A Wish List for Requirements Engineering for COTS-based Information Systems

Vito Perrone

HOC - Hypermedia Open Center
Department of Electronics and Information, Politecnico di Milano
Via Ponzio 34/5 20133 Milano (Italy)
perrone@elet.polimi.it

Abstract. This paper summarizes the main achievements of a research whose main goal was to investigate the current state-of-art in the field of requirements engineering for COTS-based systems. For this purpose, we have reviewed the most relevant contributions in this field over the last 10 years have been considered. After analyzing these research contributions, we defined a scenario composed by a number of punctual but relevant contributions and a number of methodological approaches coping with the requirements definitions for such systems. Finally, on the basis of this scenario, a *Wish List* of desirable features of a hypothetical approach has been defined and compared against the current situation. This list may act as an empirical means for evaluating new approaches addressing RE for COTS-based systems, and bases its foundations on the current needs pointed out by the major experts in this field.

1 Introduction

Many factors contribute to the success of an Information System (IS). Among them, a primary measure is certainly the degree to which it meets the purpose for which it has been thought (say *acceptance degree*). *Software systems Requirements Engineering* (RE), broadly speaking, is the process of discovering that purpose, by identifying stakeholders and their needs, and by documenting them in a form that is appropriate for analysis, communication, and subsequent implementation. Over the past decade, a huge amount of interest in methods and techniques dealing with RE has grown in the research community, leading to the definition of several well-affirmed approaches.

On the other hand, the goals of developing systems in a better, faster, and cheaper way, continue to drive software engineering practitioners and researchers to investigate software engineering methodologies [1]. To face these market drivers, the current practice of IS development consists of adopting commercial packages usually called COTS (commercial-off-the shelf components).

In the light of the above considerations, this paper addresses the issue of how to combine well-known benefits coming from applying RE approaches with the need of using COTS packages in building up a new IS? Indeed, using COTS packages requires systematic approaches that are able to consider the COTS option from the early stages of the building process and in particular to come up with requirements that make feasible such an option. Unfortunately, RE approaches used for developing systems from scratch

may be, and usually are, inadequate to front the development using COTS. Various factors affect the development of COTS based systems: COTS are those the market offer; they are sold "like-that" sometime like a black-box; their code is often not available; they have been though as general purpose solution in a specific domain; and so forth.

Starting from these considerations, this research aims at investigating the effects of this trend on requirements engineering practice. In particular, we try to collect the most important results achieved in the RE research community, putting them in the form of a *wish list* of features a "complete" approach should embody.

2 Using COTS in building up an Information System

After analyzing the most relevant contributions in RE field from one hand, and for developing COTS base systems on the other, we have been able to make a picture of the current situation in this sector. The situation can be described as composed by a number of punctual contributions and a small number of complex approaches providing a means to cope with the main issues in building a new COTS-based IS.

A first, partially surprising result of the investigation is that most of the approaches take into consideration the problem of treating COTS only operate at design level. This means that a considerable part of the research efforts dealing with COTS neglects the influence of using COTS during the requirements engineering activities, supposing that any is performed.

A second relevant result is that most of the activities performed throughout the requirements engineering process for such systems, if compared with the ones performed for a system to be built from scratch, are strongly influenced by a sort of *shift of the paradigm* according to the system is built. In the next paragraph a better explanation of this point is provided.

2.1 Paradigm Shifting

Using COTS components to build up a system implies a fundamental shift of the paradigm used to build a system from scratch and starting from its requirements. In a *custom system*[1], the process of building a system follows a paradigm in which, ideally speaking, requirements define the system to be. For such systems, the development process consists, in a broad vision, of the steps shown in fig. 1

[1] A system where a customer produces a set of requirements for hardware/software of the system and a contractor develops and delivers that system.

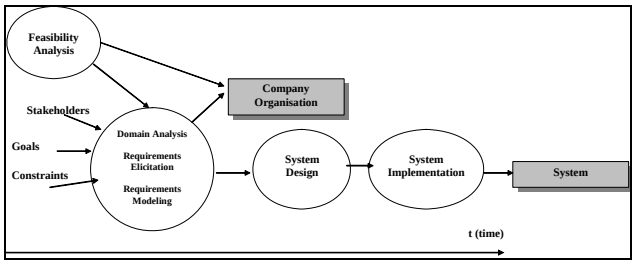

Figure 1: Typical life cycle of a software system development

The *Feasibility Analysis* phase can be seen as an early domain and requirements analysis, in which the organization has to decide whether the system makes sense and which are the main directions to be followed in order to meet its own business objectives. For a custom system, if we consider the process as a whole looking at the sequentially-ness of the above mentioned phases, regardless the intrinsic iterative nature of the overall process, we can say that the Feasibility Analysis defines the large-grain structure of the organization the system will support. Then the RE process (as a whole) defines the thin-grain structure of the organization. In this light, system design can be considered as a function (ideally) of the RE output. This is because the system design and implementation are founded on this output. This is evidently an ideal situation, based on the assumption that designers completely understand the requirements specification and the implementation team is able to fulfill the design specification. Moreover in this picture we neglect the iterative nature of the process because the main aspect we want to point out is that the organization (by means of contractors) develops the system from scratch keeping the control of all or most of the phases: collect and define requirements; identify the architecture that satisfies the requirements; design individual subsystems in detail in order to fit within the architecture; code, test, and debug modules to meet the specified requirements; integrate sets of modules and subsystems into the complete system.

In COTS based systems the fundaments of the above paradigm need to be revised. In particular, some sequences and dependencies among phases of the overall process can give out. In this research we focus on the RE part showing as using COTS impacts this crucial phase of the overall process by collecting a number of insights coming from the research community.

2.2 Defining COTS products and COTS-based systems

The definitions of COTS and COTS related concepts which can be found out in literature are usually very broad and cover a large variety of products. As a result, researchers and practitioners use the same word with different meanings. Some of these definitions are discussed in [2]. In order to avoid misunderstandings, here we provide a definition both for COTS products and COTS-based systems so that the reader can refer to these to understand the following sections. The definitions are quite general and include most of the others, making the paper consistent with most of the existing COTS definitions.

COTS Product Definition
A COTS product can be considered as one that:

- Can be sold, leased, or licensed to the general public (different license forms)
- It is offered "like-that" by a vendor trying to profit from it (or by a community trying to benefit from its usage by an increasing number of users)
- Intellectual property rights are retained by the vendor
- Identical copies are sold to a wide number of customers
- Access to source code as well as internal documentation is usually unavailable
- Complete and correct behavioral specifications are not available
- Periodical releases, with unpredictable evolutions and modifications, may be proposed by vendors
- Sometime, customers are provided with a number of predefined extension capabilities and personalization hock-points, but eventual modifications are not more under the vendor's responsibility

COTS-based system definition
The definition of COTS-based system we use, is the one proposed by Carney in [3] where he takes the point of view of the delivered system, instead of the parts: he identifies three types of COTS systems as a function of the number of COTS used and their influence on the final system: *turnkey systems* are built around a (suite of) commercial product(s); *intermediate systems* are built around one COTS but integrate other components; *integrated systems* are built by integrating several COTS, all on the same level of importance. In our investigation we refer to turnkey systems.

3 A Wish List

In this section, we attempt to summarize, in the form of a *wish list*, the desired features that an **approach** addressing requirements engineering for COTS-based systems should include or support. The proposed *wishes* are organized in four categories:

- Organizational-Management issues: aspects concerning the impact of using COTS on the organizational structure of a company. Techniques operating here may point out organizational changes that should be performed before other activities.
- RE Process issues: the impact of the COTS solution on the overall RE process. Typically this may involve an alteration both of the normal activities performed in the RE process for custom systems and addition of specific activity as well as process life cycle alteration, and so forth.
- Requirements contents issues: using a COTS solution can influence requirements acquisition, modeling and analysis. Here we refer to desired/required modifications of the way requirements are acquired, of which contents should be emphasized and which underplayed, how requirements should be specified and integrated with other specific information.
- COTS packages (or components) selection: selection of COTS package must be performed in coordination with the RE activities [4] and even more should be considered as part of the whole RE process. But which features the selection criteria

and techniques should embody? How to integrate selection with RE and which factors should be considered?

3.1 Organizational-Management issues

W1.1: Evaluating the opportunity of adopting a COTS solution (Management Issue). Before opting for a COTS-based solution, a complete approach should consider whether it is worth to follow this strategy or it is not feasible. COTS solutions are usually adopted to contain costs and to hit faster the market. Unfortunately the complexity of the system and the kind of market can make this choice worst than developing a system from scratch. There are several key management decisions to be considered before to opt for COTS software. In [5] a number of issues to be addressed are reported. Summarizing, most of them bring up the need for a systematic approach addressing these issues in the early stage of the system definition.

W1.2: Defining the business architecture and the software supply chain before or together to the software development process. This draws upon the considerations reported in [6]. Starting by these considerations, it can be argued as a particular attention should be put in defining an appropriate supply chain taking into account the business strategy. The approach should explicitly consider this factor since the very early stage of the development.

W1.3: Organization structure more flexible and open. In an approach for building custom systems, the feasibility analysis dictates the organizational structure of a company. The RE process and the following activities of the development life cycle, rely on this structure with some minor adjustments. In the case of COTS-based systems this structure can be strongly influenced by issues coming from the COTS market and products availability. For this reason, a suitable approach should consider a possible revision of the above structure, forcing its definition to be considered as part of the overall process. [6]

3.2 Requirements Engineering Process issues

W2.1: Packages selection as an integrated and parallel activity since early requirements acquisition. It is widely accepted that COTS procurement activities must be interleaved with other traditional RE activities. Traditional approaches fail to support effectively these activities which should be both iterative and concurrent [7]. There are critical relationships among technology and product selection, requirement specification, and architecture definition. If the architecture is defined just to fulfill the requirements and then COTS products are selected, there will be only a few products (when any is available) that in some way fit within the chosen architecture. Pragmatically, three essential elements (requirements, architecture, and product selection) must be worked in parallel with constant trade-offs among them. Traditional approaches tend to confine the result of the RE process (as a whole) to the architecture definition while COTS product and suppliers are evaluated to better fit the defined architecture and to stay within the budget limits. Moreover, the evaluation is often performed through a trivial questionnaire sent to the potential supplies.

W2.2: Iterative life cycle process. Nowadays ISs are strongly interactive. For such systems it's proven the requirements engineering process must be iterative. Furthermore, for COTS-based systems, this need is felt twice since the intrinsic nature of the selection procedure requires an iterative process allowing a progressive reduction of candidate packages [7] [8].
W2.3: Process guidance and techniques integration. A high desired aspect of the approach is providing process guidance, even more due the actual lacking, in this direction, of current approaches [7] [8] [1]. Well-known techniques, already operating in the various facets of the RE field, may be included instead of re-thinking to new ones. The approach should enable the procurement team selects the most suitable techniques among various options, and organizes these in an iterative process. Some of these techniques should be used in the requirements activities, whilst others should perform a selection among the candidate packages.
W2.4: Integration of multi-criteria decision making techniques. The selection process must be systematic and well defined. Activities performed in this phase have to be interleaved within the overall iterative process. On the other hand, the techniques adopted in this phase have to be explicitly defined in order to record the rationale used in the decision of chosen COTS products. There is a range of techniques that can be used. Card sorts, for example, are simple to use and can acquire requirements that discriminate between products [10]. Two other approaches, among the nowadays most used, are AHP (Analytic Hierarchy Process) and WSM (Weighted Scoring Method), but they seem to work properly as quantitative criteria but to be inadequate to support qualitative reasoning. A further challenge is to keep the rationale of decisions made over the overall life cycle not only during the evaluation process [11].
W2.5: Integration of techniques to specify products evaluation to be used in matching requirements changes and evolution. One of the main problems is to extract from the commercial product description the information needed to make a selection against the acquired requirements. Once this effort has been performed we should prevent its wasting. These techniques should allow the procurement team records the rationale for product-requirement compliance to better react to requirements modification, addition and evolution. An example of these techniques can be found in [12]. Furthermore, the recorded rationale should refer to decisions made over the development life cycle not only during the evaluation process.
W2.6 Representing and storing the acquired selection knowledge. A considerable portion of the overall effort required to select among the available COTS solutions is due to the lack of an adequate knowledge of the specific market. For this reason, the approach should provide a technique first to represent and then to store this knowledge in order to be reused in future projects [1].
W2.7 Support Tool. A suitable tool should support the overall process. This tool should provide the procurement team with a support for all the above listed issues, including a feature for customising the process on the basis of the specific domain, team composition, time constraints, available techniques, and so forth.

3.3 Requirements contents issues

W3.1: Requirements Adaptation and balancing against COTS features. In a custom system, requirements are envisioned by the stakeholders' goals and on the basis of a set

of specified constraints. In a COTS system, requirements can not neglect the availability of COTS products on the market, in that an available market may not exist to fulfill some requirements. This leads to requirements adaptation taking into account the knowledge of the existing market acquired little by little. Requirements elicitation and specification should support these adaptations [13].

W3.2: A more flexible requirements acquisition and specification. In COTS-based development, requirements statements need to be more flexible and less specific [14] [6]; otherwise it may be very hard to find out an appropriate package: products selection would be too strict or the amount of product modification would be so large that the COTS solution becomes not that worth anymore [15].

W3.3: Requirements specification should be structured in that tests cases can be easily performed since by the very early stages of the iterative process. Differently from a bespoke system in which test cases are used mostly to validate the developed application against the design in the latter stage of the product life cycle, for this kind of systems test cases can be used to select among the candidate components. Since the selection start by the very early interactions of the process, requirements are required to be well structured for test cases since there early phases [1]. A possible way to meet this desired facet of the RE process is to acquire requirements using use cases and scenarios techniques that make the requirements more amenable to test cases generation. For example in [16] an approach for decomposing goals into tasks which achieve these goals is proposed through generation of use cases that are equivalence classes of task scripts, and scenarios that are equivalence classes of use cases.

W3.4: Support of market evolution against requirements specification. One of the main issues in dealing with requirements for COTS-based systems is the impact of the evolution of used packages over the market [5]. This evolution impacts the development and maintenance of the system, therefore requirements should consider, for example, the supplier updating police as an additional selection criteria. If neglected, changes in COTS releases, competitive threats, stakeholders, reorganizations, and price structures may make requirements increasingly obsolete.

W3.5: Support of communication between the bespoke (custom) and COTS parts. A modern IS is made up both bespoke parts and COTS parts. Due their different nature, requirements giving rise to COTS parts and requirements giving rise to custom parts should be distinguished. On the other hand they are definitively related each other. These relations should be specified explicitly [5].

W3.6: Explicit consideration of the unused parts of each COTS-package. COTS package are more general purpose than requirements they answer [5]. The unused part can not be simply ignored because it may impact some functionalities and aspect of the system to-be. Requirements should handle the question of how to treat this unused part. This information will be of particular importance during the testing phase [17].

W3.7: Distinction between requirements used to select COTS packages and requirements not helping the selection. There are some requirements which in general are provided by all or most of the available package and other requirements which are very specific for the needs of a specific IS. In [9] this distinction has been pointed out calling them respectively “core” and “peripheral” requirements. Since one of the most important concerns of the RE for such systems is to define the procurement criteria, it’s evident as for this task the former should be ignored, while the latter should be emphasized and acquired in more detail [18]. Anyway the approach should provide a way to discern between these requirements categories.

W3.8: Detailing Non-Functional Requirements for components selection. Since end-users are not in a position to specify functional requirements or to control the process of component development, there is no need for detailed functional requirements. As moving the focus from functional to non-functional requirements a number of topics, which should be addressed by the approach, come into light, as reported in [19].

3.4 COTS packages selection

W4.1: Direct consideration of adaptation costs for packages selection. Although glue-code development usually accounts for less than half of the total effort for the development of the COTS-based System software, the effort per line of glue code averages about three times the effort per line of custom applications code [5]. This consideration lead to adding the adaptation costs in techniques used to select packages. The distance between a package as sold and as ready to be integrated into the system should be a driver of the procurement process.

W4.2: Performing the decision of either buying or developing. For some parts of the system, adopting a COTS package isn't always the best choice. Such a decision should be evaluated during the COTS selection activity [5].

W4.3: Consideration of contract aspects for packages selection. Non-development costs, such as licensing fees, are significant and the procurement process must optimize them. SEI identifies three significant CBS activity areas: vendor relationships, license administration, training and cultural transition [20]. All these costs can significantly impact the worth-ness of a COTS solution instead of another one.

W4.4: Using some kind of weighted metrics to evaluate package compliance against functional and non-functional requirements. Lack of such metrics makes very hard and ineffective the product selection activity. Fit criteria should be expressed in terms of logical expressions or quantifiable tests to undergo commercial requirements standards [8]. A possible technique is repertory grid analysis [21], in which stakeholders are asked for attributes applicable to a set of entities and values for cells in an entity-attribute matrix. These metrics may weight a number of factors as costs, supplier credibility, contract forms, volatility of the packages on the market, required adaptation effort, adheres to current product standards, integration level, communication required against other packages, and so forth. Therefore, the approach should define a distance or ratio scale to be used for obtaining criteria scores in evaluating different COTS products.

W4.5: Stakeholders involvement in the product evaluation. Techniques used to select components among the possible ones should directly involve stakeholders [8] to further elicit requirements or to assess those already acquired reaching a deeper detail level.

W4.6: Minimization of independent COTS products. COTS-based system development and post deployment efforts can scale as high as the square of the number of independently developed COTS products targeted for integration, because integrating n COTS products involves potentially $n(n-1)/2$ interfaces . The conventional wisdom in the use of COTS components is the more of the system that can be built using COTS components, the better. Beyond a certain point, however, an increase in the number of COTS components in a system may actually reduce the system's overall economic life span rather than increase it [22]. Taking in consideration these observations, the selection criteria should aim, among other things, to minimize the number of packages and vendors that are going to build the system.

W4.7: Support Tool. A suitable tool should support the selection activity. This tool should provide the procurements team with a support to all the above listed issues, that is, it should include a metric system, consider package costs and contract aspects, allow strategy definitions, allow an iterative process, store the acquired knowledge.

4 An Empirical Evaluation of Current Approaches

In this section we show the results obtained by matching some of the existing approaches against the wishes previously listed. The selected approaches are: RUP (Rational Unified Process) [24], OTSO (Off-The-Shell Option method) [17], MBASE (Model Based Architecting and Software Engineering) [25], CAP (COTS Acquisition Process) [26], PORE (Procurement Oriented Requirements Engineering) [4,18], CARE (COTS Aware Requirements Engineering) [1]. Information allowing this comparative study have been acquired from the literature currently available about every approach.

For each combination wish/approach we give an evaluation of how much this wish is satisfied by that approach, rating by *Poor* (either the satisfaction level is not clear or there are just some ideas concerning the wish topic), *Sufficient* (the approach takes into consideration this wish but just partially), *Complete* (the approach seams satisfying that wish completely).

	RUP	OTSO	MBASE	CAP	PORE	CARE
W1.1	Poor	Poor	Poor	Poor	Sufficient	Poor
W1.2	Poor	Poor	Poor	Poor	Poor	Sufficient
W1.3	Poor	Poor	Poor	Poor	Poor	Poor
W2.1	Poor	Complete	Poor	Poor	Sufficient	Complete
W2.2	Poor	Poor	Poor	Poor	Poor	Poor
W2.3	Sufficient	Sufficient	Poor	Sufficient	Complete	Complete
W2.4	Poor	Complete	Sufficient	Sufficient	Complete	Complete
W2.5	Poor	Sufficient	Sufficient	Sufficient	Sufficient	Sufficient
W2.6	Poor	Poor	Sufficient	Sufficient	Complete	Complete
W2.7	Complete	Poor	Poor	Poor	Poor	Poor
W3.1	Poor	Sufficient	Poor	Poor	Poor	Complete
W3.2	Poor	Poor	Poor	Poor	Complete	Complete
W3.3	Poor	Poor	Poor	Poor	Complete	Complete
W3.4	Poor	Sufficient	Poor	Poor	Sufficient	Sufficient
W3.5	Complete	Poor	Poor	Sufficient	Poor	Sufficient
W3.6	Poor	Poor	Poor	Poor	Poor	Poor
W3.7	Poor	Sufficient	Poor	Poor	Complete	Complete
W3.8	Poor	Sufficient	poor	Poor	Poor	Sufficient
W4.1	Poor	Sufficient	poor	Poor	Poor	Sufficient
W4.2	Poor	Sufficient	poor	Poor	Poor	Sufficient
W4.3	Poor	Complete	poor	Complete	Sufficient	Sufficient
W4.4	Poor	Complete	poor	Complete	Poor	Sufficient
W4.5	Poor	Poor	poor	Poor	Complete	Complete
W4.6	Poor	Poor	Poor	Poor	Poor	Poor
W4.7	Complete	Poor	Poor	Poor	Poor	Poor

Table 1: Satisfaction Level of wishes in analysed methods

The previous table can be explored both by row and by column drawing out some considerations about, respectively, the satisfaction level of a specific wish throughout all approaches, and the response of specific approaches in terms of the recognized wishes. In the following, some examples of this kind of analysis are reported:

Exploring by column:

- The RUP approach does not generally accomplish most of the desired wishes. Mostly it is due to the intrinsic nature of this approach, because, even if RUP provides support to include COTS components in a system, it operates at the logical design level of the system. From the table, it can be claimed as RUP totally satisfies

the wish of having a support tool (that is Rational Rose™) and the generic characteristics of a modern process, but is weak when dealing with specific RE issues.

- CAP is particularly weak in dealing with the integration of the requirements acquisition to the COTS selection activity. This is obvious since it starts by an already acquired requirements base to select suitable COTS package. This aspect, combined with some other lacks, makes this approach incomplete.
- By comparing the last two columns each other and against the rest, it can be noticed as the last two approaches appear to be a bit ahead since they satisfy several wishes. Furthermore, it can be noticed as CARE strengthens PORE, since it is generally stronger in all wishes, as the same authors asserted "The CARE approach draws upon the good available ideas in current RE methodologies including RUP, MBASE, and PORE" [1].

Exploring by row:

- It appears evident as all wishes dealing with the first group, that is Organizational-Management issues are generally neglected.
- The approaches have a very variable behavior in respect of *selection technique* wishes, some ones are stronger for some wishes and some others are better with other wishes. By this consideration, inspecting all the approaches and extracting respective strengths, some enhancements could be performed.
- By looking the row belonging to W3.6 it can be argued as all approaches ignore the unused parts of used COTS packages, although in [23] it's claimed as this behavior may lead to unexpected consequences in the final system.

4 Concluding Remarks and Call for Research

The inspection of the research literature concerning COTS-based systems and in particular RE for such systems, brought us to draw the current situation in this field that we can describe as composed by a number of punctual contributions as well as complex approaches composed by a number of sub-activities in charge of defining the system to be. A first interesting consideration, raised out by this analysis, is that a considerable part of the existing approaches neglects the requirements problem, providing just some features to specify COTS components in designing the system. A further contribution of this research has been to determine the impact of using COTS packages on the requirements engineering activities and process. Finally, the main contribution of this research has been to recognize and describe, in the form of a wish list, a number of desired characteristics of a suitable approach in charge of defining a system including a more or less considerable part composed by COTS packages. Moreover, the existing approaches have been reconsidered against this list, allowing a recognition of what has been already accomplished and what is desirable for future researches.

In particular, this can be translated in some concise call-for-research:

Call1: The research has definitively shown as the UML community is lacking of an approach considering COTS since the requirements engineering phase. This is clear by examining the unique contribution treating COTS with UML, that is [24], where COTS are considered only during the design phase of the system.

Call3: The selection strategy, embodying all the aspects described in paragraph 3.4, should be definable and customizable so as to adapt the method to the specific application case.
Call4: All the existing approaches show an evident lack in supporting the requirements activities for such systems by means of tools. This is recognized as a main factor that tends to enlarge the already existing gab between the research community and the industrial sector. This is because tools usually allow a reduction of the exploitation time and attract people that otherwise should perform a number of activity manually.

References

1. L. Chung and K. Cooper: "A Knowledge-Based COTS-Aware Requirements Engineering Approach". In proceedings of 4th Int. Conf. on SEKE'02, ACM Press. July 15-19, 2002, Ischia, Italy.
2. Morisio M., Torchiano M.: "Definition and classification of COTS: a proposal". In Proceedings ICCBSS, Orlando (FL) February 4-6, 2002.
3. D. Carney, F. Long: "What Do You Mean by COTS?". IEEE Software, March/April 2000, pp. 83-86
4. C. Ncube and N. Maiden: "Guiding parallel requirements acquisition and COTS software selection". In proceedings of the IEEE International Symposium on Requirements Engineering 1999.
5. Jeffrey Voas: COTS Software: "The Economical Choice?". IEEE Software, 15(2):16-19, Mar 1998.
6. Farbey, B, & Finkelstein, A.: "Software Acquisition: a business strategy analysis". In proceedings of Requirements Engineering 2001 (RE01).
7. L.Brownsword, D.Carney, T.Oberndorf: "The Opportunities and Complexities of Applying Commercial-Off-the-Shelf Components". SEI Interactive, 6/98, 1998.
8. N. Maiden and C. Ncube: "Acquiring COTS Software Selection Requirements". IEEE Software, Volume 15 Issue 2, March-April 1998, pp. 46-56
9. Finkelstein, A., Spanoudakis, G., and M. Ryan: "Software Package Requirements & Procurement". In proceedings of the 8th Int. Workshop on Software Specification & Design,IEEE CS Press, 1996.
10. N.A.M. Maiden and G. Rugg: "ACRE: Selecting Methods for Requirements Acquisition". Software Eng. J., Vol. 11, No. 3, 1996, pp. 183-192
11. Carina Alves, Anthony Finkelstein: "Challenges in COTS decision-making: a goal-driven requirements engineering perspective". In proceedings of SEKE 2002: 789-794
12. T.P. Moran, J.M. Carroll: "Design Rationale: Concepts, Techniques, and Use". Lawrence Erlbaum Assoc., Hillsdale,N.J., 1996
13. C. Alves and A. Finkelstein: "Negotiating Requirements for COTS-Based Systems". In proceedings of 8th Int. Workshop on Requirements Engineering: Foundation for Software Quality, in conjunction with RE'02.
14. D. Carney: "COTS Evaluation in the Real World". SEI Interactive December 1998.
15. D. Carney: "Requirements and COTS-Based Systems: A Thorny Question Indeed". SEI Int. June '99.
16. I. Graham: Task Scripts: "Use Cases and Scenarios in O-O Analysis". O-O Systems 3, 1996, pp.123-142
17. Kontio, J.: OTSO: "A Systematic Process for Reusable Software Component Selection". TR, Dec. 1995.
18. N. Maiden, C. Ncube, and A. Moore: "Lessons learned during requirements acquisition for COTS systems". Communications of the ACM, Vol. 40, no. 12, 1997
19. L. Beus-Dukic: "Non-functional requirements for COTS software components". In proc. of ICSE '00
20. Basili V. R. and Boehm B.: "COTS-Based systems Top 10 List". IEEE Computer, vol. 24,5 May '01.
21. Shaw M.L. and Gaines B.R.: "Requirements Acquisition". Software Engineering Journal, 1996, 11 (3)
22. Tumuluri S., Raja S., and Cooper K.: "Commercial off-the-Shelf (COTS) Software Engineering Methodologies: A Comparative Study". T.R. UTDCS-24-01, December 2001
23. M. Feather: "Language Support for the Specification and Development of Composite Systems". ACM Trans. on Programming Languages and Systems 9(2), Apr. 87, 198-234.
24. G. Booch, J. Rumbaugh, and I. Jacobson: "The Unified Modeling Language User Guide". Addison Wesley Longman, Inc., USA, 1999.
25. Boehm, B: "Requirements that handle IKIWISI, COTS, and Rapid Change". IEEE Computer, Volume: 33 Issue: 7, July 2000.
26. Ochs, M.A. et al.: "A Method for Efficient Measurement-based COTS Assessment ad Selection – Method Description and Evaluation Results". In proc. IEEE 7th International Software Metrics Symposium.