Using Business Scenarios to Surface Requirements for COTS Products

WenQian Liu
Dept. of Computer Science
University of Toronto
wl@cs.utoronto.ca

Sharon Lymer
IBM Toronto Software Lab
IBM Canada Ltd.
lymer@ca.ibm.com

Steve Easterbrook
Dept. of Computer Science
University of Toronto
sme@cs.utoronto.ca

Abstract

Constructing software by integrating commercial off-the-shelf (COTS) products is widely practised, particularly in the IT service industry. For vendors of COTS products, requirements engineering is particularly challenging. To continually improve their products, vendors must identify and analyze problems that occur when their products are used in a wide variety of integrated solutions, and they must anticipate new applications in which their products could be used. In this paper, we describe a scenario-based framework developed at the Software Group division of IBM Corporation (IBM SWG) that mimics the solution integration process for new business opportunities, allowing the development teams to evaluate their products, discover and resolve integration issues, and to surface new requirements for future releases. The paper describes the framework, with an example business scenario, and discusses the experience of using this framework at IBM SWG and the lessons learned.

1. Introduction

In order to reduce development costs, companies in many industries have reduced the size of in-house development teams and increased outsourcing for building customized applications for their own business. The use of middleware and commercial off-the-shelf (COTS) products tends to dominate over building from scratch in such development projects. This approach is commonly known as *software integration*, and a system that is put together using multiple COTS products is an *integrated solution*.

In this paper, we focus on the problem of maintaining and evolving the COTS products themselves, and in particular the problem of surfacing new requirements for such products. For example, IBM's Software Group (SWG) produces a portfolio of COTS products from the WebSphere®, DB2®, Lotus®, Tivoli®, and Rational® brands. These provide a wide range of capability, including application and process integration, information management, collaboration, systems management, and software development.

Managing the evolution of such a broad portfolio of COTS products presents many new challenges for requirements elicitation and management:

- Interface compatibility between COTS products.
- Capability overlaps or gaps.
- Time to value – COTS products need low set-up time, low installation effort, short learning curve, and be easy to tailor and incorporate into a solution.
- Usability – Different look and feel, different targeted user groups with differing skill sets and learning requirements, and discontinuities in the flow as solution tasks are executed across products, lower the overall usability.

Although the use of COTS products may cut down the development cost of individual software components, extra effort is required to deal with the kinds of integration issues listed above. COTS customers are rapidly discovering these hidden costs, leading to a demand for improvements to COTS products that can reduce the overall effort needed to build integrated solutions. This demand increases the pressure on software vendors and IT service providers to change their own development paradigm.

At IBM SWG, we responded to these challenges by creating a team to use customer solutions to understand the integration issues that arise when building solutions with COTS software. Furthermore, we needed to gain insights into the full customer development life cycle using our COTS products by exploring how individual integration solutions are envisioned, designed, implemented, tested, deployed, and managed.

Our approach, the *Business Scenarios Framework (BSF)*, is similar to scenario-based approaches in the literature [11]. However, we expand the idea of a scenario to incorporate not just a particular way of using a product, but the entire cycle by which a set of products might be integrated and deployed in response to a business need. The business scenario represents a particular business situation and the integrated software solution required to accomplish a set of business goals. It describes the solution architecture and

maps components to both current and future COTS products. Most importantly, it provides a framework for acting like the solution builder throughout the solution development process and for evaluating the COTS products as they are customized, integrated, deployed, and managed. Collectively, the business scenarios help IBM SWG product teams to not only identify integration issues, but also surface requirements for the next generation COTS products.

Currently, we have developed nine business scenarios. In this paper we will describe the development of one of these in detail, namely the Employee Workplace business scenario. This scenario addresses the problem that, in many businesses, employees struggle to manage content, to locate and access information, and to overcome technological challenges. We will describe BSF itself, illustrate its use with the Employee Workplace scenario, and then discuss our experiences in applying BSF at IBM SWG.

2. Background

Today, user-centred design and requirements engineering methodologies deal with identifying and stating software requirements. User-centred design in particular, emphasizes understanding and meeting the needs of human users with strong focus on the design and implementation of the user interface. Requirements engineering identifies the goals of stakeholders and precisely elaborates the desired application behaviour to determine requirements. With the focus on human users and application-level needs, requirement elicitation methods from these approaches are not well-suited to understanding and determining COTS requirements. For example, there is little or no guidance for gathering requirements for COTS products that are to be used by large numbers of broader solutions.

Much of the literature on COTS products in software engineering is oriented towards the COTS consumer, providing advice on procurement (e.g., see [1, 2]). Various frameworks have been proposed for the evaluation and selection of COTS components [5, 6, 10], and alternative development processes are proposed to effectively leverage the needs of COTS-based system construction [9, 3], Recent reports have also described the experiences and lessons learned in integrating COTS components into systems (e.g., [4]). Our perspective is different from these in that we come from the vendor's point of view, and our concern is with managing the requirements of COTS products to determine what goes into the future releases.

Unlike application-level software that tends to have well-understood human users and usage, COTS products often have both system and human users, and usage scenarios that are as varied as the applications that utilize the COTS products. Many additional requirements arise from the concerns of the systems integrator, who must deal with conflicting software dependencies and prerequisites, duplicate components, mismatched architectures, and missing or incomplete functionality coverage across the COTS products required in an integrated solution. This makes the process of requirements elicitation much more difficult.

Another challenge is the need for a COTS product to integrate and inter-operate with other COTS products to produce the overall capability needed by an integrated solution. Again, the problem is the multiplicity of solutions. Several solutions may require the same underlying COTS products but each solution drives a different set of usage patterns and combinations of interactions across the middleware. Different non-functional requirements and constraints are highlighted by each solution.

Finally, it is essential that COTS vendors understand not only the runtime requirements of their product, but also the other aspects of its use. For example, what tools are provided to aid with overall solution development? What are the administrative interfaces that assist with the management of the solution? What programming model is used to access the COTS function? What are the required performance characteristics?

Our approach provides a systematic framework for eliciting requirements that arise from all these concerns. Requirements elicitation techniques are usually conceived as suitable for one of three types of software development: building custom software for a single customer; creating generic software that is targeted for mass market; or procuring COTS products and developing glueware that will serve in the company's own business context [7]. Our approach combines aspects of all three of these contexts. Our COTS products themselves are intended for a mass market, but the requirements can only be elicited by considering how the products will be procured and integrated by a specific customer, and the problems that may arise when a solution provider attempts to use them for a custom solution.

BSF is an adaptation of scenario-based methods that have been widely used in requirements engineering and software design [11]. The contexts within which scenarios are used are diverse, ranging from the social environment of the system to the event sequencing in a design [8]. The purpose of using scenarios is typically to describe the "continuum from the real world descriptions and stories to models and specifications" [11]. Our framework is a variation of scenario-based methods. A business scenario includes a COTS-based software application and its development life cycle; it includes both the business context of the system and detailed event sequences within the system.

3. Business Scenario Framework (BSF)

The business scenario approach was introduced by IBM SWG to better understand their customers and

the integration-intensive software development projects they undertake. We use the Business Scenario Framework (BSF) to provide a basis for documenting and investigating integrated solutions and mimicking the development activities required to produce the solution.

The core of the framework is the *business scenario* – a representation of a solution desired by COTS products customers. It tends to be a generalized or prototypical solution that illustrates the typical needs of similar businesses to solve a particular set of business problems. It describes the ways businesses currently use or want to use COTS products to accomplish specific business goals.

BSF consists of the business scenario artifacts and a defined process. The process covers the production of the artifacts and the activities for analyzing and evaluating the COTS products. An important part of the process is an emulation of a solution builder through a typical development cycle from requirements gathering and architectural design to implementation and testing. The real solution builder would be either the IT organization for the COTS software customer or an IT services provider. Since the purpose of emulating the COTS software integration life cycle is to test the compatibility and inter-operability of participating COTS products to ensure they integrate well, just enough of a solution is constructed to evaluate the business scenario; we do not build a comprehensive integrated solution.

3.1. Artifacts

The BSF artifacts are *Solution Definition, Solution Builder Critique, Architecture, Design, Analysis and Evaluation Criteria, COTS Product Use Cases, Implementation Assets, Test Cases,* and *Results*.

The *Solution Definition* describes the business opportunity and the business problems to be solved by the integrated solution for the selected business situation. The software users are identified and described. The user experience with the integrated solution is highlighted through usage stories and described in more detail through use cases. The desired features are outlined and a solution diagram is included. In addition, constraints, assumptions and dependencies are identified and non-functional requirements such as those for usability, performance, or reliability are defined.

It is important to review and validate the information in the solution definition with appropriate solution builders. The responses and comments from validation sessions are documented in the *Solution Builder Critique*.

The architecture document and associated UML models form the *Architecture* artifact for the BSF. Architectural goals and constraints are investigated and detailed. The key items in the domain model for the integrated solution are identified and modelled using UML class diagrams. The logical view of the architecture presents the proposed subsystems. The necessary subsystem components are determined by producing use case realizations expressed as sequence diagrams for any use case with architectural significance. The *Design* artifacts (document and associated models) revisit and refine aspects of the architectural artifacts and provide additional design perspectives through the process, deployment, data, and implementation views.

The *Analysis and Evaluation Criteria* is crucial. This document identifies available COTS products that correspond to subsystems or components of the integrated solution, selects the specific releases of the COTS products to be used in further solution builder emulation activities and describes areas of focus for analysis and evaluation. The features of the integrated solution are prioritized for investigation and important functions or capability are highlighted. In addition, the solution development facets such as installation and configuration, solution development, and security are selected for the exploration and evaluation. Important use cases are identified and specific evaluation criteria for each facet are defined. Exploration of the solution development facets leads to the identification and development of relevant *COTS Product Use Cases*.

Emulating solution development activities as part of the BSF process leads to a partial *Implementation* of the integrated solution. Any code resulting from the implementation phase is provided as part of the BSF. In preparation of the test phase of the BSF process, *Test Cases* are developed and included as BSF artifacts.

Finally, the outcome of the analysis and evaluation activities are provided in the *Results* artifact. There are two parts: an experience report and a set of recommendations. The experience report presents an evaluation of the COTS products used in the various emulated solution builder activities and discusses the level of satisfaction with the experience, with emphasis on any inhibitors. The Recommendations suggests improvements to COTS product inter-operability within the context of the integrated solution and in some cases, recommend enhancements or changes to the COTS products to better meet the needs of the solution.

3.2. Process

The process begins with the identification of a business situation requiring a software solution that uses at least one COTS product of interest to the provider. A key aspect of the process is to act as solution builders throughout the development process and mimic their activities at each stage. Figure 1 shows how emulated activities are interspersed with analysis activities that help with determining or evaluating the capability and behaviour of the COTS products.

We start by analyzing the business situation, determining the needs, and defining an integrated solution. It is essential to validate this definition. Appropriate COTS soft-

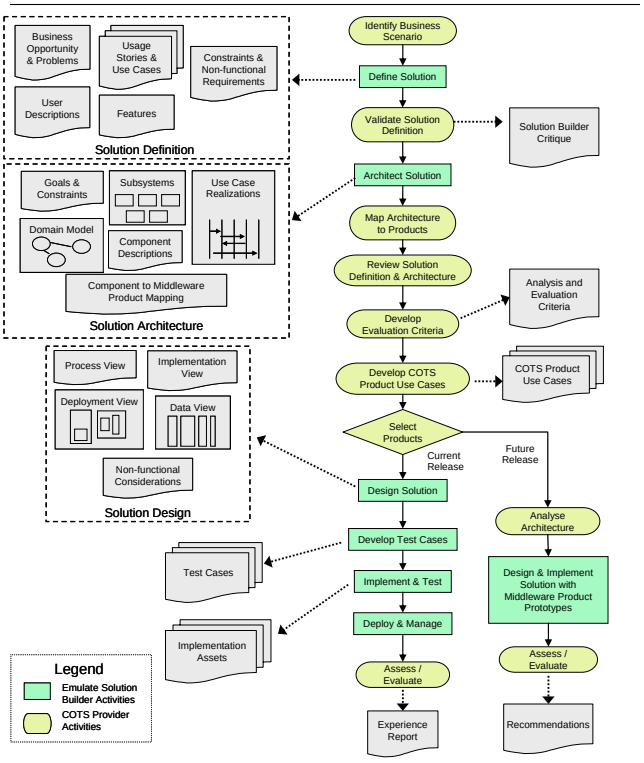

Figure 1. The BSF Process.

ware customers or IT service providers are identified. The solution definition is validated with them on how well it reflects the business needs identified, how accurately the business problems are represented, and whether the stated features and capabilities meet their needs.

Next, we produce an architecture that involves:

- identifying main objects in the business domain such as a customer order,
- identifying the use cases that affect the architecture,
- determining the logical subsystems, their major components and interfaces, and
- describing and exploring realizations of significant use cases using sequence diagrams.

Then, we map our current and future COTS products to the subsystems and components identified in the software solution. The choice of COTS products influences the solution development life cycle. Exploring the effects of each choice is essential to understanding COTS product requirements and reducing integration effort. Such effects include new constraints and utilities, and in general, new ways of handling various facets of solution development. These facets include: solution evaluation and planning, installation and configuration, solution development, solution management, problem determination, and proficiency of serviceability, support, migration, re-engineering, quality of service, and security.

The effects are explicitly identified for each solution and evaluation criteria defined for more rigorous analysis and assessment in minimizing the impact to the overall solution development process. In particular, new and changed solution builder tasks, as a result of the introduction of the COTS products, are determined and documented as COTS product use cases.

Next, we act like the solution builder and design the integrated solution with a set of released COTS products. Typically, this involves designing the key elements of the software solution and producing a partial implementation. The design includes the identification of subsystems, packages and classes, and the definition of the runtime architecture in terms of processes and threads, its physical distribution, implementation layers, and its persistent data. The design is documented and reviewed to ensure its consistency with the solution definition and architecture developed earlier, and to verify its emphasis on desired focus areas and high priority features. In addition, test cases are provided to verify these aspects.

Then, a partial implementation of the software solution is constructed according to this design. Focus is on the exploration and evaluation of the COTS development utilities and environment using the solution development facet evaluation criteria produced earlier in the process. In particular, the COTS product use cases associated with the solution development facet are exercised and evaluated.

Next, we return to emulating the solution builder and test the partial implementation of the integrated solution. The test phase is not intended to eliminate defects from the software solution but rather to assess the operation of the COTS products within the solution. Again the evaluation criteria are used to assess solution development facets such as the installation, configuration, and the problem determination features of the COTS products.

Finally, the software solution is deployed to and managed in a test environment that is configured similarly to a typical production environment. As with the test phase, the focus of this step is to exercise the COTS product use cases and evaluate any effects on the solution management, quality of service, and security facets.

Throughout the process, the experience with the COTS software is evaluated, assessed and documented in an experience report. Product defects are reported and the resolution is tracked using the same support channel as an actual COTS product customer. The experience report and defects are the key results of this part of the process.

We use the same approach to improve the design of future product releases. In this case, early designs or prototypes for future products are used during the emulation of the design phase. Naturally, it is not possible to realistically

mimic solution builder activities in this situation, so more analysis is required rather than explicit evaluation. Typically, an architectural analysis is conducted to explore the integration of the COTS software within the solution. This results in the identification of integration issues or gaps. In addition, early product prototypes are used to investigate the affects of the COTS on the other facets of the solution development life cycle such as utilities for implementing, deploying, or managing the solution. The outcome of the exploration is a set of recommendations for enhancing the planned product or the addition of new features to close any identified gaps. These recommendations are documented and communicated to the COTS development teams for consideration in their requirements process.

3.3. Roles

Developing a business scenario requires time and resource. However, this effort is well spent since it results in the elimination of costly integration problems that might otherwise remain undiscovered until late in the COTS product development cycle. The BSF approach demands particular skills and experience. The following roles are recommended.

The *Business Scenario Architect* identifies use cases and features for the solution definition, develops the architecture and the analysis and evaluation criteria, and oversees analysis and evaluation activities. This is usually a senior architect with broad technical expertise and in-depth knowledge of the COTS products associated with the scenario.

The *Program Manager* produces the solution definition, arranges its validation and produces the Solution Builder Critique. Schedules and tracks activities, and handles communication with COTS product development teams. Project management expertise and a strong technical background are important for this role.

The *Business Scenario Developer* develops solution and COTS product use cases, produces the Design artifact under direction of the business scenario architect and implements a partial solution using a particular set of COTS products. If future products are being evaluated, then the implementation is done with early product prototypes. A representative set of solution test cases are produced to exercise key aspects of the solution. The primary goal of the business scenario developer is to conduct analysis and evaluation of the COTS products throughout the implementation, test, management, and deployment phases of the solution development cycle. Typically, this role is given to an experienced software developer.

The complete business scenario process typically involves a business scenario architect, a program manager, and one or two business scenario developers. With this size team, the overall process from business scenario identification to the production of an experience report takes approximately six to nine months. Naturally, the solution implementation and test activities need to be aligned with the availability of the desired products. If the business scenario is used in the analysis of future products, the activities leading to the solution architecture take three to four months and the design and analysis activities are constrained to about three months.

3.4. Example

Teams at IBM SWG started using the business scenario approach about three years ago. Initially, only the WebSphere brand COTS products were explored. However, in the last year the business scenarios began to include COTS products that span across the IBM Software portfolio. Currently, we have developed nine business scenarios. These include:

Employee Workplace describes a solution that enhances employee effectiveness and productivity by providing employees with self-service applications and makes it easier for them to collaborate across departments and locations. It also improves the employee's ability to effectively manage the growing volumes of corporate information regardless of format, such as documents, e-mail, digital images, Web content.

Mergers and Acquisitions describes the situation of a merger between a large traditional insurance company providing insurance through agents and a new Internet operation working entirely through the Web. The solution provides a single customer view of the merged companies to give a single administrative view and cut operational costs.

Private Exchange examines a supplier's use of a Business-to-business (B2B) exchange to streamline the delivery of product information to potential buyers. The solution centres on the automatic processing and publishing of product information.

We will use one of our more recent business scenarios, Employee Workplace (EW), to illustrate some of the important aspects of the BSF artifacts. The actual business scenario artifacts are too extensive to present in this paper. Instead, sample text or diagrams have been extracted from the Employee Workplace artifacts to provide a flavour of these artifacts and the kind of content presented in them.

Employee Workplace provides a solution to the many challenges businesses face in an effort to have a more efficient and productive organization and to deal with increasing volumes of digital content. The business perspective for the solution is provided in the solution definition artifact and includes a description of the business problems that need to be addressed, the proposed features of the solution, its users and use cases.

The primary business problems fall into four categories: (i) Employee productivity inhibitors; (ii) Difficulties managing information; (iii) Difficulties finding information; (iv) Business risks.

The key features of the Employee Workplace solution are:

- Collaborative capability: e-mail, instant messaging, calendar, and Web conferences
- Document and Web content management, including life-cycle management and search
- Task management
- Records and retention management
- Employee directory
- Application integration services

There are three primary types of users of the Employee Workplace solution: employees who use the workplace to carry out various roles within the organization; records managers responsible for creating and administering the records and retention management system; and solution administrators who maintain the running solution.

There are several use cases for the Employee Workplace solution. Some of the more important use cases are:

- Collaborating with other employees using instant messaging and e-mail
- Creating, browsing and editing documents or Web content
- Searching for content
- Creating content life cycles and completing tasks associated with the life cycle
- Creating and maintaining a retention plan including creating, viewing, suspending, and destroying records.
- Managing access to users, groups, and resources
- Backing up and restoring repositories

The Employee Workplace solution definition was reviewed with a few major IBM customers. These customers agreed that the solution definition reflected the reality of their businesses and accurately described the business problems and solution needs. This discussion and comments were documented in the *Solution Builder Critique*, but cannot be published to protect the privacy of our customers.

Another major artifact is the architecture of the business scenario. The first section in this artifact is the discussion of architectural goals and constraints. Some of the goals and constraints for the Employee Workplace architecture include a single point of access to business services, security, and ease of solution administration.

An important part of the architecture is the domain model, which is described using UML class diagrams. A

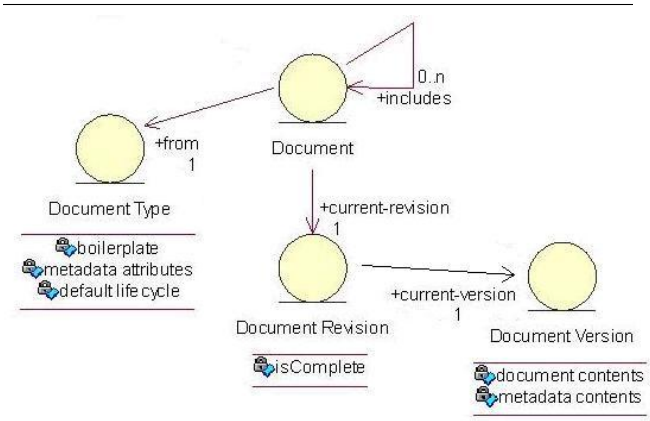

Figure 2. Document Types

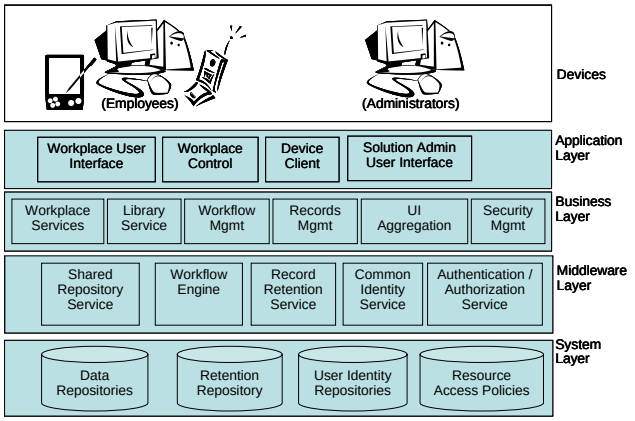

Figure 3. Subsystems and Components

key business object for Employee Workplace is a *document type*, which is prescribed format that each document conforms to (see Figure 2). It may specify required metadata attributes and a boilerplate for the document body. For an owners manual, the metadata attributes may be the car's make and model, and the boilerplate may be the layout of chapter and section headings such as Introduction, Maintenance Schedule, etc.

The EW solution has a number of subsystems and components as shown in Figure 3. Each subsystem is described in detail in the actual *Architecture* artifact.

A key part of the architecture is the exploration of significant use cases in terms of their realizations as described in the solution definition. For example, "*Create new document revision*" is a use case and its realization is shown in Figure 4 using UML sequence diagram.

Finally, the architecture includes a mapping of both current and future IBM software products that correspond to components in the subsystem diagram. The EW ar-

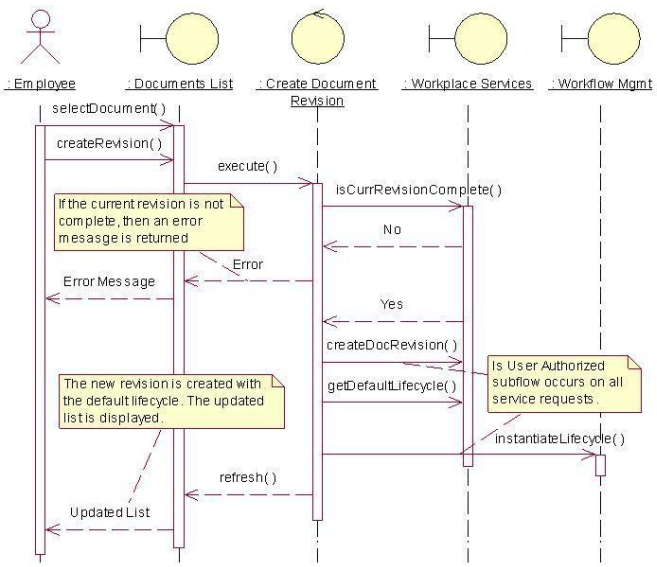

Figure 4. Use Case Realization

chitecture includes the current[1] IBM products: IBM® Workplace Team Collaboration™, Messaging and Documents 2.0, DB2® Content Manager 8.2, DB2 Records Manager 3.1.2, WebSphere® Portal 5.0.2, and Tivoli® Directory Server 5.1.

The design artifact is a document that discusses the various perspectives of the Employee Workplace design using a particular set of IBM COTS products as the components and covers the implementation, deployment, and data views. This material is too detailed to include in the paper.

The analysis and evaluation criteria are documented for the selected product sets. The criteria are presented by feature and also by solution development facet. Based on the usage stories in the solution definition, important capabilities for each feature are identified and prioritized for evaluation. Expectations for each product set are noted, including any known limitations. In addition, this artifact provides a detailed discussion for each solution development facet. In Employee Workplace, the installation and configuration facet was selected, since the IBM® Workplace™ family is a relatively new addition to the IBM software portfolio. The solution management facet was selected because of the large numbers of products in the solution and the goal of ease of administration. The security facet was selected because it is important for employees to be able to transparently access the services offered by the various COTS products constituting the solution with a single sign-on. Finally, the solution development facet was selected, because we wanted to explore the ability for solution builders to integrate existing systems into the Employee Workplace and to develop the types of content that need to be incorporated in the solution.

The COTS product use cases and the corresponding users are identified and elaborated for each solution development facet. For example, the use cases for Employee Workplace solution development include:

- Develop, test, and deploy templates for Web content presentation.
- Develop and test portlets for application integration.
- Develop and test new workplace applications such as departmental applications, expense reporting, etc.

The implementation assets for Employee Workplace were created in Java™ code and are not included this paper. The test cases are also too extensive too include, but follow a typical format. However, there are a limited number since the purpose of the testing is not to evaluate the solution implementation. Rather the goal is to test the interoperability of the COTS products in the solution and to exercise the important features as identified in the analysis and evaluation criteria.

The Employee Workplace business scenario effort involved the evaluation of recently released IBM COTS products and also the analysis of up-coming products. Based on the selected solution development facets, the experience report evaluated the experience installing, configuring, developing, managing, and securing the COTS products in the context of the Employee Workplace. Recommendations for more effective interoperability with future releases have been conveyed to the product development teams.

4. Lessons Learned

The lessons learned in our application of the Business Scenario Framework (BSF) are based on our observations and include areas for improvement and identification of limitations that need to be addressed in future.

Lesson 1 *Using business scenarios, we are able to improve the design and quality of our COTS products. Product developers gain a deeper understanding of the customer requirements that affect product usage and behaviour by considering the concrete business situations and the various development perspectives offered in the business scenario.*

Without the information provided by business scenarios, development teams tend to take a product-centric approach and typically have little opportunity to explore the integration requirements that arise when the COTS product is used with others in a solution. There is a tendency to focus on the production usage rather than to consider product requirements arising from software development facets such as product installation and configuration, using the product

1 We cannot disclose information about planned future IBM products.

to develop and manage the integrated solution, and the affects of their product on the overall solution quality.

Business scenarios provide two perspectives to the developers. The first provides the context of use – a spectrum of applications of the product. The second highlights the spectrum of solution development activities as the COTS product is used throughout the integrated solution development life cycle from requirements gathering to implementation, test, and deployment.

Typically, the COTS product is designed to serve a particular need and thus the set of required capability, such as messaging or database services, seems clear. However, unless various integrated solutions are considered, interoperability challenges may be missed. For instance, some COTS software provides underlying services to other COTS software in the solution. For example, database services may be required by application services or content management. It is essential that each COTS product have the same prerequisite level of the underlying COTS software. We have found the prerequisite releases to be inconsistent in many cases. Similarly, the configuration of shared services needs to be consistent across the integrated solution. Often, we have found that different configurations of a particular product are demanded by other products in the solution. These mismatches must be resolved so that the service may be shared, and avoid the need to install multiple copies of the same COTS product.

Once the runtime behaviour is established, it is important to consider the other facets of product behaviour. For instance, the programming model and development tools for utilizing the COTS product in integrated solution development are a key to COTS acceptance. The COTS product needs to be easy to install, configure, and manage since it is only a part of a larger integrated solution effort. In fact, business scenarios were originally introduced in IBM SWG to reduce the effort our customers were spending on these activities with early releases of WebSphere Application Server. Installation and configuration of Application Server has vastly improved since version 3.0. Other COTS products have made improvements to the development environment to make it easier to build and test the integrated solutions described in the business scenarios.

Lesson 2 *The framework introduces a rigorous process and establishes the essential content and activities of business scenarios, which not only helps to maintain consistent quality among scenarios, but also reduces effort.*

When business scenarios were first developed, an ad hoc approach was employed in the production of business scenario artifacts. The business scenario development process emphasized activities such as scenario selection, development, validation, and testing. The process provided little guidance in the creation of artifacts or their composition. This resulted in large variations in the type of content and level of detail. This lack of consistency made it more difficult to communicate business scenarios to development organizations and also made it harder to teach new team members how to develop scenarios.

Our current approach, using the business scenario framework presented in this paper, specifies the business scenario development activities and artifact composition in more detail. Since our goal is to mimic our customers' experience with our COTS products, we adopted RUP® (Rational Unified Process®) in order to formalize the process of mimicking customer activities and allow us to produce consistent artifacts. The standard RUP vision, use case and software architecture document templates were tailored for our needs. The use of the RUP-based templates determine the composition of the business scenario artifacts. Guidance and sample text indicate the necessary level of detail. This makes the approach easier to teach and is allowing the team to develop consistent artifacts. In addition, use of the RUP vision encouraged us to focus on additional aspects of the integrated solution such as the users of the solution and the solution features that were not always considered or documented with the previous approach. Often a better understanding of these aspects of the integrated solution leads to a better understanding of the COTS users and the features needed to support the solution.

In some cases, the new approach increases the effort to produce the business scenario *solution definition* and *architecture* artifacts since it is a more comprehensive approach. However, the time to review these artifacts is reduced since the artifact composition is more clearly defined and easier to compare to and contrast with other business scenarios. It is also easier to produce the design artifacts using the consistent and more detailed information provided in the solution definition and architecture. Overall, the business scenario development effort is reduced.

Lesson 3 *It is crucial to balance the effort to produce the business scenario against the potential to discover integration issues or surface new requirements. The effort is affected by the choice of the integrated solution described by the business scenario, its breadth and the extent to which the solution is implemented.*

It is extremely important to carefully select the business scenario in order to maximize the benefits of the scenario development effort. To be beneficial to the COTS provider, the solution described in the scenario needs to apply to a significant share of COTS product customers. This is accomplished by finding common business objectives and developing general solutions to meet those objectives. The generalized solution must preserve the essential aspects of the solution requirements derived from unique business situations. Also, in order to maximize exploration of prod-

uct integration, the selected business scenario must require several COTS products that are of interest to the COTS provider. Increasing the coverage of COTS products often results in a broader solution scope and a larger scenario. This leads to other potential problems as discussed below. Interestingly, as the success of the business scenario approach is recognized, it is our experience that COTS product development teams lobby for the inclusion of their product in a scenario. Avoid "force fitting" a COTS product into an existing scenario. Instead, seek a solution that naturally includes a COTS product and is of significance to the business of COTS customers.

Large business scenarios have the advantage of wide domain coverage and are likely to uncover more integration issues and surface more new requirements. However, because of their size and complexity, there is a higher cost to create, implement, and test the resulting solution. It is also more difficult to communicate a larger business scenario to interested COTS development organizations. Product teams are reluctant to resolve integration issues or consider new requirements if these requests are not easily justified with information from the business scenario. Conversely, a smaller scenario is cheaper to create, implement and test, and it is easier for the development organization to see the role of the product under development in the solution described by the scenario. However, a less complex solution typically integrates fewer COTS products, making it less useful for discovering interoperability issues. If the business context of the solution is too narrow, critical elements may be excluded and the needed behaviour of the COTS product may not be realized. Therefore, it is important to weigh the effort to develop the scenario against the potential to discover problems when choosing the scope of a business scenario.

Ideally, the business scenario framework would provide guidance to assist with decisions about solution selection and scope. From our experience some suggestions are: consider the strategic importance of particular products to the COTS provider; identify the key features of the integrated solution through validation of the solution definition and focus on these features in the partial implementation of the solution; identify important solution development facets for the integrated solution, and minimize emulation activities for other areas.

Lesson 4 *While the use of business scenarios helps to identify and resolve integration issues, a more important benefit of business scenarios is to gain a better understanding of the integrated solutions COTS customers want. This understanding is essential to determine the COTS requirements that these solutions demand.*

Our original motivation for BSF was to respond to the integration challenges COTS customers experience during the development of integrated solutions with IBM software products. Using BSF to think and act as the customer, IBM SWG has successfully identified problems such as conflicting software dependencies, overlapping components, and difficulties installing and configuring COTS within the integrated solution.

However, we have observed that this approach tends to surface new COTS requirements not previously considered. The business scenario illustrates the use of a combination of COTS products that must work in concert to provide the capability for the integrated solution. When considered together, especially in terms of common goals for the overall solution, it may be possible to utilize the same underlying service; for example, the access and management of shared information such as user credentials may be shared.

Based on this observation, we believe that the business scenario framework provides a powerful method for surfacing COTS requirements. For this reason, we have made the elicitation of requirements our priority in our future efforts with BSF.

Lesson 5 *The use of an independent team to produce business scenarios reduces the effectiveness of analysis and evaluation of COTS, because of communication challenges with the product development organization. To overcome this problem, we propose integrating the business scenario framework into the product development process as a requirement elicitation method.*

The challenge of communicating the findings from each business scenario with the COTS development teams can lead to limited acceptance of the recommendations. It can be particularly hard to convince the development teams to accept and implement new requirements. This is largely due to the effort involved in determining the relevance of the business scenario to a particular COTS. Usually, the team has to go through the entire set of business scenario artifacts in detail in order to understand the business context for their product and thus understand the rationale for the business scenario analysis and evaluation. Recommendations that are accepted are often not adopted since they are received too late in the product development cycle.

In future, we want to explore the adoption of BSF by the COTS development teams and have business scenarios integrated into the development process. Product teams would be responsible for selecting and using scenarios oriented to their product, increasing the relevance and avoiding the problem of acceptance. Tighter integration of the processes and the elimination of an independent organization resolves the communication issues and promotes the elicitation of only relevant requirements.

Although this may solve our current problem, it raises new issues to be explored in future. How feasible and effective is it to have individual product teams independently create business scenarios? Will business scenarios continue

to cover multiple products so that the requirements that lie between products are identified? How is effort coordinated across products so as to avoid the redundancy that may occur because of overlapping scenarios developed by different products? How is the collective effort balanced against the individual efforts?

5. Conclusions and Future Work

As COTS-based system construction emerges to be a mainstream development method, new challenges are faced both by the consumers and providers of COTS products.

In this paper, we presented the Business Scenario Framework, a variation of the scenario-based approaches in requirements engineering, to address some of the challenges faced by COTS providers. We have applied evolving versions of the framework to nine scenarios during the past two and a half years, to identify integration issues in the current releases of the COTS products and surface requirements for future releases. We described in detail the artifacts and processes involved in a business scenario development and our experience in doing so.

The lessons we drew from our experience reflect both positive results and limitations. The approach enhanced the quality of COTS products in the short term by resolving integration issues. However, we need more comprehensive techniques for identifying and managing new requirements for long-term success. The approach also deepened developers' understanding of customer requirements. The defined process of the BSF allowed us to maintain consistent quality among scenarios and reduced the effort.

Two areas for improvement were identified. We need better techniques to scope the scenarios, both in the content and coverage. Second, we need to overcome communication difficulties between the business scenario team and the COTS developers. We believe this can be solved by integrating the Business Scenario Framework into the product teams' development process as a requirement elicitation method.

In future, we would like to gear the Business Scenario Framework towards the requirement elicitation of COTS products and middleware. We will focus on how results collected from the scenarios help the product teams to make architectural decisions, not only in discovering issues, but also in successfully implementing the resolutions of the issues. We also plan to define more fine-grained tasks, refine the analysis and evaluation on the scenarios to reduce the dependence on experienced designers, and explore the roles and skills needed for the elicitation work.

Acknowledgements

We thank Marin Litoiu and Mike Starkey for feedback and review; the SWG Business Scenarios and Solution Test teams at IBM for the business scenarios. This work is funded under IBM CAS Fellowship and National Sciences and Engineering Research Council of Canada.

References

[1] International Conference on COTS-Based Software Systems. `http://www.iccbss.org/`.

[2] SEI: COTS-Based Systems (CBS) Initiative. `http://www.sei.cmu.edu/cbs/`.

[3] C. Albert and L. Brownsword. "Meeting the Challenges of Commercial-Off-The-Shelf (COTS) Products: The Information Technology Solutions Evolution Process (ITSEP)". In *Proceedings of ICCBSS'02*, pages 10–20, 2002.

[4] T. G. Baker. "Lessons Learned Integrating COTS into Systems". In *Proceedings of ICCBSS'02*, pages 21–30, 2002.

[5] J. Bhuta and B. Boehm. "A Method for Compatible COTS Component Selection". In *Proceedings of ICCBSS'05*, 2005.

[6] S. Ghosh, J. Kelly, and R. Shankar. "Enabling the Selection of COTS Components". In *Proceedings of ICCBSS'05*.

[7] A. M. Hickey and A. M. Davis. "Elicitation technique selection: how do experts do it?". In *Proceedings of RE'03*, pages 169–178, 2003.

[8] K. Kuutti. "Workprocess: Scenarios As a Preliminary Vocabulary". In J. Carroll, editor, *Scenario Based Design*. 1995.

[9] M. Morisio, C. B. Seaman, A. T. Parra, V. R. Basili, S. E. Kraft, and S. E. Condon. "Investigating and improving a COTS-based software development". In *Proceedings of ICSE'00*, pages 32–41, 2000.

[10] F. Sudaman and C. Mingins. "BiCom: An Evaluation Framework for COTS Components". In *Proceedings of ICCBSS'03*, pages 207–218, 2003.

[11] A. Sutcliffe. "Scenario-based requirements engineering". In *Proceedings of RE'03*, pages 320–329, 2003.