WinWin Spiral Approach to Developing COTS-Based Applications
EDSER-5 Position Paper
Barry Boehm, Dan Port, Ye Yang
University of Southern California, Texas A&M University*
{boehm, dport, yey}@cse.usc.edu

Abstract
Data collected from five years of developing e-service applications at USC-CSE reveals that an increasing fraction have been commercial-off-the-shelf (COTS)-Based Application (CBA) projects: from 28% in 1997 to 60% in 2001. Data from both small and large CBA projects show that CBA effort is primarily distributed among the three activities of COTS assessment, COTS tailoring, and glue code development and integration, with wide variations in their distribution across projects. We have developed a set of data-motivated composable process elements, in terms of these three activities, for developing CBA’s as well an overall decision framework for applying the process elements. We present a real-world example showing how it operates within the WinWin Spiral process model generator to orchestrate, execute, and adapt the process elements to changing project circumstances.

1. Definitions and Context
1.1 Definitions
We adopt the SEI COTS-Based System Initiative’s definition [7] of a COTS product: A product that is:
- Sold, leased, or licensed to the general public;
- Offered by a vendor trying to profit from it;
- Supported and evolved by the vendor, who retains the intellectual property rights;
- Available in multiple identical copies;
- Used without source code modification.

We also follow the SEI in defining a COTS-Based System very generally as “any system, which includes one or more COTS products.” This includes most current systems, including many which treat a COTS operating system and other utilities as a relatively stable platform on which to build applications. Such systems can be considered “COTS-based systems,” as most of their executing instructions come from COTS products, but COTS considerations do not affect the development process very much.

To provide a focus on the types of applications for which COTS considerations do affect the development process, we define a COTS-Based Application as a system for which at least 30% of the end-user functionality (in terms of functional elements: inputs, outputs, queries, external interfaces, internal files) is provided by COTS products, and at least 10% of the development effort is devoted to COTS considerations. The numbers 30% and 10% are not sacred quantities, but approximate behavioral CBA boundaries observed in the application projects. There was a significant gap observed in COTS-related effort reporting. The projects observed either reported less than 2% or over 10% COTS-related effort, but never between 2-10%.

In our six years of iteratively defining, developing, gathering project data for, and calibrating COTCS cost estimation model, we identified four primary sources of project effort due to CBA development considerations. These are defined in COCOTS as follows:
- COTS Assessment (A) is the activity whereby COTS products are evaluated and selected as viable components for a user application.
- COTS Tailoring (T) is the activity whereby COTS software products are configured for use in a specific context. This definition is similar to the SEI definition of “tailoring” [10].
- COTS Glue Code (G) development and integration is the activity whereby code is designed, developed, and used to ensure that COTS products satisfactorily interoperate in support of the user application.

1.2 CBA Growth Trend
An increasing fraction of CBA projects have been observed in over five years’ USC-CSE e-services project data. As seen in figure 2.1, the CBA fraction has increased from 28% in 1997 to 60% in 2001. Major considerations for adopting COTS products in these projects are: 1) the clients’ request, 2) the schedule constraint, 3) compliance with organization standards, and 4) the budget constraint. The primary reason for the growth in COTS content has, however, been the large increase in the number of COTS products providing application functions. In 1997, most of the teams were programming their own search engines and Web crawlers, for example; by 2001 these functions were being accomplished by COTS products.

Some of our USC-CSE affiliates have reported similar qualitative trends, but this is the first quantitative data they and we have seen on the rate of increase of CBA projects under any consistent definition and in any application sector (e-services applications probably have higher rates of increase than many other sectors). We have experienced many notable effects of this increase: for example, programming skills are necessary but not sufficient for developing CBA’s (see also 8,9,10,11).
2. The CBA Process Decision Framework

Figure 2.1 presents the dominant decisions and activities within CBA development as abstracted from our observations and analysis of USC e-services and CSE-affiliate projects. This represents the overall CBA decision framework that composes the assessment, tailoring, glue code, and custom code development process elements within an overall development lifecycle.

Some explanation of Figure 2.1 is in order. The CBA process is undertaken by “walking” a path from “start” to “Non-CBA Activities” that connects (via arrows) activities as indicated by boxes and decisions that are indicated by ovals. Activities result in information that is passed on as input to either another activity or used to make a decision. Information follows the path that best describes the activity or decision output. Only one labeled path may be taken at any given time for any particular walk; however it is possible to perform multiple activities simultaneously (e.g. developing custom application code and glue code, multiple developers assessing or tailoring).

The small circles with letters A, T, G, C indicate the assessment, tailoring, glue code, and custom code development process elements respectively. With the exception of the latter, each of these areas will be expanded and elaborated in the sections that follow. Each area may enter and exit in numerous ways both from within the area itself or by following the decision framework of Figure 2.1. In addition, this scheme was developed from and is consistent with the CBA activity distributions of Figures 2.3. In particular, only (and in fact all) “legal” distributions are possible (e.g. that all distributions have assessment effort is consistent with all paths in the framework initially passing through the assessment element (or area “A”). We now summarize the less obvious aspects of each process area.

**Figure 1.2** CBA Growth in Small E-Service Projects

**Figure 2.1. CBA Effort Decision Framework**

**P1: Identify OC&P’s: Evaluation Criteria, Weights and Scenarios.** This is the entrance to the CBA process where the initial evaluation attributes and desired operational outcomes for the application are established. Risk considerations, stakeholders’ priority changes, new COTS releases and other dynamic considerations may significantly alter the objectives, constraints, and priorities (OC&P’s). In particular, if no suitable COTS packages are identified, the stakeholders may change the OC&P’s and the process is started over with these new considerations.

**P2: Identify alternatives: Candidate COTS products.** This and activity P1 establish the entry conditions for an Assessment activity.

**P5: Multiple COTS cover all OC & P’s?** If a combination of COTS products can satisfy all the OC&P’s, they are integrated via glue-code. Otherwise, COTS packages are combined to cover as much of the OC&P’s as feasible and then custom code is developed to cover what remains.

**P6: Can Adjust OC & P’s?**
When no acceptable COTS products can be identified, the OC&P’s are re-examined for areas that may allow more options. Are there constraints and priorities that may be relaxed that have eliminated some products from consideration? How firm are the objectives and if adjusted slightly will it enable consideration of more products? Are there analogous areas in which to look for more products and alternatives?

**P8: Coordinate Application Code development and Glue Code effort.** Custom developed components must eventually be integrated with the chosen COTS products. The interfaces will need to be developed so they are compatible with the COTS products and the particular...
glue code connectors used. This means that some glue code effort will need to be coordinated with the custom development.

3. Example WinWin Spiral Approach to CBA Development

3.1. Elaborated WinWin Spiral Model

Figure 3.1 provides a more detailed and concise version of the Win Win Spiral Model than that presented in [5]. It returns to the original four segments of the spiral, and adds stakeholders’ win-win elements in appropriate places. It also emphasizes concurrent product and process development, verification and validation; adds priorities to stakeholders’ identification of objectives and constraints; and includes the LCO, LCA, and IOC anchor point milestones [19] also adopted by the Rational Unified Process.

3.2. Example CBA: Oversize Image Viewer

One of the USC e-services COTS-based applications involved the development of a viewing capability for oversized images. The original client needed a system to support viewing of digitized collections of old historical newspapers, but other users became interested in the capability for dealing with maps, art works and other large digitized images. The full system capability included not just image navigation and zoom-in/zoom-out; but image catalog and metadata storage, update, search, and browse; image archive management; and access administration capabilities.

Several COTS products were available for the image processing functions, each with its strengths and weaknesses. None could cover the full system capability, although other COTS capabilities were available for some of these. As the initial operational capability (IOC) was to be developed as a student project, its scope needed to be accomplished by a five-person development team in 24 weeks. The application described in the next section makes some small simplifications of the project for the sake of brevity, but the overall COTS decision sequence and spiral cycles happened largely as described.

3.3. Applying the Decision Framework and the WinWin Spiral Model

The process description provided here for the Oversize Image Viewer (OIV) project covers the project’s first three spiral cycles. Each cycle description begins with its use of the WinWin Spiral Model, as the primary sequencing of tasks is driven by the success-critical stakeholders’ win conditions and the project’s major risk items.

The OIV process description for each cycle then discusses its use of the CBA Process Decision Framework and its process elements. It shows that the framework is not used sequentially, but can be re-entered if the Win Win Spiral risk patterns cause a previous COTS decision to be reconsidered. The resulting CBA decision sequence for the OIV project was a composite process, requiring all four of the Assessment, Tailoring, Glue Code, and Development process elements.

Table 1 provides a spiral model template that is an update of the template used in the original spiral model paper [5]. It shows the major spiral artifacts and activities in the OIV project’s first three spiral cycles. The discussion below indicates how these were determined by the major stakeholder OC&P’s and project risk items.

3.3.1. Spiral Cycle 1

The original client was a USC librarian whose collections included access to some recently-digitized newspapers covering the early history of Los Angeles. Her main problem was that the newspapers were too large to fit on mainstream computer screens. She was aware that some COTS products were available to do this. She wanted the student developer team to identify the best COTS product to use, and to integrate it into a service for accessing the newspapers’ content, covering the full system capability described in section 4.2 above. Lower priorities involved potential additions for text search, usage monitoring, and trend analysis.

Her manager, who served as the customer, had two top-priority system constraints as her primary win conditions. One was to keep the cost of the COTS product below $25K. The other was to get reasonably mature COTS products with at least 5 existing supported customers.

The student developer team’s top-priority constraint was to ensure that the system’s Initial Operational Capability (IOC) was scoped to be developable within the 24 weeks they had for the project.

The team quickly used these top-priority constraints to filter out two COTS products: system XYZ was too expensive, and system ABC had only one beta-test customer. The other two OIV COTS products, ER Mapper and Mr. SID, had different user interfaces; the major risk was to select one that users would subsequently find unacceptable. This risk was addressed by exercising the two products; this stage of the COTS assessment concluded that ER Mapper had considerably
stronger performance and image navigation characteristics than Mr. SID. Mr SID’s main advantage was that it ran on Windows, Unix, and Macintosh platforms, while ER Mapper was only running on Windows. As the client had a Windows-based operation, ER Mapper was identified as the best candidate. Plans were made to tailor it for the overall product solution, and integrate it with other COTS and/or application code, as ER Mapper was not a complete application solution for such functions as cataloguing and search.

When the customer reviewed these plans, however, she felt that the investment in a campus OIV capability should also benefit other campus users, some of whom worked on Unix and Macintosh platforms. She committed to find representatives of these communities to participate in a re-evaluation of ER Mapper and Mr. SID for campus-wide OIV use. The client and developers concurred with this revised plan for spiral cycle 2.

Use of the CBA Decision Framework in Cycle 1
The first three steps of spiral cycle 1 in Table 1 (Stakeholders, OC&P’s, Alternatives) include COTS products as alternatives and establish the preconditions (top-level evaluation criteria, weights, and scenarios; candidate COTS products) for entering the CBA Assessment decision framework in Figure 2.1 and 3.2. Spiral step 4 (Evaluation in Table 1) establishes the entry into Assessment in Figures 3.1 and 3.2.

Following the Assessment Framework in Figure 3.2, the initial filtering step eliminated some candidates (XYZ and ABC), but not ER Mapper or Mr. SID. The risk assessment in Table 1 required the two COTS products to be exercised, which involved Tailoring to accommodate the newspaper image files, but not glue code at this point. The evaluation identified ER Mapper as the best OIV solution, but only as a partial solution for other needed functions such as cataloguing, search, and archiving.

Thus the Assessment process element (Figure 3.2) exits back to the overall CBA decision Framework (Figure 2.1) in the “Partial COTS solution best” direction. But it cannot proceed further until the Win Win Spiral process determines whether either applications code or added COTS products or both need to be developed for the rest of the application (a lower risk decision deferred to a subsequent spiral cycle).

However, spiral cycle 1 ended with a new decision to revisit Assessment with likely new OC&P’s emerging from other-OIV-user stakeholders as evaluation criteria. Thus we can see that the CBA decision framework is not sequential, but needs to be recursive and reentrant depending on risk and OC&P decisions made within the Win Win Spiral process.

3.3.2. Spiral Cycle 2
With the new Unix and Mac OIV stakeholders, a new win-win set of OC&P’s emerges, including not only Unix and Mac OIV usability but also interoperability with other selected COTS products on all three platforms. The new evaluation/COTS assessment confirmed that Mr. SID was usable on all three platforms, but that ER Mapper had only general plans for Unix and Mac versions.

When ER Mapper declined to guarantee early Unix and Mac versions, Mr. SID became the new choice for the OIV functions. Concurrent assessment of candidate COTS products for the non-OIV functions converged on MySQL for catalog database support and Java for GUI support. Although the initial evaluation indicated that these were interoperable with Mr. SID, a fully interoperable build-upon (vs. throwaway) prototype was scheduled to be developed and interoperability-verified in spiral cycle 3. The other outstanding risk identified was that the system’s GUI needed prototyping with additional end-user representatives also planned for spiral cycle 3.

Spiral cycle 2 ended with a WinWin Spiral LCO (Life Cycle Objectives) milestone review. At the LCO review, all of the stakeholders agreed to support the commitments allocated to them in the plans.

Table 1. Spiral Model Application to
Use of the CBA Decision Framework in Cycle 2

The new stakeholders and OC&P’s in cycle 2 required the project to backtrack to the beginning of the Assessment process element in Figure 2.1 and 3.2. For the OIV function, ER Mapper was filtered out without further evaluation when it declined to guarantee early Unix and Mac versions. Some tailoring was required to verify that Mr. SID performed satisfactorily on Unix and Mac platforms.

Concurrently, Assessment filtering and evaluation tasks were being performed for the cataloguing and GUI functions.

This concurrency is a necessary attribute of most current and future CBA processes. Simple deterministic process representations are simply inadequate to address the dynamism, time-criticality, and varying risk/opportunity patterns of such CBA’s. However, the Win Win spiral process provides a workable framework for dealing with risk-driven concurrency, and the composable CBA decision framework and process elements provide workable approaches for handling the associated CBA activities. The dynamism and concurrency makes it clear that the CBA process elements need to be recursive and reentrant, but they provide a much-needed structure for managing the associated complexity.

3.3.2. Spiral Cycle 3

The additional end-user stakeholder communities increased the risk of developing GUI’s that were fine for some users and unsatisfactory to others. These risks were resolved by involving representative end users in exercising GUI prototypes for various cataloguing, search, and navigation functions. The major CBA processes involved the Assessment of detailed interoperability characteristics of Mr. SID, MySQL, and the GUI software on the Windows, Unix, and Mac platforms. This involved invocation of both the Tailoring and Glue Code process elements.

The other major risk was the fixed 24-week IOC development schedule. This was handled via the Schedule as Independent Variable (SAIV) process described in [18]. The SAIV process requires customers and users to prioritize their desired capabilities. The priorities are used to define a core capability clearly buildable within the fixed schedule, and to architect the application for ease of adding or dropping borderline-priority features. This approach was satisfactory to the stakeholders, and resulted in a successfully transitioned Initial Operational Capability at the end of the 24 weeks.

Use of the CBA Decision Framework in Cycle 3

The Assessment process for interoperability of Mr SID, My SQL, and the Java GUI components on the Windows, Unix, and Mac platforms did not involve a comparative evaluation of alternative COTS products, although alternatives would have been necessary in case one of the COTS products had proved completely inadequate. The interoperability assessment involved both tailoring of the COTS products for the three platforms and some glue code to (successfully) enable interoperability.

Subsequent spiral cycles to develop the core capability and the IOC did not involve further Assessment, but involved concurrent use of the Tailoring, Glue Code, and custom development processes.

3.4. Summary of CBA Decision Framework Use

The use of the CBA decision framework during the three spiral system definition cycles and the subsequent development activity can be summarized by the sequence A, T; (AA); A, (TG); (TGC). The first spiral cycle involved Assessment supported by Tailoring. The second cycle involved two concurrent pure Assessments for the OIV COTS choice and for the other COTS choices. The third cycle involved an interoperability Assessment supported by concurrent Tailoring and Glue Code processes. The final development activity involved concurrent Tailoring, Glue Code, and custom development processes.

4. Conclusions

Using the WinWin Spiral Model’s risk-driven approach coupled with the CBA decision framework as a process model generator, however, enabled projects to generate appropriate combinations of A, T, G, and C process elements that best fit their project situation and dynamics. An extensive discussion of its application to an actual CBA project is provided as an example.

The resulting combinations of A, T, G, and C elements serve as a sort of genetic code for the projects CBA process which can be used to identify and compare it with other projects CBA processes. The analogy can be stretched too far, but it suggests several attractive directions for future research, such as determining how best to represent the concurrency and backtracking aspects; validating and refining effort distributions based on process elements; assessing the validity of the process elements and decision framework in other CBA sectors; and identifying common process element configurations, valid and invalid configurations, or large-grain CBA process patterns.

5. References