Chapter I
Using Patterns for Engineering High-Quality E-Commerce Applications

Pankaj Kamthan
Concordia University, Canada

Hsueh-Ieng Pai
Concordia University, Canada

ABSTRACT

In this chapter, we view the development and maintenance of high-quality electronic commerce (e-commerce) applications from a Web engineering perspective. A methodology for deploying patterns as means for improving the quality of e-commerce applications is presented. To that regard, relevant quality attributes and corresponding stakeholder types for the e-commerce applications are identified. The role of development process, the challenges in making optimal use of patterns, and feasibility issues involved in doing so, are analyzed. The activities of a systematic selection and application of patterns are explored. Examples illustrating the use of patterns during macro- and micro-architecture design of business-to-consumer (B2C) e-commerce applications are given. The implications of the use of patterns in a Semantic Web context are briefly highlighted.

INTRODUCTION

Over the past decade, electronic commerce (e-commerce) (Kalakota & Whinston, 1996) has opened new vistas for many sectors of society including businesses and has revolutionized the way business is conducted. In particular, e-commerce applications have revolutionized the way business is conducted today, and reduced the gap between small-and-medium size enterprises (SMEs) and large corporations.

Although e-commerce applications have offered rich prospects, they have also brought various concerns to the providers (Kamthan, 1999).
Indeed, the successes have come with their share of failures (Nguyen, Johnson, & Hackett, 2003), many of which have been attributed to issues of quality (Pertet & Narasimhan, 2005).

A commitment to “high-quality” is both an imperative and a challenge to the providers of e-commerce applications. Indeed, empirical studies have shown (Sharkey, Scott, & Acton, 2006) that the quality of an e-commerce application is directly related to its “dimensions” of success. At the same time, it is crucial that efforts towards assuring the quality of these applications remain predictable and feasible in the long-term.

In this chapter, our interest is in a systematic approach of engineering large-scale and quality-centric e-commerce applications based on the knowledge garnered from past experience and expertise (Kamthan, 2008). For that, we rely on the notion of the problem-solving approach of patterns (Appleton, 1997; Buschmann, Henney, & Schmidt, 2007). As discussed later, the use of patterns has several advantages over other approaches in terms of quality improvement, including that the approach is preventative rather than curative (Dromey, 2003), is supported by developmental processes, and provides practical solutions along with their reasoning for problems that have been tackled in the past.

The rest of the chapter is organized as follows. We first outline the background necessary for the discussion that follows, and state our position in that regard. This is followed by the presentation of a pattern-oriented electronic commerce applications methodology (POECAM) for systematically addressing the quality of e-commerce applications. POECAM includes a model consisting of quality attributes at different tiers and the role of patterns as means for addressing them. Next, challenges and directions for future research are outlined. Finally, concluding remarks are given.

**BACKGROUND**

In this section, we present the motivation for a systematic approach to address the quality in e-commerce applications and a synopsis of patterns.

**Challenges to Development of E-Commerce Applications**

There are certain defining characteristics that make e-commerce applications unique compared to other software applications, and lead to special considerations towards their quality management.

The market may often dictate the direction of the development and the evolution of an e-commerce application. Typically, e-commerce applications have relatively short time-to-market. This additional constraint on scheduling can adversely impact the activities related to quality assurance and evaluation that the provider of an e-commerce application needs to contend with. For example, a provider may have to compromise the time allocated to verification (say, via inspections or testing). Furthermore, the information in e-commerce applications may also have to be frequently modified to reflect the state-of-the-art and/or to maintain competitive advantage.

The consumers of an e-commerce application need not be colocated; indeed, they may be in different jurisdictions in the same country or in different countries. The laws that govern the provider and the consumers of an e-commerce application may be different. For example, let A and B be two different jurisdictions and let P be a product available for purchase in jurisdiction A. Then, although browsing information on P may be legal in B, purchasing it may not be.
The consumers of an e-commerce application may reside in different locales with different time zones. The expectations of the consumers with respect to the uninterrupted servability of the application can place unique constraints on the providers, particularly towards maintenance.

There can also be stark variations in the personal preferences and abilities of consumers and their computing environments. For example, it cannot be assumed that every consumer may be familiar with the intricacies of the Web or that the consumer may be using a high-end computer with a specific user agent. The providers of e-commerce applications need to deal with such diversity.

The “faceless” nature of e-commerce can contribute to unfavorable perceptions related to trustworthiness, particularly in the time of crisis, such as (frequent) failed transactions. The providers of e-commerce applications once again need to take steps to establish credibility (Kamthan, 2007a) with the consumers.

**An Engineering View of E-Commerce Applications**

The need for managing increasing size and complexity of Web applications and the necessity of a planned development has led to the discipline of Web engineering (Ginige & Murugesan, 2001), which has been treated comprehensively in recent years (Kappel, Pröll, Reich, & Retschitzegger, 2006; Mendes & Mosley, 2006; Rossi, Pastor, Schwabe, & Olsina, 2008). We define Web engineering as a discipline concerned with the establishment and the use of sound scientific, engineering, and management principles and disciplined and systematic approaches to the successful development, deployment, and maintenance of Web applications.

For the sake of this chapter, we will consider an e-commerce application to belong to a special class of Web applications that is product of a Web engineering process. In particular, an e-commerce application can be viewed as a heterogeneous combination of views, that is, an interactive information system in a distributed (client-server) environment with a commercial intent.
The Interplay of Patterns, Development Process, and Quality of E-Commerce Applications

The reliance on past experience and expertise is critical to any development. A pattern is an entity of knowledge that provides a proven solution to a recurring problem in a given context (Appleton, 1997).

Formally, a pattern is typically described (Meszaros & Doble, 1998) using an ordered list of elements (highlighted in italics in the rest of the chapter) labeled as (pattern) name, author, context, problem, forces, solution, example, and related patterns. At times, the labels may vary across community, and other (optional) elements, such as those related to metadata, may be included to enrich the description. The name element of a pattern is often a metaphor reflecting what the solution may be about, the author element gives the identity of the pattern author(s), the context element provides the situation or preconditions within which the problem occurs, the forces element provides the constraints that are resolved to arrive at a solution, the solution element provides an abstract solution to the problem and is shown to work in practice via an example element, and the related patterns element outlines any other pattern(s) to which a pattern is related to in some way.

In the past decade, patterns have been discovered in a variety of domains of interest, including those that are applicable to the development of Web applications in general and e-commerce in particular. There are collections of patterns available in print form (Adams, Koushik, Vasudeva, & Galambos, 2001; Marks & Hong, 2006; Montero, Lozano, & González, 2002; Rossi & Koch, 2002; Rossi, Lyardet, & Schwabe, 2000; Van Duyne, Landay, & Hong, 2003; Weiss, 2003). There are collections of patterns also available in electronic form (over the Web) such as the Portland Pattern Repository, the Amsterdam Collection of Interaction Design Patterns, and the Yahoo! Design Pattern Library.

Indeed, patterns have been used for the development of certain e-commerce applications (Garzotto, Paolini, Bolchini, & Valenti, 1999; Markiewicz, Lucena, Alencar, & Cowan, 2002; Montero et al., 2002; Montero, López-Jaquero, & Molina, 2003). However, in these cases, the relation of patterns to any development process and/or to the improvement of quality is not discussed explicitly.

In general, patterns are especially suitable as means for addressing quality concerns. They go through a comprehensive review by other experts in the domain that enables them to be appropriately described and documented. They provide solutions that are specific to problems in a given context, balance competing forces, are based on established principles, are rationalized, and are proven to work in practical situations. The evaluation of usability of e-commerce applications by determining the absence/presence of design patterns has been carried out (Georgiakakis, Psaromiligkos, & Retalis, 2006; Sartzetaki, Psaromiligkos, Retalis, & Avgeriou, 2003).

There are some patterns available specifically for addressing maintainability concerns of e-commerce applications (Weiss, 2003). However, in some cases the solutions are highly technology-specific, not all the mandatory elements of a pattern are appropriately documented, and the integration of patterns into any development process is not mentioned.

There are also some patterns available for addressing usability concerns of e-commerce applications (Graham, 2003; Perzel & Kane, 1999). However, usability is viewed as an atomic (non-decomposable) concept, the patterns are strongly oriented towards user-interface design, and their integration into any user-centered development process is not shown explicitly.
A SYSTEMATIC APPROACH FOR INTEGRATING PATTERNS IN E-COMMERCE APPLICATIONS

In this section, we introduce POECAM, which is inspired by previous work on integrating patterns and Web engineering (Kamthan, 2008). POECAM consists of the following interrelated and nonlinear sequence of steps:

1. Setting Goals
2. Identifying Stakeholders
3. Selecting the Development Process Model
4. Identifying and Organizing Quality Concerns
5. Acquiring and Selecting Suitable Patterns
6. Applying Patterns

From a practical standpoint of integrating patterns in e-commerce applications, each of the steps must be feasible. If it is not, we revert back to it and move forward once the necessary modifications are made. The feasibility study could be a part of the overall e-commerce application project management planning activity. Figure 1 gives an overview of the steps in POECAM.

We now discuss the work involved in each of these six steps in detail.

Step One: Setting Goals

It is known that a project without clear goals will not achieve its goals clearly (Gilb, 1988). A POECAM approach must be based upon high-level organizational (business), social, and/or technical goals that need to be accomplished.

For instance, a technical goal for integrating patterns in the overall development strategy could be to rely on past experience and expertise in the domain to produce high-quality e-commerce ap-
Using Patterns for Engineering High-Quality E-Commerce Applications

Applications from stakeholders’ perspective. For a goal to be realized in practice, however, it must also be feasible.

**Step Two: Identifying Stakeholders**

There are systematic approaches for identification and refinement of stakeholder classes (Sharp, Galal, & Finkelstein, 1999). We identify two broad classes of stakeholders with respect to their roles in relationship to an e-commerce application: a *producer* (provider, manager, engineer, or maintainer) is the one who develops, deploys, or maintains the e-commerce application, and a *consumer* (novice or expert user) is the one who uses the e-commerce application for some purpose.

We emphasize that the above classification is based on the role of a stakeholder. For example, both the engineer and the user could be the same person, but their roles with respect to the interaction with an e-commerce application are different.

The viewpoint-oriented requirements definition (VORD) (Kotonya & Sommerville, 1998) also gives us a way of classifying aforementioned stakeholders of interactive systems in a client-server environment such as e-commerce applications. From a VORD perspective, the provider and the manager are indirect viewpoints, while the engineer and the maintainer are direct viewpoints in their relationship to the e-commerce application.

**Step Three: Selecting the Development Process Model**

The inclusion of patterns in the development of e-commerce applications cannot be ad-hoc or an afterthought. We recommend POECAM within the framework of an existing process environment that already has broad community and tool support in order to keep the cost to a minimum and the learning curve low.

In light of the unique characteristics of e-commerce applications, it is crucial that the development process be flexible and user-centered. A flexible user-centric process for developing e-commerce applications will typically be nonlinear (iterative and incremental) and, to that regard, we recommend the adoption of one of the following process models. Extreme programming (XP) (Beck & Andres, 2005) is a mature and broadly-used agile methodology for software development. XP places “lightweight” requirements on resources and is suitable for SMEs. On the other hand, the unified process (UP) (Jacobson, Booch, & Rumbaugh, 1999) is an archetype of model-based and uses a case-driven process framework, of which the rational unified process (RUP) (Kruchten, 2004) is an instance. RUP requires heavy business and technical modeling and documentation, and is especially suited for large businesses.

Both XP (Kappel et al., 2006; Wallace, Raggett, & Aufgang, 2002) and RUP (Kappel et al., 2006) have been “customized” for Web applications and by reference to e-commerce applications. There is provision of the use of patterns during the design phase in both XP and RUP.

**Step Four: Identifying and Organizing Quality Concerns**

For the purpose of this chapter, we focus on the semiotic quality of e-commerce applications. From a semiotics (Shanks, 1999; Stamper, 1992) viewpoint, we can view an e-commerce application as an information system on six interrelated levels: physical, empirical, syntactic, semantic, pragmatic, and social. In this chapter, we shall restrict ourselves to the discussion of the pragmatic level, which is responsible for the relation of signs to their interpreters.

Next, we contend that pragmatic quality is a multidimensional concept, and decompose it into granular levels that consist of known attributes that can be addressed directly or indirectly. For the definitions of these quality attributes, we
Finally, we assign patterns as means for improving the quality attributes. Table 1 summarizes this construction.

We contend that the quality attributes in Table 1 are necessary, but make no claim of their sufficiency. We also note that the pragmatic quality attributes (discussed later) in Table 1 cannot (at least mathematically), with respect to stakeholders, be completely satisfied. For example, an a priori guarantee that an e-commerce application will be usable to all users at all times in all computing environments that the users deploy, is not realistic.

The quality attributes in Table 1 are not mutually exclusive. Indeed, the quality attributes in Tier 3 depend on that in Tier 2, which in turn depend on Tier 1. For example, if a user cannot read, the user cannot comprehend the information in an e-commerce application, and thereby cannot use it to its full potential. Similarly, for an e-commerce application to be reliable, it must be available.

Furthermore, the quality attributes within the same tier in Table 1 are not necessarily mutually exclusive. For example, the steps taken towards improving reliability (say, fault tolerance) may lead to redundant source code or data (that can be unfavorable to maintainability) but enable ease-of-use (that can be favorable to usability).

The Quality-Stakeholder Contract

For the sake of this chapter, we view pragmatic quality as a contract between an e-commerce application and a stakeholder. For simplicity, we will limit ourselves to the discussion of (not necessarily mutually exclusive) stakeholders of the type end-user and engineer.

The relevance of quality attributes in Table 1 varies with respect to stakeholder types:

- **Pragmatic-Tier 1.** The quality attributes of direct concern to an end-user are aesthetics, availability, familiarity, and readability. The quality attribute of direct concern to an engineer is space and time efficiency.

---

**Table 1. A model for the pragmatic quality of e-commerce applications**

<table>
<thead>
<tr>
<th>Semiotic Level</th>
<th>Quality Attributes</th>
<th>Means for Quality Assurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Quality Concerns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pragmatic</td>
<td>[Tier 3] Maintainability, Usability</td>
<td>Patterns</td>
</tr>
<tr>
<td></td>
<td>[Tier 2] Comprehensibility, Performance, Reliability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Tier 1] Aesthetics, Availability, Efficiency, Familiarity, Readability</td>
<td></td>
</tr>
<tr>
<td>Physical, Empirical, Syntactic, and Semantic Quality Concerns</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using Patterns for Engineering High-Quality E-Commerce Applications

- **Pragmatic-Tier 2.** The quality attributes of direct concern to an end-user are comprehensibility, performance, and reliability. The quality attribute of direct concern to an engineer is comprehensibility.

- **Pragmatic-Tier 3.** The quality attribute of direct concern to an end-user is usability. We will view accessibility as a special case of usability (Mendes & Mosley, 2006). The quality attribute of direct concern to an engineer is maintainability. We will consider modifiability, portability, and reusability as special cases of maintainability (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996).

Finally, we note that the significance of quality attributes will vary across business models (Bambury, 1998) as well as different types of e-commerce applications. For example, the quality needs of a music shopping portal will vary from that of a fine art auction application.

**Step Five: Acquiring and Selecting Suitable Patterns**

The challenges in the patterns’ acquisition and selection process stem from a variety of factors:

- **Availability of Patterns.** For an adoption of a pattern-based approach to the development of e-commerce applications, it is important to have design and implementation patterns that can sufficiently “map” the solution space. However, there is no a priori guarantee that for every quality-related problem, there will be suitable pattern(s) available for it.

- **Findability of Patterns.** There are currently several challenges that inhibit our ability to detect and find desirable patterns (Manolescu, Kozaczynski, Miller, & Hogg, 2007). We consider three of these challenges:
  - **Classification of Patterns.** There is no currently unique classification of patterns. There are some patterns that may be classified into familiar categories (like “structural” or “behavioral”) while others may be presented as a loosely related informal collection. Apart from some isolated efforts (Van Welie & Van Der Veer, 2000; Schumacher et al., 2006), patterns are currently not organized by quality attributes and in only very few cases organized as directly relevant to e-commerce. This could adversely impact the task of locating desirable patterns (Segerstål & Jokela, 2006).
  - **Relationships among Patterns.** Often, patterns do not exist in isolation and are part of an overall vocabulary (e.g., a “pattern system” or a “pattern language”) that attempts to solve a larger problem than possible with an individual pattern. The relationships among patterns can be implicit or explicit. Indeed, due to the context-driven relationships among them, the use of one pattern can lead to the commitment of using other pattern(s). For example, selection and application of a pattern for a shopping system would place us in a context that would require the selection of a pattern for payment, which in turn would require the selection of a pattern specific to authentication.
  - **Similarity among Patterns.** The name of a pattern plays a crucial role in acquiring and selecting desirable candidates. There are some patterns that have similar or same names but have semantically different functionality. There are also other patterns in different pattern collections that have similar intent or functionality but have
different names. For example, the NEWS pattern (Rossi et al., 2000) is similar to the WHAT’S NEW PAGE pattern (Van Duyne et al., 2003). These patterns may have been (re)discovered independently.

- **Cost of Deploying Patterns.** There is evident cost associated with the deployment of patterns that cannot be disregarded. It is known that patterns are means of conceptual reuse, and there is a cost in terms of time, effort, and resources of learning and adaptation involved in any reuse. The issue of cost is particularly acute in educational contexts where affordability of commercial resources (such as books on patterns) is a constant concern.

### Step Six: Applying Patterns

There are three main nonmutually exclusive concerns in the application of patterns: the understanding of the pattern description, the order in which patterns are applied, and the result upon the composition of patterns.

1. **Understanding the Pattern.** The understanding of the underlying problem, the context in which it occurs, and the trade-offs and consequences of the proposed solution are imperative. For example, patterns that suggest solutions involving the use of different colors will not be applicable in contexts where the underlying monitor does not support it (such as when the screen is monochrome) or if the user is color blind.

2. **Order of Application of Patterns.** In the design phase, the patterns for a high-level (macro-architecture) design are applied first, followed by the patterns for a low-level (micro-architecture) design. In each of these levels, the order of the application of patterns does matter and, if ignored, it may not be at all possible to apply other patterns and/or lead to a nonoptimal result.

3. **Composition of Patterns.** The relationships among patterns can be favorable, unfavorable, or neutral. Even when a certain order in the application of patterns is followed, it is the result upon the composition of patterns that should appear suitable. Although the application of individual patterns may appear acceptable, the result with a multiplicity of patterns may not.

For the remainder of this section, we will assume that, among the given possibilities (Adams et al., 2001), the e-commerce application under development is of the type business-to-consumer (B2C) and being accessed via stationary devices (e.g., desktop computers) or pseudo-stationary devices (e.g., notebook computers). Furthermore, even though patterns are applicable to all the phases of a development process, including requirements, design, and implementation for considerations of space, we will limit ourselves in this chapter to addressing the role of patterns in the design phase of e-commerce applications.

As evident from the discussion that follows, the patterns presented here form a skeleton sequence that traverses through several existing collections of patterns. Our selection of patterns is based on their generality, neutrality with respect to any specific application domain, broad availability, parity to the quality attribute at hand, suitability of the context and the forces (where available), and the reputation of the authors.

In order to distinguish the patterns from the main text, their names are listed in uppercase. The Appendix summarizes the quality attributes and patterns mentioned in this section.
Macro-Architecture Design of E-Commerce Applications

The macro-architecture design is the place where high-level design decisions, independent of any implementation paradigm or technology, are made.

Patterns Applied from a Business Viewpoint

An e-commerce application will implicitly or explicitly target some domain. The choice of the domain name (such as .net or .com) does not always or automatically reveal the nature of the domain. There are patterns available for certain common genres, like ENABLING INTRANETS (for businesses in general), PERSONAL E-COMMERCE (for small businesses), VALUABLE COMPANY SITES (for large organizations), and STIMULATING ARTS AND ENTERTAINMENT (for news organizations) (Van Duyne et al., 2003). The use of such genre-specific patterns can increase user familiarity with the e-commerce application.

Furthermore, the organization owning an e-commerce application may wish to serve (potential) consumers in diverse cultural and/or geopolitical situations (i.e., in different countries and using different natural languages). This could be done using the LOCALE HANDLING pattern (Busse, 2002). In retrospect, it increases maintenance responsibilities.

Patterns Applied from a Technical Viewpoint

The macro-architecture patterns that we suggest are based on the fact that e-commerce applications are a class of distributed request-response-type interactive systems. Specifically, the applicable patterns are the CLIENT-SERVER pattern (Schmidt, Stal, Rohnert, & Buschmann, 2000), followed by the APPLICATION SERVER pattern (Manolescu & Kunzle, 2001), which in turn is followed by the MODEL-VIEW-CONTROLLER (MVC) pattern (Buschmann et al., 1996).

The CLIENT-SERVER pattern supports maintainability. For example, a server or resources on

Figure 2. A view of the macro-architecture design patterns in the development of e-commerce applications
the server-side could be modified without impacting the client. Also, a single server can support multiple clients simultaneously, or a client could make simultaneous requests for resources residing on multiple servers. The APPLICATION SERVER pattern also supports maintainability: it isolates the e-commerce application from other aspects on the server-side such that the communication between the application itself and the Web server takes place via the SINGLE POINT OF ACCESS (Yoder & Barcalow, 1997) pattern. This separation allows the e-commerce application to evolve independently.

In applying the MVC pattern, the principle of separation of structure from presentation of content in a markup document leads to a separation of semantically-different aspects into three components, namely model, view, and controller. This minimizes the coupling between these components. Therefore, the same model in a MVC could be used with multiple views and multiple controllers. For example, the same information could be transformed and delivered to different browser environments or user needs. This improves the maintainability of an e-commerce application.

Figure 2 presents an abstract view of the aforementioned macro-architecture design patterns.

There are several implementations of MVC available in a variety of programming languages such as Java and hypertext preprocessor (PHP), and application frameworks like asynchronous JavaScript and XML (AJAX) and Rails.

The CUSTOMIZED PRESENTATION TO HOST pattern (Adams et al., 2001) mimics some aspects of MVC. It aims to improve usability by providing a user-friendly interface. However, there is a tight coupling between the customized presentation view and the host application. Hence, any changes to the host application may require changes to the presentation view. This is unfavorable to maintainability.

**Reliability Design**

For addressing reliability (specifically, availability) concerns, the macro-architecture design of server-side components of an e-commerce application could use a number of patterns (Ahluwalia & Jain, 2006; Manolescu & Kunzle, 2001). For example, the extra measures to support the availability of an e-commerce application (unrelated to the functionality of the application) could be included by using the INTRODUCE REDUNDANCY pattern.

One way to introduce redundancy is to have a cluster of multiple servers such as suggested by the FAIL-OVER THROUGH CLUSTERING pattern, where if one (primary) server fails, the other (secondary) server takes over the responsibility. In retrospect, redundancy also increases maintenance responsibilities.

If and when the need arises, a failure message could be relayed directly using the FAILURE NOTIFICATION pattern or indirectly using the HEARTBEAT pattern, where an engineer is informed via periodic broadcasts that a specific Web server is available; the absence of such a message would then imply its unavailability.

**Micro-Architecture Design of E-Commerce Applications**

The micro-architecture design is the place where low-level design decisions are cast.

For the rest of the section, we will focus only on the design aspects that impact pragmatic quality. As such, in the following, our attention is geared more towards client-side rather than server-side concerns.

Interaction design (Preece, Rogers, & Sharp, 2002) is perhaps the most crucial client-side concern among e-commerce applications. The goal of interaction design is to make both the content and the user interface useful, easy-to-use, and enjoy-
able. Many of the patterns available for interaction design for interactive systems in general are also applicable to e-commerce applications.

We now consider four of the most critical interaction design aspects of e-commerce applications, namely information design, navigation design, search design, and presentation design independent of any specific domain. We note that these aspects of design are not mutually exclusive.

**Information Design**

The delivery of information could either be static or dynamic. The dynamic delivery can take place via conversion of a *single source* of information in, for example, a database, to a format suitable for the Web, which enables adaptive maintenance. This can be accomplished by using the GENERIC CONTENT FORMAT pattern (Zdun & Vogel, 2002) for representing the information inside the “model” in MVC.

It is evident that not all users are the same; the properties of an e-commerce application like structure, content, and links could be “personalized” for each user. This is possible via patterns for personalization (Rossi, Schwabe, Danculovic, & Miaton, 2001), namely STRUCTURE PERSONALIZATION, CONTENT PERSONALIZATION, and LINK PERSONALIZATION. This will improve the user’s experience with (and thus the usability of) the application. However, an optimal personalization would usually require the user to release some degree of personal information (say, via registration on a Web Portal), which in turn would lead to loss of privacy.

The information presented on a single “Web page” is often aggregated from several sources. For example, the entry point (from user-perspective) to an e-commerce application can be realized using the HOME PAGE pattern (Graham, 2003). Then, the “home page” of a news organization served from the main source may include a latest news ticker from one server, weather information from another server, the stock market information from a financial Web service, and periodically changing advertisements from yet another source.

The delivered information needs to be organized. This can be systematically realized by the use of the WHOLE-PART pattern (Buschmann et al., 1996), which enables a hierarchical organization of objects. Since each of these objects can be modified or replaced independently, the WHOLE-PART pattern supports maintainability. Also, since a “part” can correspond to more than one “whole,” the WHOLE-PART pattern also supports reusability. However, multiple indirections stemming from client requests and responses for fulfilling them can lead to a loss of performance, particularly when each “part” itself is structured as WHOLE-PART.

Next, we look at the classification of information, which is a conventional approach by humans for understanding information. The information organization patterns (Van Duyne et al., 2003), when used appropriately, aid readability, comprehensibility, and usability. For example, the GRID LAYOUT pattern suggests the organization of information in a single document into a grid of rows and columns, where each atomic information element is made to fit within this grid. For special cases, such as a shopping system, the information of products could be further organized using the CATALOG pattern (Fernandez, Liu, & Pan, 2001). The WHAT’S NEW PAGE pattern provides newly added information and could include the CHRONOLOGICAL ORGANIZATION pattern.

The users of an e-commerce application can vary in their capabilities and preferences, and may find one view of information to be more usable than another. The MIRRORWORLD pattern (Germán & Cowan, 2000) provides two or more views of the same information. Specifically, information in these views could be presented (Tidwell, 2006) in TWO-PANEL SELECTOR pattern when we have two different views that are to be presented simultaneously, or CLOSABLE PANELS or CARD STACK patterns when we have several
different views to be presented in such a way that only one view is visible at a time in each panel or stack, respectively.

Now, documents in an e-commerce application may contain images for presenting some information such as the corporate logo or product pictures. The FAST-DOWNLOADING IMAGES pattern (Van Duyne et al., 2003) suggests the creation of images optimized for color and size in an appropriate format, and thus aids accessibility and performance. The REUSABLE IMAGES pattern (Van Duyne et al., 2003) suggests caching images that appear at multiple places in an e-commerce application, and thereby aids performance.

To improve usability, there should be a provision in the information design to support an internal locus of control (thereby provide options to a user) and for users to recover (say, from inadvertent errors). The MULTI-LEVEL UNDO pattern (Tidwell, 2006) provides a way to easily reverse a series of actions performed by the user in an e-commerce application that can track user session and maintain state.

**Navigation Design**

Navigation is traversal in information space (Lynch & Horton, 2002) for some purpose, such as casual or targeted browsing for information or complementing a reading sequence (like in electronic books). Both the intra- and the inter-document navigation within the context of an e-commerce application are realized by the use of hypermedia (Germán & Cowan, 2000).

There are various patterns for navigating through an e-commerce application that have been proposed over the years (Lyardet & Rossi, 1998; Marks & Hong, 2006; Tidwell, 2006; Van Duyne et al., 2003). The navigation patterns, when use appropriately, aid usability. For example, the BREADCRUMBS pattern (Marks & Hong, 2006; Van Duyne et al., 2003) could be used to inform the user of the user’s location. The Yahoo! Directory was one of the earliest users of the BREADCRUMBS pattern. The CLEAR ENTRY POINTS pattern (Tidwell, 2006) presents only a few entry points into the interface, which can restrict the navigation to a specific category and make it task-oriented. For special cases, such as a shopping system, the products could be navigated through well-defined steps using the SHOPPING PROCESS pattern (Fernandez et al., 2001).

The FLY-OUT MENU pattern (Marks & Hong, 2006) could be used to present content organized in a “compound” menu where each menu item itself has a submenu that expands only upon interaction and when the user desires. This enables a large amount of navigation information to be “hidden” from the user and presented only “on-demand,” thereby improving both (spatial) efficiency and readability. The FLY-OUT MENU pattern could itself be arranged horizontally or vertically as suggested by the HORIZONTAL NAVIGATION or VERTICAL NAVIGATION patterns (Marks & Hong, 2006), respectively.

Any navigation design must take exceptional behavior into consideration to support usability. The SESSION pattern (Weiss, 2003) can help maintain the state of the e-commerce application in the event of an interruption of navigation flow. The MISSING LINK pattern (German & Cowan, 2000) informs the user that certain hyperlink does not exist and suggests alternatives.

There are navigation design patterns that enable efficient use of space and aid comprehensibility (Tidwell, 2006). For example, the WIZARD pattern leads the user through the interface step by step for carrying out tasks in a prescribed order. It can also be used to implement a context-sensitive help on a given functionality. The RESPONSIVE DISCLOSURE pattern starts with a very minimal interface, and guides a user through a series of steps by showing more of the interface as the user completes each step. These two patterns could, for example, be used for carrying out a registration process. Now, during such a process, the user may have to be presented with several options (e.g., multiple mailing addresses or credit cards).
However, the designer is faced with the problem that the information on all of them would not fit on a single panel and a user does not have to see the details of all the options simultaneously. In such a case, the CARD STACK pattern could be used where one option is visible by default and the others are hidden.

**Search Design**

The goal of searching is finding information. Even though searching is not native to e-commerce applications, it has become ever more challenging as the amount of information to be searched through increases.

Various patterns for searching have been proposed over the years (Lyardet, Rossi, & Schwabe, 1999; Marks & Hong, 2006; Van Duyne et al., 2003). When used appropriately, searching patterns aid comprehensibility and performance.

The use of a STRAIGHTFORWARD SEARCH FORMS pattern (Van Duyne et al., 2003) with a SIMPLE SEARCH INTERFACE pattern (Lyardet et al., 1999), that requires minimal technical background on part of the user, will contribute towards comprehensibility. The use of a SELECTABLE SEARCH SPACE pattern (Lyardet et al., 1999) that can restrict the search to a specific category, the SELECTABLE KEYWORDS pattern (Lyardet et al., 1999) that can suggest keywords for improving subsequent search results based on the past experience, and the ORGANIZED SEARCH RESULTS pattern (Van Duyne et al., 2003) that presents a summary of the most relevant search results, can all improve the effectiveness of the searching activity. The Google search engine implements some of these search patterns.

When search results are too numerous such that they could cognitively overload a user, they could be split into multiple sections that could be navigated sequentially using the PAGING pattern (Marks & Hong, 2006). Finally, we note that the ACCOUNT SETUP pattern (Marks & Hong, 2006) allows reuse of a user’s personal information and can assist in realizing the efficacy of the SELECTABLE KEYWORDS pattern, especially on Web portals.

Any search design must also take exceptional behavior into consideration to support usability. For example, long keywords may be difficult for a user to remember and are prone to typographical

**Figure 3. An assembly of some interaction design patterns for an abstract e-commerce application**
Using Patterns for Engineering High-Quality E-Commerce Applications

errors. The use of the AUTO COMPLETE pattern (Yahoo! Design Pattern Library) can circumvent this problem.

Presentation Design

It has been shown in surveys (Fogg, Soohoo, Danielson, Marable, Stanford, & Tauber, 2002) that users highly value the aesthetics or the “look and feel” of an e-commerce application. An e-commerce application that is appropriately presented also gives the perception of “professionalism” that is essential for establishing credibility. The elements of presentation apply to all aspects of design discussed previously.

For the purpose of aesthetics, a variety of patterns could be used. For example, using the DEEP BACKGROUND pattern (Tidwell, 2006), an image or gradient could be placed in the background of a document in such a way that it visually recedes behind the foreground elements.

In traditional “brick and mortar” shops, certain products that are seasonal, for special occasions, on sale or otherwise recommended, are often displayed in a special manner. This selected list of products could be highlighted via the FEATURED PRODUCTS pattern (Van Duyne et al., 2003).

It is known that colors can have a positive impact both cognitively and aesthetically if used appropriately, in particular taking into the considerations of the color blind (Rigden, 1999). Often, organizations have their favorite color(s) that are prominently reflected in their logo, and placed on their business cards and office supplies. Using patterns like FEW HUES, MANY VALUES, or COLOR-CODED SECTIONS (Tidwell, 2006), an e-commerce application could be given a unique “identity.” It is known that the user interface must reflect the state of the software. So, for example, the option in a navigation bar selected by a user could be reflected uniquely upon visitation by the user. There are of course other presentation issues that are crucial, such as choice of fonts and their properties, layout and positioning of user interface components, use of white space, and so on. These could also be addressed via an appropriate use of patterns.

Example

Figure 3 gives an example of a sequence of structural (specifically, visual) patterns for interaction design in an abstract commercial setting. This could serve as a typical manifestation of the “what’s new page” or a “products page” of an organization.

This sequence of patterns in the example can be enriched by extension in two different directions: (1) if necessary, further patterns can be added to the sequence, and (2) the example itself could be a part of the overall information architecture of an e-commerce application, the design of which is also based on a pattern.

POECAM in Perspective

In this section, we briefly outline the scope and limitations of POECAM for e-commerce applications.

First of all, for POECAM to be applicable, the organization should be at a level of maturity so that it is able to adopt a systematic process for developing the e-commerce application. However, achieving such a level has an associated cost that is neither trivial nor automatic (Paulk, Weber, Curtis, & Chrissis, 1995). Furthermore, engineers must be trained to make appropriate use of patterns, which often comes with experience. For example, not being able to understand the context in which a problem occurs can lead to a misapplication of the solution of the pattern, and hence a nonoptimal or even undesirable result. Finally, although it is not an inherent limitation of POECAM, we note that the list of patterns presented in this chapter is by no means complete and is subject to evolution.
FUTURE TRENDS

The work presented in this chapter can be extended in a few different directions, which we discuss briefly.

Extensions of POECAM

One possible extension of the model presented in Table 1 is increasing the granularity of the quality attributes at the level Pragmatic-Tier 1, and thereby adding another level (say, Tier 0) underneath it. In that case, for example, fault tolerance and recoverability could be candidate quality attributes that belong to the level Pragmatic-Tier 0. Some patterns for fault tolerance and recoverability have been suggested (Ahluwalia & Jain, 2006). Other possible extensions of the model are the use of patterns for improving higher- or lower-level semiotic quality concerns.

Use of Patterns in E-Commerce Applications for the Social Web

In recent years, the term social Web, or more commonly known by the pseudonym Web 2.0 (O’Reilly, 2005), has been put forth to describe the apparent “humanization” and “socialization” of the Web as it moves towards becoming a means of participation and collaboration driven by “collective intelligence.” An elaboration of the social level of Table 1 could help towards accommodating e-commerce applications targeted for Web 2.0.

For example, one extension of interest would be addressing the social quality concerns, namely credibility, legality, privacy, and security. (We note that not all issues pertaining to these attributes, such as those related to the underlying platform or operating system, are within the scope of the development of e-commerce applications.) These can be organized in multiple different tiers, with credibility placed at a higher tier than legality, privacy, and security.

There are patterns that partly address credibility and legal concerns, for example, those for writing policy statements (Perzel & Kane, 1999) such as information, reputation, and warranty policies (Kaluscha & Grabner-Kräuter, 2003), and for licensing (Perry & Kaminski, 2005), respectively. For instance, including a pattern for a policy statement on advertising would contribute to the overall credibility of the e-commerce application in which the example from the previous section resides. Also, patterns for privacy (Hafiz, 2006; Romanosky, Acquisti, Hong, Cranor, & Friedman, 2006; Van Duyne et al., 2003; Weiss, 2003) and security (Adams et al., 2001; Marks & Hong, 2006; Schumacher, Fernandez-Buglioni, Hybertson, Buschmann, & Sommerlad, 2006; Yoder & Barcalow, 1997; Van Duyne et al., 2003; Weiss, 2003) have been proposed in recent years, many of which apply to the development of e-commerce applications.

The aforementioned extensions, however, would require that the aspects of micro-architecture design to which the patterns are applied are essentially different than those dealt with in this chapter. For example, in case of applying patterns for privacy and security, the attention would be more on the server-side rather than on the client-side components. Furthermore, since steps taken towards the aforementioned social quality concerns are not always favorable to maintainability and/or usability or some of the quality attributes they depend upon, care would need to be taken in selecting and applying these patterns in the development of e-commerce applications.

Use of Patterns in E-Commerce Applications for the Semantic Web

The Semantic Web has recently emerged as an extension of the current Web that adds technological infrastructure for better knowledge representation, interpretation, and reasoning (Hendler, Lassila, & Berners-Lee, 2001). It thus
makes information far more amenable to machine-consumption than that is possible within the current Web. An elaboration of the semantic level of Table 1 could help towards accommodating applications for the Semantic Web.

An ontology is an explicit formal specification of a conceptualization that consists of a set of concepts in a domain and relations among them (Gruber, 1993). Ontologies provide precise means of representing knowledge and, arguably, form one of the most important layers in the Semantic Web infrastructure. By enabling better opportunities for organization and inferencing from given information (via navigation or searching), ontologies can play a crucial role in e-commerce applications of the future (Kamthan & Pai, 2006a, 2006b).

The ontology for Web application patterns (OWAP) (Kamthan & Pai, 2006a) is an ontology of typical patterns, many of which occur in e-commerce applications. With the help of a reasoner, it is possible to devise complex query formulations to make implicit knowledge in OWAP explicit. Thus, we can obtain more “intelligent” answers than that possible in a setting of, say, a database-backed e-commerce application where the search results are typically based on a simple keyword-based fetch and return.

As ontologies increase in size and complexity their methodical development becomes a necessity, and for which several approaches have been suggested (Gómez-Pérez, Fernández-López, & Corcho, 2004). The efforts towards a systematic approach towards addressing the quality of ontologies for the Semantic Web are at present scarce (Burton-Jones, Storey, Sugumaran, & Ahluwalia, 2005) and patterns for developing ontologies are currently in their infancy (Gangemi, 2005).

Use of Patterns in M-Commerce Applications

As mobile devices proliferate, they are increasingly being used for B2C commerce. The need for a systematic approach towards the engineering of mobile commerce (m-commerce) applications (Kamthan, 2007b; Mahmoud & Maamar, 2006) then arises.

The constrained environment of mobile devices (as compared to desktop computers) poses unique challenges pertaining to quality of applications available on them, and is of concern to both the producers and the consumers. For example, the restricted user interface and limited processor speed require that the producers take special measures to ensure usability and performance of m-commerce applications. Thus, the interplay between the stakeholder quality concerns (Chan & Fang, 2001; Kamthan, 2007c) and patterns (Ihme & Abrahamsson, 2005; Mazhelis, Markkula, & Jakobsson, 2005) for m-commerce applications, similar to POECAM, would be of interest.

POECAM in E-Commerce Education

There are various challenges being faced in e-commerce education (Chan, 2001), including the orientation of courses. The curriculum at educational institutions related to e-commerce applications has traditionally been dominated by courses that focus on technological (Yan & Fang, 2005) rather than theoretical aspects of development. In recent years, there has been a movement towards embracing the “engineering” aspects of e-commerce applications (Treese & Stewart, 2003), although a commitment to quality in general and patterns in particular is currently lacking.

POECAM presents an opportunity for integration of patterns in e-commerce-related course projects in such a way that theory and practice meet, thrive, and harmonically coexist. However, such integration cannot be ad hoc; it needs to be aligned with the e-commerce body of knowledge adopted and the teaching strategies and learning theories being followed. In this regard, an examination of the positioning of POECAM among the options for e-commerce education systems (Toraskar & Lee, 2006) would be of interest.
CONCLUSION

A lasting commitment towards quality has both ethical (Tavani, 2004) and business implications. For longevity and for acceptance by stakeholders, a disciplined but sufficiently agile approach towards the development of large-scale e-commerce applications is necessary. Integral to this is a systematic and lasting view towards quality, and means for addressing it.

As reflected by POECAM, patterns provide one practical means for addressing the quality of e-commerce applications, if they are found, adopted, and applied with feasibility issues in consideration. Also, for patterns to continue being useful as the source of guidance and knowledge, they must be adequately described and documented, freely available and readily findable, and evolve with the needs of the e-commerce domain.

In conclusion, we believe that an initial investment in a quality-centric approach to e-commerce applications using patterns is in interest of all stakeholders where the benefits can outweigh the costs in the long-term. For that to be realized, a reflection on and a shift in the current organizational culture (Wiegers, 1996) may be necessary. Furthermore, for a strategic change in an organization to be effective, it needs to be predictable and in turn systematic. Therefore, the integration of patterns in general and the adoption of POECAM in particular cannot be ad-hoc/spontaneous, and immune/isolated from its organizational context (Pettigrew, 1985). It has to be a part of the overall organizational vision/willingness/capability to change, and needs to be planned in advance and executed on demand.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their feedback and suggestions for improvement.

REFERENCES


Marks, M., & Hong, D. (2006). *Web design patterns collection technical design* (Center for Docu-
Using Patterns for Engineering High-Quality E-Commerce Applications


APPENDIX

Table 2 provides a mapping between quality attributes and corresponding patterns. A (+) symbol post-fixed to a pattern name reflects a positive impact on the corresponding quality attribute, whereas a (−) symbol reflects a negative impact. As mentioned earlier, the list of patterns is by no means complete and is subject to evolution. The rating scheme can also evolve to become more granular.

Table 2. Pragmatic quality attributes of an e-commerce application and corresponding patterns along with their impact ratings.

<table>
<thead>
<tr>
<th>Pragmatic Quality Attribute</th>
<th>Pattern(s)</th>
</tr>
</thead>
</table>
| Aesthetics                  | COLOR-CODED SECTIONS (+)  
DEEP BACKGROUND (+)  
FEATURED PRODUCTS (+)  
FEW HUES, MANY VALUES (+) |
| Availability                | FAIL-OVER THROUGH CLUSTERING (+)  
FAILURE NOTIFICATION (+)  
HEARTBEAT (+)  
INTRODUCE REDUNDANCY (+) |
| Comprehensibility           | CATALOG (+)  
CHRONOLOGICAL ORGANIZATION (+)  
GRID LAYOUT (+)  
PAGING (+)  
RESPONSIVE DISCLOSURE (+)  
SIMPLE SEARCH INTERFACE (+)  
STRAIGHTFORWARD SEARCH FORMS (+)  
WIZARD (+) |
| Efficiency                  | CARD STACK (+)  
FLY-OUT MENU (+)  
ORGANIZED SEARCH RESULTS (+) |
| Familiarity                 | ENABLING INTRANETS (+)  
PERSONAL E-COMMERCE (+)  
VALUABLE COMPANY SITES (+)  
STIMULATING ARTS AND ENTERTAINMENT (+) |
| Maintainability             | APPLICATION SERVER (+)  
CLIENT-SERVER (+)  
CUSTOMIZED PRESENTATION TO HOST (−)  
FAIL-OVER THROUGH CLUSTERING (−)  
INTRODUCE REDUNDANCY (−)  
LOCALE HANDLING (−)  
MODEL-VIEW-CONTROLLER (+)  
WHOLE-PART (+) |
| Performance                 | ACCOUNT SETUP (+)  
FAST-DOWNLOADING IMAGES (+)  
REUSABLE IMAGES (+)  
SELECTABLE KEYWORDS (+)  
SELECTABLE SEARCH SPACE (+)  
WHOLE-PART (−) |

Continued on following page
### Table 2. continued

<table>
<thead>
<tr>
<th>Readability</th>
<th>FLY-OUT MENU (+)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GRID LAYOUT (+)</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL NAVIGATION (+)</td>
</tr>
<tr>
<td></td>
<td>VERTICAL NAVIGATION (+)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reliability</th>
<th>FAILURE NOTIFICATION (+)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>INTRODUCE REDUNDANCY (+)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usability</th>
<th>AUTO COMPLETE (+)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BREADCRUMBS (+)</td>
</tr>
<tr>
<td></td>
<td>CARD STACK (+)</td>
</tr>
<tr>
<td></td>
<td>CLEAR ENTRY POINTS (+)</td>
</tr>
<tr>
<td></td>
<td>CLOSABLE PANELS (+)</td>
</tr>
<tr>
<td></td>
<td>CONTENT PERSONALIZATION (+)</td>
</tr>
<tr>
<td></td>
<td>CUSTOMIZED PRESENTATION TO HOST (+)</td>
</tr>
<tr>
<td></td>
<td>FAST-DOWNLOADING IMAGES (+)</td>
</tr>
<tr>
<td></td>
<td>HOME PAGE (+)</td>
</tr>
<tr>
<td></td>
<td>LINK PERSONALIZATION (+)</td>
</tr>
<tr>
<td></td>
<td>MISSING LINK (+)</td>
</tr>
<tr>
<td></td>
<td>MIRRORWORLD (+)</td>
</tr>
<tr>
<td></td>
<td>MULTI-LEVEL UNDO (+)</td>
</tr>
<tr>
<td></td>
<td>NEWS (+)</td>
</tr>
<tr>
<td></td>
<td>SESSION (+)</td>
</tr>
<tr>
<td></td>
<td>SHOPPING PROCESS (+)</td>
</tr>
<tr>
<td></td>
<td>STRUCTURE PERSONALIZATION (+)</td>
</tr>
<tr>
<td></td>
<td>TWO-PANEL SELECTOR (+)</td>
</tr>
<tr>
<td></td>
<td>LOCALE HANDLING (+)</td>
</tr>
<tr>
<td></td>
<td>WHAT’S NEW PAGE (+)</td>
</tr>
</tbody>
</table>