that requirements should be elicited and analyzed at the same time when the COTS products are evaluated.

In 1999, several approaches were proposed:
(i) The CEP (Comparative Evaluation Process) approach introduced the use of the so-called confidence factor (CF). The more reliable the source of data, the higher a CF value that source gets. Any estimate we make should be adjusted based on the CF value of the source based on which these estimations are made.

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

(ii) The STACE (Social-Technical Approach to COTS Evaluation) approach [20] emphasized the importance of non-technical issues, e.g. social, human, and organizational characteristics, during the evaluation process.
(iii) The CRE (COTS-based RE) approach emphasized the importance of non-functional requirements (NFR) as decisive criteria when comparing COTS alternatives. CRE uses the NFR framework [21] to model NFRs.

In 2000, Ochs et al. [22] proposed the COTS acquisition process (CAP) which emphasized the concept of a "tailorable evaluation process". This means the evaluation process (including the evaluation criteria themselves) should be tailored based on the available effort for the project. Ochs' approach relied on experts' knowledge to tailor the process.

In 2001, a project was started by Chung et al. to define a COTS-Aware Requirements Engineering (CARE) approach [23-26]. CARE uses a flexible set of requirements based on different agents' views. For this, CARE proposes a method to define relevant agents as well as the system goals and requirements.

In 2002, another set of approaches were proposed:
(i) the PECA (Plan, Establish, Collect, and Analyze) approach [27] from SEI [5] describes a detailed tailorable COTS selection process and gives guidelines which the experts can use to tailor the process.
(ii) The BAREMO approach explained in detail how a decision can be made based on the AHP [7] method.
(iii) The storyboard approach [28] suggests to incorporate use-cases and screen-captures during the requirements engineering phase to help customers understand their requirements, and thus acquire more appropriate COTS products.
(iv) The combined selection approach [29] tries to select multiple COTS that are evaluated, first on the local scale to evaluate each COTS in isolation from the others, and then on the global scale to select the best combination of COTS.

In 2003, two more approaches were proposed:
(i) The WinWin spiral model [3] which is a risk-driven approach that suggests to identify, analyze and resolve risks in an iterative evaluation process.
(ii) The approach by Erol et. Al. [30] suggests the use of fuzzy theory to quantify qualitative data, and then to use optimization techniques to determine optimal (or near optimal) solutions.

In 2004, the DesCOTS system [31] was presented based on the work done in [11]. This system integrates several tools to define evaluation criteria using quality models such as ISO/IEC9126 [13].

In 2005, the MiHOS (Mismatch-Handling aware COTS Selection) approach was developed [32]. MiHOS relies on the GCS method and introduces a process for handling mismatches between COTS products and requirements. MiHOS uses techniques such as linear programming to identify near optimal solutions.

3.2. Surveying COTS selection approaches

Appendix 1 describes the eighteen approaches mentioned in section 3.1 in detail. Table 1 compares these approaches in terms of the following criteria:

1) GCS: Conformance to the *GCS* method.
2) EVAL: Evaluation strategy used.
3) SNG: Suitability for single COTS selection.
4) MLT: Suitability for multiple COTS selection.
5) MISM: Ability to address COTS mismatches in a systematic way during/after the selection process.
6) TAILOR: Tailorability of the process based on experts' knowledge. Satisfying this criterion does not necessarily imply the existence of any systematic tailoring techniques.
7) TS: Availability of tool support to facilitate the application of the approach.

Table 1 Comparing COTS selection approaches

APPROACH			COMPARISON FACTORS						
ID	Name	Year	GCS	EVAL	SNG	MLT	MISM	TAILOR	TS
A1	OTSO	95/96	✓	PF	✓	×	×	×	×
A2	IusWare	1997	✓	Any	✓	×	×	×	×
A3	PRISM	1997	✓	PF	✓	×	×	×	×
A4	CISD	1997	✓	PF/PZ	✓	~	×	×	×
A5	PORE	1998	✓	PF	✓	×	~	×	✓
A6	CEP	1999	✓	PF	✓	×	×	×	×
A7	STACE	1999	✓	KS/PF	✓	×	~	×	×
A8	CRE	1999	✓	PF	✓	×	~	×	×
A9	CAP	2000	✓	PF	✓	×	×	✓	×
A10	CARE	2001	✓	PF	✓	×	~	×	✓
A11	PECA	2002	✓	Any	✓	×	~	✓	×
A12	BAREMO	2002	step5	-	✓	×	×	×	✓
A13	Storyboard	2002	×	KS/PF	~	✓	~	×	×
A14	CS	2002	✓*	-	~	✓	~	×	×
A15	WinWin	2003	✓	PF	✓	~	~	×	×
A16	Erol's	2003	✓	-	✓	×	×	×	×
A17	DesCOTS	2004	✓	PF	✓	×	×	×	✓
A18	MiHOS	2005	✓	KS/PF	✓	×	✓	×	✓

PF: *Progressive filtering* — ✓ *fully satisfies the criterion*
KS: *Keystone* — × *does not satisfy the criterion*
PZ: *Puzzle assembly* — ~ *partially,informally,or implicitly satisfies the criterion*

**CS uses other approaches, e.g. OTSO, for local COTS evaluation.*

4. Evaluating existing approaches

In this section, we list some of the open research issues that have not been sufficiently addressed by existing COTS selection approaches.

As we have seen in section 3, there is a great variety of approaches to tackle the COTS selection problem. Practitioners have two main requests: (i) they are wondering about the effectiveness of the different approaches, and (ii) they need guidance on when and how to choose from these approaches. In his paper [1], R. Glass stated a clear message from software practitioners to software researchers: "*What help do practitioners need? We need some better advice on how and when to use methodologies*". This brings us to the first open research issue:

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

$\mathfrak{R}_1$: "*To support selection of appropriate and effective methods for our context*"

Furthermore, although most of the proposed approaches were developed for general use, there is no commonly accepted approach for COTS selection [6]. Also, these approaches were proposed without a clear explanation of how they can be adapted to different domains and projects. This means, even if we can satisfy $\mathfrak{R}_1$ and identify an approach that best-fits a specific context, customizing this approach to better fit the context would be problematic. This leads to the second open research issue:

$\mathfrak{R}_2$: "*To show how COTS selection approaches can be adapted to fit into different contexts*"

The next open issue is more relevant to the decision making during the selection process. Current approaches suggest using decision making techniques such as weighted score method (WSM) or analytic hierarchy process (AHP) [7]. However, there are several limitations to these techniques [33]. For example, these techniques estimate the fitness of COTS candidates based on 'one' total fitness score. This is sometimes misleading due to the fact that high performance in one COTS aspect might hide poor performance in another aspect. This leads to the third open research issue:

$\mathfrak{R}_3$: "*To use more informative decision making techniques*"

A follow up on the above problem is that, current decision making techniques usually suggest 'one' optimal solution to the decision makers who can either accept or reject it. This limits the decision makers' participation in and control over the decision making process. An alternative is to use decision support techniques. Decision support is a way of helping decision makers under different levels of uncertainty, complexity, and changing problem parameters [2]. Decision support suggests that instead of giving only one optimal solution, decision makers should be given several optimal or near optimal solutions from which they can either accept one or re-run the process again after adjusting its parameters until they reach a reasonable solution. This allows decision makers to have more participation in and control over the decision making process. This leads to the fourth open research issue:

$\mathfrak{R}_4$: "*To use decision support techniques in order to help making more efficient decisions*"

The next research issue is related to having different stakeholders with different views. Currently, this issue is resolved implicitly or informally. However, what is needed is a more robust negotiation component through which COTS can be progressively selected based on functional and non-functional requirements, architecture, and at the same time resolving conflicts between stakeholders. This leads to the fifth issue:

$\mathfrak{R}_5$: "*To provide a negotiation component for resolving conflicts between stakeholders*"

The remaining issues are related to the COTS market and their vendors. Most current approaches assume that the knowledge required for evaluating COTS products is available and reliable. Current approaches (e.g. [4]) suggest that COTS evaluation should initially rely on vendors' documentations. Then, for specific requirements, we should rely on vendor-led demonstration and on user-led experimentation. Nevertheless, the problem here is three-fold: (i) searching for and collecting the knowledge might very effort consuming. This leads to another open research issue:

$\mathfrak{R}_6$: "*To design knowledge repositories and show how they can be used for evaluating COTS*".

(ii) Some vendors are more reliable than others, and hence their documents can be used during the evaluation more broadly in order to reduce the evaluation effort related to user-led experimentations. This leads to another open research issue:

$\mathfrak{R}_7$: "*To develop techniques to evaluate vendors' credibility, and adjust the COTS evaluation results (obtained from vendors' documentations) based on such credibility estimation*".

5. An outlook

The seven issues described in section 4 can be analyzed for improving COTS selection practices. In this section, we show possible directions that can be used to address these eight issues.

For $\mathfrak{R}_1$: A high-level roadmap was given in [1] to support selecting appropriate methods for different contexts. The main idea is to taxonomize existing methods, to categorize problem domains in terms of what they need, and finally to find a way to map these two categories to each other. In addition, further confirmative studies should be conducted to analyze the effectiveness of the wide variety of approaches.

For $\mathfrak{R}_2$: Initial efforts [34, 35] developed a framework for customizing the COTS selection process. The main idea is to define a set of possible options for each activity during the process, and then give support to select the best option based on such project characteristics as effort, criticality, etc. Another useful reference can be found in [36]. Although this reference is not COTS oriented, it shows one method of process tailoring.

For $\mathfrak{R}_3$: The evaluation criteria can be defined in a hierarchical manner using goal-oriented definition [37]. Some COTS selection approaches already use goals [31]. However, it is not clear yet how to estimate the satisfaction of each goal. For this purpose, the relationships between goals should be quantified in terms of how each goal contributes to or prevents the satisfaction of other goals. The work done in [38] can be used as a reference for this purpose.

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

For $\Re_4$: There are many possible sources that explain the decision support concept [39-41]. An initial effort was done in [32] to incorporate decision support techniques in the COTS selection process.

For $\Re_5$: Negotiation components are essential parts of decision support systems [6]. Group decision making techniques [43, 44] might be very useful here. Also, agent-based systems can be employed to support the same purpose [45].

For $\Re_6$: The concept of learning software organization (LSO) [39] can be useful to help designing the required knowledge repositories. An initial effort was proposed in [46] where a framework for evaluating COTS with the support of knowledge bases is described. Also, there are useful online repositories for certain domains, e.g. [47]. However, more repositories are needed to cover the large spectrum of COTS domains.

For $\Re_7$: The work proposed by the approach A6 [48] (see Appendix-1) shows how to adjust evaluation results based on the credibility of the data source. However, different vendors should have different credibility factors which could be estimated using technique that can handle uncertain qualitative information, e.g. Bayesian Networks [49].

6. Conclusions

In this paper, we explored the evolution of COTS selection practices, and we surveyed and compared 18 of the most significant COTS selection approaches. In spite of the great variety of these approaches, there still many open issues related to the COTS selection process that need further research. We have listed some of these issues and showed some possible venues that could be taken to address them.

7. References

[1] D. G. Firesmith, "Achieving Quality Requirements with Reused Software Components: Challenges to Successful Reuse," in *MPEC'05*, St. Louis, Missouri, 2005.

[2] M. R. Vigder, W. M. Gentleman, & J. Dean, "COTS Software Integration: State of the art," NRC, 39198, Jan 1996.

[3] B.Boehm, D.Port, and Y.Yang,"WinWin Spiral Approach to Developing COTS-Based Applications,"*EDSER-5*,Oregon,2003.

[4] N.A.Maiden & C.Ncube, "Acquiring COTS Software Selection Requirements,"*IEEE Software,* vol.15(2),1998.pp. 46-56.

[5] SEI: Software Engineering Institute in Carnegie Mellon Univ, at http://www.sei.cmu.edu.

[6] G. Ruhe, "Intelligent Support for Selection of COTS Products,", *LNCS, Springer,* vol. 2593 2003. pp. 34-45

[7] T. L. Saaty, *The analytic hierarchy process*. New York: McGraw-Hill, 1990.

[8] D. Carney, "Evaluation of COTS Products: Some Thoughts on the Process," SEI Interactive, CMU university, Sept 1998,

[9] D. Carney, "COTS Evaluation in the Real World," SEI Interactive, Carnegie Mellon University, Dec 1998,

[10] J. Kontio, G. Caldiera, and V. R. Basili, "Defining factors, goals and criteria for reusable component evaluation," in *CASCON'96* Toronto, Ontario, Canada: IBM Press, 1996.

[11] X.Franch & J.P.Carvallo, "Using Quality Models in Software Package Selection,"*Software,*vol.20(1),Jan/Feb 2003. pp. 34-41

[12] J.P.Carvallo,X.Franch,G.Grau,&C.Quer,"COSTUME: a method for building quality models for composite COTS-based software systems," in *QSIC'04*, Germany, 2004, pp. 214 - 221

[13] ISO/IEC1926-1,*Software Engineering - Product Quality Model -Part1:Quality Model*. Geneve, Switzerland: ISO, 2001.

[14] D. Kunda and L. Brooks, "Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection: A Case Study," *ECIS'00*, Austria, 2000.

[15] P. A. Oberndorf, L. Brownsword, E. Morris, and C. Sledge, "Workshop on COTS-Based Systems," SEI Institute, CMU, Special Report CMU/SEI-97-SR-019, Nov. 1997,

[16] J. Kontio, "OTSO: A Systematic Process for Reusable Software Component Selection," University of Maryland, Maryland, CS-TR-3478, December 1995,

[17] M. Morisio and A. Tsoukis, "IusWare: a methodology for the evaluation and selection of software products," *IEE Software Engineering* vol. 144 (3), June 1997.

[18] R. W. Lichota, R. L. Vesprini, and B. Swanson, "PRISM: Product Examination Process for Component Based Development," *SAST '97*, 1997, pp. 61-69.

[19] V.Tran & D.Liu,"A Procurement-centric Model for Engineering Component-based Software Systems," *SAST '97*, 1997,pp.7079.

[20] D. Kunda & L.Brooks, "Applying social technical approach for COTS selection,"*UKAIS'99,* Univ of York, McGraw Hill, 1999.

[21] L.Chung, B.A.Nixon, E.Yu, and J.Mylopoulos, *Non-Functional Requirements in Software Engineering*: Kluwer Academic, 99.

[22] M.Ochs, D.Pfahl, G.Chrobok-Diening, and B. Nothhelfer-Kolb, "A COTS Acquisition Process: Definition and Application Experience," *ESCOM-SCOPE'00*, Munich, Germany, 2000.

[23] L. Chung and K. Cooper, "A COTS-Aware Requirements Engineering (CARE) Process: Defining System Level Agents, Goals, and Requirements," Department of Computer Science, The University of Texas, Dallas, TR UTDCS-23-01, Dec 2001.

[24] L. Chung and K. Cooper, "Knowledge-based COTS-aware requirements engineering approach,"*SEKE'02,*Ischia,Italy, 2002.

[25] L. Chung and K. Cooper, "Defining Goals in a COTS-Aware Requirements Engineering Approach," *Systems Engineering journal,* vol. 7 (1), 2004, pp. 61-83.

[26] L. Chung and K. Cooper, "Matching, Ranking, and Selecting COTS Components: A COTS-Aware Requirements Engineering Approach," in *MPEC'04,* Scotland, UK, 2004.

[27] S. Comella-Dorda, J. C. Dean, E. Morris, and P. Oberndorf, "A Process for COTS Software Product Evaluation," in *ICCBSS'02*, Orlando, Florida, 2002, pp. 86 - 96.

[28] S.Gregor, J.Hutson, & C.Oresky, "Storyboard Process to Assist in Requirements Verification and Adaptation to Capabilities Inherent in COTS," in *ICCBSS'02*, Florida, 2002, pp. 132-141.

[29] X. Burgues, C. Estay, X. Franch, J. A. Pastor, and C. Quer, "Combined Selection of COTS Components," in *ICCBSS'02*, Orlando, Florida, 2002, pp. 54-64.

[30] I. Erol and W. G. Ferrell-Jr., "A methodology for selection problems with multiple, conflicting objectives and both qualitative and quantitative criteria " *International Journal of Production Economics* vol. 86 (3), Dec 2003. pp. 187-199

[31] G. Grau, J. P. Carvallo, X. Franch, and C. Quer, "DesCOTS: A Software System for Selecting COTS Components," in *EUROMICRO'04*, Rennes, France, 2004, pp. 118-126.

[32] A. Mohamed, G. Ruhe, and A. Eberlein, "Decision Support for Handling Mismatches between COTS Products and System Requirements," in *ICCBSS'07,* Banff, Canada, 2007.

[33] C. Ncube and J. C. Dean, "The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components," in *ICCBSS 2002*, 2002, pp. 176-187.

[34] A. Mohamed, G. Ruhe, and A. Eberlein, "Towards a Customizable Approach for COTS Selection," in *SEA2004*, Cambridge, MA, USA, 2004, pp. 665-671.

[35] A. Mohamed, G. Ruhe, and A. Eberlein, "Customization of COTS selection Process," SEDS Lab, UofC, Calgary, TR042/2005, *http://www.enel.ucalgary.ca/~asamoham/SEDS/TR042-2005-Short.pdf.*

[36] L. Jiang, A. Eberlein, and B. H. Far, "Methodology for Requirements Engineering Process Development," in *ECBS'04* Brno, Czech Republic, 2004.

[37] A. V. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," in *5th IEEE International Symposium on Requirements Engineering*, 2001, pp. 249-263.

[38] H. Kaiya, H. Horai, and M. Saeki, "AGORA: attributed goal-oriented requirements analysis method," in *RE'02*, Essen. Germany, 2002, pp. 13- 22.

IEEE COMPUTER SOCIETY

[39] G. Ruhe, "Software Engineering Decision Support - A New Paradigm for Learning Software Organizations," *Lecture Notes in Computer Science,* vol. 2640 Nov 2003. pp. 104-113
[40] G. Ruhe, A. Eberlein, and D. Pfahl, "Quantitative WinWin: a new method for decision support in requirements negotiation," in *SEKE'02*, Ischia, Italy, 2002, pp. 159-166.
[41] G. Ruhe and A. Ngo-The, "Hybrid Intelligence in Software Release Planning," *International Journal of Hybrid Intelligent Systems,* vol. 1 (2), 2004. pp. 99-110
[43] C. L. Hwang and M. J. Lin, *Group Decision Making under Multiple Criteria-Methods and Applications* Springer-Verlag, Berlin, Heidelberg, New York, NY, 1987.
[44] F. Herrera, E. Herrera-Viedma, and J. Verdegay, "A Model of Consensus in Group Decision Making under Linguistic Assessments," *Fuzzy Sets and Systems,* vol. 78 1996. pp. 73-87
[45] T. Wanyama and B. H. Far, "A Multi-Agent Framework for Conflict Analysis and Negotiation: Case of COTS selection," *Transactions of Institute of Electronics, Information and Communication IEICE, Special Issue of Software Agent and its Applications,* vol. E88-D (9), Sept. 2005. pp. 2047-2058
[46] A. Mohamed, T. Wanyama, G. Ruhe, A. Eberlein, and B. Far, "COTS Evaluation Supported by Knowledge Bases," *Lecture Notes in Computer Science,* vol. 3096 2004/09 2004. pp. 43-54
[47] CMS-Matrix, "Avail. at http://www.cmsmatrix.org."
[48] B. C. Phillips and S. M. Polen, "Add Decision Analysis to Your COTS Selection Process," *The Journal of Defense Software Engineering,* April 2002 2002.
[49] F. V. Jensen, *Introduction to Bayesian Networks*. NewYork: Springer-Verlag, 1996.
[50] J. Kontio, "A Case Study in Applying a Systematic Method for COTS Selection," in *18th International Conference on Software Engineering*, Berlin, Germany, 1996, pp. 201-209.
[51] J. Kontio and R. Tesoriero, "A COTS selection method and experiences of its use," in *The Twentieth Annual Software Engineering Workshop* Greenbelt, Maryland, 1995.
[52] C. Ncube and N. Maiden, "Guiding parallel requirements acquisition and COTS software selection," in *IEEE International Symposium on Requirements Engineering*, University of Limerick, IRELAND, 1999, pp. 133 - 140.
[53] C. Ncube and N. A. Maiden, "PORE: Procurement-Oriented Requirements Engineering Method for the Component Based Systems Engineering Development Paradigm," *Intr. Workshop on Component Based Software Eng.,* Los Angeles, CA, 1999.
[54] B. Kitchenham, "DESMET: A method for evaluating Software Engineering methods and tools," Department of Computer Science, University of Keele, UK, TR96-09, 1996,
[55] D. Kunda, "A social-technical approach to selecting software supporting COTS-Based Systems," PhD Thesis, Department of Computer Science, University of York, Oct 2001
[56] S. Kunda, "A social-technical approach to selecting software supporting COTS-based systems," Thesis, Univ. of York, 2001
[57] C. Alves and J. Castro, "CRE: A Systematic Method for COTS Components Selection," *SBES'01* Rio de Janeiro, Brazil, 2001.
[58] S. Comella-Dorda, J. C. Dean, G. Lewis, E. Morris, P. Oberndorf, and E. Harper, "A Process for COTS Software Product Evaluation," CMU Univ., SEI Institute, CMU/SEI-2003-TR-017, July 2004.
[59] A. Lozano-Tello and A. Gomez-Pérez, "BAREMO: how to choose the appropriate software component using the analytic hierarchy process," in *SEKE'02*, Italy, 2002, pp. 781-788.
[60] ExpertChoice-homepage:: http://www.expertchoice.com.
[61] Y. Yang, J. Bhuta, B. Boehm, D. Port, and C. Abts, "Composable Process Elements for Developing COTS-Based Applications," in *EDSER-5*, Portland, Oregon, 2003.
[62] B. Boehm, "A Spiral Model of Software Development and Enhancement," *Computer,* vol. 21 May 1988. pp. 61-72
[63] Y. Akao, *Quality Function Deployment: Integrating Customer Requirements into Product Design*. Cambridge, MA: Productivity Press, 1999.

Appendix-1: Current COTS Selection Approaches in a Nutshell

This appendix summarizes current COTS selection approaches. The approaches are described in tables 2 to 19. For each approach, the following items are described:

- The main idea that distinguishes this approach from other approaches.
- The main steps of the approach.
- The evaluation strategy used by the approach.
- The pros and cons of the approach.
- The availability of tool support (TS).

Please refer to sections 2.1and 2.2 for further details about these items.

Many of the approaches described in this appendix share the following cons:

- *REQ-ASSUMPTION*: the approach assumes the requirements already exist and fixed.
- Multi-COTS: the approach does not support multiple COTS selection for COTS intensive systems.
- *MISMATCHES*: It is not clear how to deal with COTS mismatches. The approach assumes that there is a set of COTS products that satisfy most of the requirements, at least to an acceptable level.
- *AHP/WSM-WEAKNESSES*: the approach uses AHP or WSM, and therefore inherits their weaknesses; e.g. consolidating the results into a single score is misleading, because strong aspects mask weak ones.

The acronyms described above are used when describing the approaches.

Table 2 The OTSO Approach

A1 : OTSO Approach (1995) [10, 16, 50, 51]	
Main Idea	OTSO (Off-The-Shelf Option) compares COTS products based on two factors: *value* and *cost*: (i) The *value* is estimated based on hierarchical evaluation criteria which consist of functionalities, qualities, strategic concerns, and architectural constraints. The evaluation criteria is influenced by 5 factors: application requirements and architecture, project constraints, availability of required features, and organization infrastructure (e.g. level of experience). AHP or WSM [7] are used to calculate the overall value of each alternative. (ii) The *cost* is estimated based on: acquisition cost, further development costs, and integration cost. The cost and benefit of each alternative are consolidated using AHP[7]
Main Steps	1. *Evaluation criteria*: define evaluation criteria. 2. *Search*: search the market for possible COTS. 3. *Screening*: filter out the COTS that do not comply with the must-have requirements. 4. *Evaluation*: evaluate the benefit and cost of COTS. 5. *Analysis*: use AHP/WSM to consolidate the evaluation results and select a COTS. 6. *Deployment*: integrate the selected COTS into system. 7. *Assessment*: assess the success of the selection process as a feedback for future uses.
Eval. Str.	▪ Progressive filtering
Pros	▪ Since OTSO is one of the first approaches reported in the literature, it served as a basis for other approaches. ▪ The feedback from the 'assessment' step helps to improve future selection processes in the organization.
Criticism	▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES*
TS	▪ Not available.

Table 3 The IusWare Approach

A2 : IusWare Approach (1997) [17]	
Main Idea	IusWare (IUStitia softWARis) is a two phase approach that is designed to evaluate COTS products in a formal and rigorous way. IusWare relies on multi-criteria decision aid (MCDA) [33] to select COTS products.
Main Steps	IusWare is a two phase approach: Phase 1. Design the evaluation model a. Identify the actors relevant to the evaluation, their role, objective of the evaluation, and available resources.

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

	b. Identify the evaluation type, either (i) ranking products from highest to lowest, or classifying products in 2 sets:the best and the remaining. (ii) formal description of the products. c. Define a hierarchy of evaluation attributes, often corresponding to quality attributes. d. Define the measures for the evaluation attributes e. Choose an aggregation technique, e.g. WSM or AHP[7] Phase 2. Apply the evaluation model
Eval. Str.	▪ Depend on Step (b) in Phase1.
Pros	▪ Focus on quality attributes. ▪ Formalize the evaluation process (although this might be seen sometimes as a disadvantage with light-weight projects -as it increases the complexity of the process).
Criticism	▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES*
TS	▪ Not available.

Table 4 The PRISM Approach

A3 : PRISM Approach (1997) [18]	
Main Idea	The PRISM (Portable, Reusable, Integrated, Software Modules) approach comprises two parts: (i) a generic component architecture, and (ii) a product evaluation process (PEP). The PEP process focuses on prototyping to ensure the selected product complies with industry standards represented by the generic architecture.
Main Steps	1. Identification: based on initial criteria, identify products that fit into the generic architecture. 2. Screening: identify one or more products with best fitness for further examination. 3. Stand-alone test: evaluate products against reliability, reusability, and system requirements. 4. Integration test: estimate how readily the product is to be integrated to the architecture. 5. Field test, re-evaluate the product after deployment in the target context
Eval. Str.	▪ Progressive filtering
Pros	▪ Address make-or-buy decisions. ▪ Address issues related to the integration into system. ▪ Emphasize the importance of prototyping: first outside the context, and second inside the context.
Criticism	▪ Identifying a generic architecture can be in many cases valid only for specific context. ▪ Many aspects of the approach are vague, e.g. how to acquire the requirements?, how to consolidate the results?, how to define the test cases?, etc. ▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES.*
TS	▪ Not available.

Table 5 The CISD Approach

A4 : CISD Approach (1997) [19]	
Main Idea	*The CISD (COTS-based Integrated Systems Development)* is a waterfall-style approach to identify and evaluate COTS products and then integrate them into the system.
Main Steps	a) Product identification a1. Identify requirements a2. Identify products and collect data about them a3. Prioritize products into sets for further evaluation. b) Product evaluation: Evaluate the product sets to find the optimal combination for integration. The optimal combination should include a set of collaborative COTS products. The products are evaluated for their 'individual functionality', 'interoperability of products', and 'performance of individual products and product sets'. c) Product integration: Build necessary product adapters into the selected product set.
Eval. Str.	▪ Progressive filtering + Puzzle assembly
Pros	▪ Support the selection of multiple COTS components in COTS intensive systems
Criticism	▪ CISD is a waterfall-style approach as each phase depends on the previous one ▪ Not clear how to evaluate a selected set of products and how they interact, e.g. in terms of resources. ▪ *REQ-ASSUMPTION, MISMATCHES*
TS	▪ Not available.

Table 6 The PORE Approach

A5 : PORE Approach (1998) [4, 52, 53]	
Main Idea	The PORE (Procurement-Oriented Requirements Engineering) approach focuses on the requirements engineering phase of the COTS procurement process. It suggests acquiring the requirements as the same time as acquiring and analyzing COTS products. PORE suggests iterating between requirements acquisition and product selection and rejection until finding a COTS that satisfies a sufficient number of the requirements (i.e. progressive filtering evaluation). PORE integrates techniques such as: feature analysis techniques [54] to help scoring the compliance of COTS to requirements, and multi-criteria decision making techniques [7] to help selecting a COTS.
Main Steps	PORE defines three templates for evaluating COTS products through three main phases: *Phase1.* First template gives guidance to acquire customer requirements and product information from COTS vendors given documents and information. *Phase2.* Second template gives guidance to acquire customer requirements and product information from vendors-led demonstrations; using test cases. *Phase3.* Third template gives guidance to acquire customer requirements and product information from customer-led product exploration; using trial versions
Eval. Str.	▪ Progressive filtering
Pros	▪ The parallel requirement acquisition and COTS selection means the defined requirements inform the selection process and vice versa, which is more realistic than assuming a fixed set of requirements. ▪ PORE suggests using the Analytic Hierarchy Process (AHP) method [7] only after Phase3 after defining a small number of products. this leads to minimizing the extra effort that is required by the AHP method.
Criticism	▪ PORE (as stated by its authors [52]) is labor intensive. ▪ PORE partially addresses the COTS mismatch problem. The progressive definition of the system requirements allow to refine those requirements in a way that reduces the mismatches with the products specifications. However, it is not clear it is till not clear how to deal with the situation when many mismatches are still unresolved. PORE does not define a systematic process that can be applied to (i) identify the influence of the remaining mismatches on the COTS selection decision, and (ii) to handle the remaining mismatches after selection. ▪ PORE does not provide clear guidelines on when to stop acquiring requirements and COTS information. ▪ *Multi-COTS, AHP/WSM-WEAKNESSES.*
TS	▪ PORE Process Advisor (prototype tool)

Table 7 The CEP Approach

A6 : CEP Approach (1999) [48]	
Main Idea	The CEP (Comparative Evaluation Process) approach is an advancement of OTSO approach. CEP suggests the use of the credibility feature. The more confidence in the source of data is, the higher 'confidence factor (CF)' that source gets. The CF is multiplied by the criteria scores when making product selection decisions. For example, consider a criterion is evaluated and gets a score of 100% satisfaction by a COTS product. If this criterion is evaluated based on 'verifying' the criterion, then CF=1.0 and the final score is still 100%. However, if the evaluation is based on 'vendor documents', then CF = 0.3 and the final score is 30% only. Eventually, CEP uses weighted averages to consolidate the evaluation results.
Main Steps	1. *Scoping evaluation effort*: set the expected effort and schedule for the evaluation activities. 2. *Searching / Screening*: search for COTS candidates

Proceedings of the 14th Annual IEEE International Conference and
Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

	based on 'must-have' criteria, and filter out the ones that do not satisfy them. 3. *Criteria definition*: give a detailed definition of the evaluation criteria. CEP defines several categories of evaluation criteria: functional, basic, architecture, management, and strategic. 4. *Evaluation*: evaluate COTS alternatives and estimate the CF factor for each criterion. 5. *Analysis*: analyze evaluation results and compare COTS alternatives.
Eval. Str.	▪ Progressive filtering
Pros	▪ Relies on a detailed evaluation criteria (i.e. related to management, etc – see main steps) ▪ Introduce the CF concept (although it might still need some improvements to be applicable)
Criticism	▪ Using the same CF value for all vendors is a strong assumption. Some vendors are more reliable and credible than others. ▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES*
TS	▪ Not available.

Table 8 The STACE Approach

A7 : STACE Approach (1999) [14, 20, 55, 56]	
Main Idea	STACE (Social-Technical Approach to COTS Evaluation) emphasizes the importance of non-technical issues when defining the evaluation criteria and conducting the evaluation process. The non-technical issues include social, human, and organizational characteristics, e.g. political and economic factors. STACE also emphasizes the customer participation during evaluation. In [55], studies were conducted to compare the application of STACE in developing and developed countries.
Main Steps	1. *Requirements elicitation*: the high level requirements are elicited from stakeholders, market studies, system documents, and domain knowledge 2. *Social-technical criteria definition*: the high level requirements are decomposed into measurable attributes. The decomposition addresses the non-technical characteristics (e.g. social-economic). 3. *Alternatives identification*: the high level requirements are decomposed into measurable attributes. The decomposition addresses the non-technical characteristics (e.g. social-economic). 4. *Evaluation*: the COTS alternatives are evaluated and ranked according to the social-technical criteria.
Eval. Str.	▪ Keystone evaluation, and Progressive filtering
Pros	▪ Incorporate the non-technical characteristics in the evaluation process. ▪ Support the negotiation between requirements elicitation and COTS evaluation through emphasizing customer involvement.
Criticism	▪ STACE is a very extensive process that requires more effort to apply. ▪ *Multi-COTS, MISMATCHES, AHP/WSM-Weaknesse*s.
TS	▪ Not available.

Table 9 The CRE Approach

A8 : CRE Approach (1999) [57]	
Main Idea	CRE (COTS-based Requirements Engineering) emphasizes the importance of non-functional requirement (NFR) as a decisive criteria when comparing COTS alternatives. CRE uses NFR framework [21] to model the NFRs for this purpose. Evaluating all NFRs requires more effort than most organizations have. So, CRE suggests select the most promising COTS candidates for detailed evaluation.
Main Steps	1. *Identification*: the core requirements and COTS candidates are identified. 2. *Description*: further requirements are defined. The NFR framework is used to model the NFRs. 3. *Evaluation*: COTS candidates are evaluated and WSM or AHP [7]are used for decision making. 4. *Acceptance*: the highest ranked COTS is selected if it passes some legal acceptance tests; if not, the next COTS is taken, and so on.
Eval. Str.	▪ Progressive filtering.
Pros	▪ Using the NFR framework during COTS evaluation facilitates addressing the NFRs.
Criticism	▪ Evaluating all NFRs (even for small set of COTS candidates) adds extra effort that is not available in most real situations. ▪ It is not clear how quality issues are verified, e.g. how a required level of quality is reached. ▪ It is not clear how to deal with unsatisfied quality attributes. ▪ *Multi-COTS, MISMATCHES, AHP/WSM-Weaknesses*.
TS	▪ Not available.

Table 10 The CAP Approach

A9 : CAP Approach (2000) a; Ochs, 2001 #184}}	
Main Idea	CAP (COTS Acquisition Process) is a measurement oriented approach where the evaluation process is tailored based on an estimation of the measurement effort. The tailoring is applied to: (i) a general evaluation criteria (called evaluation taxonomy) that is defined as a part of CAP, and (ii) the evaluation depth ranging from using only documents (as source of data) to using prototyping.
Main Steps	CAP compromises three main components: a) "Initialization", which deals with planning the evaluation process and its cost estimation: a1. Tailor and weight taxonomy of evaluation criteria a2. Estimate the cost of applying CAP a3. Elaborate measurement plane b) "Execution", which provides guidance to conduct the evaluation process itself: b1. *Exploration* for (i.e. identification of) COTS products b2. *Collecting measures*, where the COTS products are initially evaluated b3. *Screening*, where products with less compliance with the criteria are filtered out b4. *Collect measures*, where the COTS products are evaluated more extensively b5. *Ranking of COTS using AHP* [7] b6. *Make-or-Buy*, where the highest ranked COTS is selected if it passes a make-or-buy decision. c) "Reuse" component, where all info. gathered by other parts are stored for future use.
Eval. Str.	▪ Progressive filtering
Pros	▪ The introduction of the concept "tailorable evaluation process" which is useful to customize the process based on available resources. ▪ The introduction of a reusable taxonomy for the evaluation criteria.
Criticism	▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES*.
TS	▪ Not available.

Table 11 The CARE Approach

A10 : CARE Approach (2001 - ongoing) [23-26]	
Main Idea	CARE (COTS-Aware Requirements Engineering) approach focuses on the requirements engineering phase of the selection process. CARE defines two types of requirements: *native* (requirements acquired from customers) and *foreign* (requirements that may be implemented by existing COTS products). CARE tries to bridge the gap between the native and the foreign requirements by either asking the customers to change a goal or requirements or asking the vendors to customize the COTS product.
Main Steps	1. *Define goals*: define system goals (i.e. native goals) 2. *Match goals*: search for COTS products that match the goals (functional first, non-functional second).

Proceedings of the 14th Annual IEEE International Conference and
Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

	3. *Rank components*: perform a gap analysis to rank COTS products. 4. *Negotiate changes*: if mismatches found, negotiate changes to system goals or COTS components. 5. *Select Components*: select a set of COTS components.
Eval. Str.	▪ Progressive filtering
Pros	▪ Suggest to use flexible requirements which is more realistic than defining a fixed set of requirements ▪ Partially addresses the COTS mismatch problem. ▪ Support the definition of a plausible set of requirements based on different agents views.
Criticism	▪ Although CARE emphasizes the importance to map system requirements to product specs, the following weaknesses was identified: CARE suggests negotiating the resolution of the mismatches and how this influences the system and the product. However, this was given as a high level guideline which means, there is no systematic process that can be applied to (i) identify the influence of these mismatches on the COTS selection decision, and (ii) to handle the identified mismatches. What CARE suggests is similar to what currently happens (in an ad-hoc manner) in real-world selection. ▪ Not clear how to define requirements for multiple COTS selection in COTS intensive systems.
TS	▪ CARE Assistant Tool (prototype tool)

Table 12 The PECA Approach

A11 : PECA Approach (2002) [27, 58]	
Main Idea	PECA (Plan, Establish, Collect, and Analyze) is a high level process that has to be tailored for each specific project. PECA elaborates all activities in the evaluation process, and provides several alternatives for each activity to be used during process tailoring.
Main Steps	a) Plan the evaluation. a1. Form evaluation team a2. Create charter a3. Identify stakeholders a4. Pick the approach a5. Estimating resource & schedule b) Establish the criteria b1. Identify evaluation criteria b2. Construct criteria c) Collect the data - Estimate how each product complies with the evaluation criteria d) Analyze the data - analyze the data to select a COTS product using some techniques such as gap analysis.
Eval. Str.	▪ Depend on the project context.
Pros	▪ The tailorability of PECA makes it applicable in many contexts.
Criticism	▪ No clear guidance on how to tailor the process. the process only defines different alternatives, and the tailoring process based on human experience. ▪ *Multi-COTS, MISMATCHES.*
TS	▪ Not available.

Table 13 The BAREMO Approach

A12 : BAREMO Approach (2002) [59]	
Main Idea	The BAREMO (Balanced Reuse Model) focuses on how to use the AHP technique [7] to select COTS products.
Main Steps	1. Specify project objectives 2. Construct a decision tree (where root node is the problem objective, intermediate nodes are criteria, leaf nodes are alternatives) 3. Generate a comparison matrices for criteria at the same level in the decision tree 4. Value each alternative at the leaf nodes with respect to connected criteria at intermediate nodes. 5. Calculate a final value for each alternative using weighted addition scales.
Eval. Str.	▪ Not applicable.
Pros	▪ .Inherit the advantages of AHP technique.
Criticism	▪ Does not define a COTS evaluation / selection mechanism. ▪ Although BAREMO investigates the use of AHP in details, the concept itself is not new and has been already used in many other approaches.. ▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES* .
TS	▪ . Any available AHP tool, e.g. ExpertChoice [60]

Table 14 The Storyboard Approach

A13 : Storyboard Approach (2002) [28]	
Main Idea	Storyboard approach incorporates 'use cases' and 'screen captures' during the requirements engineering phase to help customers understand their requirements, and thus, acquire a more appropriate COTS that can be integrated with minimal customization.
Main Steps	1. Determine the requirements that can be satisfied by COTS products. 2. Develop use-cases for the identified requirements. 3. Identify and evaluate COTS products based on Steps 1 and 2 4. Based on the use-cases in Step2, develop storyboard using screen captures. Screen captures can be taken from actual COTS products, or from a custom interface developed using e.g. JAVA.
Eval. Str.	▪ Progressive filtering and Key-stoning.
Pros	▪ Provide a means to communicate requirements among stakeholders more clearly and concisely. ▪ Provide a means to manage users' expectations. ▪ Support the selection of multiple COTS components in COTS intensive systems.
Criticism	▪ Non functional requirements are not addressed. ▪ No formal evaluation process is presented. ▪ Not clear how to handle possible conflicts between the COTS products integrated together; i.e. resource usage. ▪ *MISMATCHES.*
TS	▪ Not available.

Table 15 The Combined-Selection Approach

A14 : Combined Selection Approach (2002) [29]	
Main Idea	The Combined-Selection (CS) approach is used to select multiple COTS products that all together satisfy the requirements. This approach performs its activities at two levels: local and global. *The global level* (i) addresses the overall process of the combined selection, (ii) fires individual selection processes for each area, and (iii) tries to find the best overall combination of products. *The local level* use existing COTS evaluation and selection techniques (e.g. OTSO [16] or PORE [4]) to select individual COTS that are combined at the global level.
Main Steps	1. Plan the global selection process and firing individual selection processes. (global level) 2. Identify COTS candidates for individual areas. (local level) 3. Identify global COTS integration scenarios; e.g. product A will cover area 1, while product B will cover area 2 and part of area 1. (global level) 4. Evaluate individual scenarios at each individual area. (local level) 5. Evaluate integration scenarios (global level). 6. Select the COTS products.
Eval. Str.	▪ Depends on the process used at the local level; e.g. progressive filtering, if PORE is used.
Pros	▪ Support the selection of multiple COTS components in COTS intensive systems.
Criticism	▪ CS inherits the weaknesses of the method used at the local level. ▪ No formal models are presented to guide the integration process and identifying the conflicts between the selected COTS components.

Proceedings of the 14th Annual IEEE International Conference and
Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

	▪ *REQ-ASSUMPTION, MISMATCHES.*
TS	▪ Not available.

Table 16 The WinWin Approach

A15 : WinWin Approach (2003) [3, 61]	
Main Idea	The WinWin spiral model is a risk-driven approach adapted from the classical software development spiral model [62]. WinWin follows an iterative process in which the requirements are acquired in parallel to evaluating the COTS products. WinWin emphasizes concurrent product identification and process refinement. A decision framework is used to provide guidance for the COTS-based development decisions, e.g. make-or-buy, COTS selection, COTS tailoring, glue-coding, etc.
Main Steps	WinWin uses five iterative steps, in which more stakeholders and OC&P (Objectives, Constraints and Priorities) are identified in successive iterations, and more refinement of the process is applied: 1. Identify stakeholders, and system OC&P 2. Evaluate products with respect to OC&P, and address risks (e.g. related to customer support) 3. Elaborate product and process definition 4. Verify and validate product and process definition. 5. Stakeholders' review and commitments
Eval. Str.	▪ Progressive filtering
Pros	▪ Addressing risks early in the selection process reduces the cost of handling them in the future. ▪ The decision framework provides an inclusive description of the actual development process. ▪ Gradually defining the stakeholders' needs as well as the process fits more in many real-world situations. ▪ WinWin addresses the selection of multiple COTS components in intensive COTS-based system.
Criticism	▪ Although multiple COTS selection is support, no detailed guidelines (or formal process) were defined to elaborate this issue. ▪ Although WinWin emphasizes the importance to address risks (including those related to COTS mismatches) during its spiral process, this was given as a set of high level guidelines. That is, there is no systematic process that can be applied to (i) identify the influence of these mismatches on the COTS selection decision, and (ii) to handle the identified mismatches after selection.
TS	▪ Not available.

Table 17 The Approach by Erol et al.

A16 : The Approach by Erol et al. (2003) [30]	
Main Idea	The approach by Erol et al. is an evaluation approach that supports selecting a COTS product from a finite set of products based on: (i) more than one objective and (ii) a set of quantitative (e.g. cost) and qualitative (e.g. linguistic variables) data. The approach uses fuzzy QFD (Quality Function Deployment) [63] to collect and quantify the qualitative data. Then, goal programming (a generalization of linear programming) is used to suggest near optimal solutions to the decision maker.
Main Steps	1. Acquire requirements and product information. 2. Transform qualitative data to quantitative data using fuzzy QFD. 3. Construct a multi-objective model for objectives: (i) maximize customer value, and (ii) minimize cost. 4. Solve the model several times with slightly different parameter values to get multiple feasible solutions from which the decision maker can choose one solution.
Eval. Str.	▪ Not applicable
Pros	▪ Address qualitative data. ▪ Provide more than one feasible solutions to support making the final decision.
Criticism	▪ Do not address activities other than COTS evaluation. ▪ *REQ-ASSUMPTION, Multi-COTS, MISMATCHES.*
TS	▪ Not available.

Table 18 The DesCOTS system

A17 : DesCOTS system [31]	
Main Idea	DesCOTS ((**D**escription, **e**valuation and **s**election of **COTS** components) system includes a set of tools that can be used to evaluated COTS products based on quality models such as ISO/IEC9126 [13]. DesCOTS is built on the work proposed in [11]
Main Steps	DesCOTS follows the GCS process and uses the following steps to define the evaluation criteria: 1. From ISO/ICE9126, determine the quality characteristics and subcharacteristics that will be used. 2. refine the subcharacteristics as need; e.g. 'suitability' to basic suitability, and added suitability 3. Refine subcharacteristics into measurable attributes. 4. Refine derived attributes to basic ones; e.g. GUI quality to user friendliness, depth of menus, etc. 5. Identify relationships between quality entites 6. Determine metrics for attributes.
Eval. Str.	▪ Progressive filtering.
Pros	▪ Detailed method to define the evaluation criteria. ▪ Considers quality characteristics. ▪ Can be adapted to many domains.
Criticism	▪ *Multi-COTS, MISMATCHES.*
TS	▪ Yes

Table 19 The MiHOS Approach

A18 : MiHOS Approach [32]	
Main Idea	MiHOS (Mismatch-Handling aware COTS Selection) focuses on handling the mismatches between COTS candidates and the requirements. MiHOS firstly provides guidelines to handle the mismatches that are detected during the COTS selection process, and secondly uses decision support techniques to (i) study the cost of resolving these mismatches on the COTS selection decision, and (ii) support handling the mismatches of the selected COTS product.
Main Steps	MiHOS has two components: the "COTS selection" component which follow exactly the GCS process, and the "mismatch handling" component which is divided into two parts: Part1 (of mismatch handling component): 1.1 Analyze the detected mismatch 1.2 Decide whether the mismatch should be tolerated or resolved by (i) adjusting the requirements, or by (ii) adjusting the COTS product. If (ii) is chosen, then the mismatch is postponed until a COTS is selected. Part2 (of the mismatch handling component): 2.1 Modeling: define problem parameters, e.g. available resources for resolving the mismatches of the selected COTS product, the possible techniques to resolve each mismatch, cost of each technique, etc. 2.2 Exploration: use optimization techniques to fine optimal or near optimal solutions. 2.3 Consolidation: experts should review the results and either accept one solution or adjust the problem parameters in (2.1) and run the model again.
Eval. Str.	▪ Keystone identification / Progressive filtering
Pros	▪ Can handle mismatches during/after selection process. ▪ Uses goal-driven criteria definition which allows for more informative decisions. ▪ Use interactive decision support techniques which enable decision makers to have more participation in and control over their decisions.
Criticism	▪ Requires effort more than most selection approaches. ▪ Is not intended to address quality requirements. ▪ No support for the selection of multiple COTS components in COTS intensive systems ▪ *Multi-COTS.*
TS	▪ MiHOS-TS, a prototype.

Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)
0-7695-2772-8/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY