Journal of Mobile Multimedia, Vol. 4, No. 3&4 (2008) 0165-184

TOWARDS HIGH-QUALITY MOBILE APPLICATIONS BY A SYSTEMATIC INTEGRATION OF PATTERNS

PANKAJ KAMTHAN

Concordia University, Montreal, Canada

kamthan@cse.concordia.ca

Received April 20, 2008
Revised July 7, 2008

The development and evolution of mobile applications is viewed from an engineering perspective. A methodology for deploying patterns as means for improving the pragmatic quality of mobile applications is proposed. To that regard, relevant quality attributes and corresponding stakeholder types for mobile applications are identified. The role of the development process and the challenges in making optimal use of patterns are presented. The underling feasibility issues involved at each step are analyzed. The activities of selection and application of patterns are explored. The use of patterns during macro- and micro-architecture design of mobile applications is illustrated. The implications of the use of patterns in a Mobile Social Web context are briefly highlighted.

Keywords: Evolutionary Development, Feasibility, Pattern Language, Semiotic Quality, Mobile Web Engineering

1 Introduction

In the last few years the ability to access the Internet via a wireless device has been remarkably successful. Indeed, many of the predictions about mobility (Varshney & Vetter, 2002) — the cost, volume, and growth of business; number of users and subscribers; affordability and proliferation of wireless devices such as phones, media players, and Personal Digital Assistants (PDAs); availability of a variety of applications such as electronic learning, electronic mail, flight information and reservation, instant messaging, shopping, sport scores, stock quotes, and weather reports — have largely been realized today. In general, mobile access has opened new vistas for many sectors of society including academia and industry, and continues to play an increasingly vital role in our daily activities of communication, information, and entertainment. In particular, mobile applications have revolutionized the way business is conducted today (Mennecke & Strader, 2003), and reduced the gap between small-and-medium size enterprises (SMEs) and large corporations.

However, mobile applications also have had their share of setbacks and failures (Nguyen, Johnson, & Hackett, 2003), many of which are attributed to quality (Ghosh & Swaminatha, 2001). As mobile applications evolve to become 'social' entities for interaction, collaboration, and participation (Golding, 2008), it is unlikely that the quality-related challenges of the past will simply disappear and, unless addressed, reincarnate and continue to linger in the future. As a side-effect, the sustainability of the successes of mobile applications could be brought into question.

The need for mobile applications to exhibit ‘high-quality’ is critical to all stakeholders involved. If unaddressed, there is a potential for a resource in a mobile application to be inaccessible by a visually impaired user, abruptly freeze upon user interaction, or be prohibitive to perfective maintenance by an engineer.

In this paper, our interest is in a systematic approach of engineering quality-centric mobile applications based on the knowledge garnered from past experience and expertise. For that, we rely on the notion of the problem-solving approach of patterns (Buschmann, Henney, & Schmidt, 2007b). The use of patterns has several advantages over other approaches in terms of quality improvement, including that the approach is preventative rather than curative, is supported by developmental processes, and provides practical, rationalized solutions for problems that have been tackled in the past.

The rest of the paper is organized as follows. In Section 2, we outline the background necessary for the discussion that follows and an overview of previous work. This is followed in Sections 3 and 4 by the presentation of a *Pattern-Oriented Mobile Web Engineering Methodology (POMWEM)* for systematically addressing the quality of mobile applications. In Section 5, challenges and directions for future research are outlined. Finally, in Section 6, concluding remarks are given.

2 Background and Related Work

In this section, the motivation for a systematic approach to addressing the quality in mobile applications and a synopsis of patterns is presented. For the sake of this paper, we define a *mobile application* to be an electronic service to a person with a wireless device such that (1) the person is mobile and (2) the device is connected to the Internet.

2.1 Characteristics of Mobile Applications

There are certain defining characteristics that make mobile applications unique compared to other software applications, and lead to special considerations towards their quality:

- **Social.** From a social viewpoint, the demographic of mobile applications continues to get broader as the cost of mobile access becomes increasingly attractive and wireless devices become increasingly affordable. Indeed, the use of wireless devices (and, by reference, the mobile applications installed on them) has become increasingly common among people of all age groups, educational background, and mental/physical ability.
- **Organizational.** From an organizational viewpoint, the engineering and deployment of mobile applications is particularly challenging due to number of factors including the rise of mobile networks/protocols with varying configurations that can vary across jurisdictions and proliferation of wireless devices from different vendors with diverse specifications.
- **Technical.** From a technical viewpoint, the engineering and deployment of mobile applications is especially challenging in the light of network-specific constraints of variable (and unacceptably low) bandwidth and potential for degraded connectivity, and device-induced constraints of heterogeneity and multiplicity, inhibited power source, small memory, limited processing power, and restricted ergonomic possibilities (such as absence of a pointing device, miniature and minimal keyboard, if any, and relatively ‘primitive’ user interface capabilities).

The context in which an application operates is almost always relevant but even more apparent in a mobile environment due to the number of possible variants that emerge from the aforementioned organizational and technical challenges. The significance of context in designing mobile applications has been emphasized (Tarasewich, 2003).

2.2 An Engineering View of Mobile 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 (Rossi et al., 2008). The past approaches for developing mobile applications have largely been from a managerial viewpoint (Coyle, 2001) and/or from a technological viewpoint (Jones & Marsden, 2006; Heyes, 2002; Hjelm, 2000; Paavilainen, 2002). It is only recently that an engineering viewpoint to developing mobile applications has been undertaken (Gao et al., 2006; Mahmoud & Maamar, 2006; Ocampo et al., 2003; Salmre, 2005).

The increasing scale and significance of mobile applications necessitates a departure from an ad-hoc approach to the development of mobile applications striving for 'high-quality.' Indeed, analogous to Web Engineering, we define *Mobile Web Engineering (MWE)* to be a discipline concerned with the establishment and use of sound scientific, engineering, and management principles, and disciplined and systematic approaches to the successful development, deployment, and evolution of 'high-quality' mobile applications. For the sake of this paper, we will consider a mobile application to be a product resulting from a MWE process.

2.3 Quality of Mobile Applications

There have been a few initiatives for addressing the quality of mobile applications, which we now briefly discuss. It appears that there is a lack of a generic quality model for mobile applications. In some cases, the significance and means of improvement of individual quality attributes like accessibility (Trewin, 2006), credibility (Kamthan, 2007b), and usability (Chan & Fang, 2001; Gorlenko & Merrick, 2003; Weiss, 2002) have been discussed. The relevance of a subset of the quality attributes in the ISO/IEC 9126-1 Standard in the context of mobility has been shown (Spriestersbach & Springer, 2004), however, there is no consideration for certain quality attributes such as reliability, and means of quality improvement are not discussed.

There are guidelines and 'best practices' available for the purpose of cognitive walkthrough and heuristic evaluation of user interfaces of mobile applications (Chan et al., 2002; Gong & Tarasewich, 2004). However, these guidelines do not appear to correspond to any quality model or organized in any known manner; not all guidelines are rationalized; in absence of context, some guidelines can appear unnecessary (such as 'use a flat hierarchy'), constraining (such as 'provide indication of signal strength and downloading progress on every screen') or ambiguous (such as 'do not require users to remember items'); and, as a whole, they do not address certain usability issues. Furthermore, guidelines, in general, are known (Vanderdonckt, 1999) to have several limitations such as being 'absolute,' not illustrating relationships to each other, and oriented towards experts rather than novices. The World Wide Web Consortium (W3C) initiative on the Mobile Web Best Practices suffers from limitations similar to guidelines that it is vague. For example, the recommended practices such as 'take reasonable steps to work around deficient implementations,' 'keep style sheets small', or 'use clear and simple language' are not readily verifiable.

2.4 Patterns for Mobile Applications

The reliance on past experience and expertise is critical to any development. A *pattern* is commonly defined as a proven solution to a recurring problem in a given context (Buschmann, Henney, & Schmidt, 2007b). A unique aspect of a pattern is its structure, which we now briefly discuss.

There are several possible views of a pattern. From a structural viewpoint, a pattern is typically described using an ordered list of elements (henceforth, highlighted in italics) that are labeled as (pattern) *name*, *author*, *context*, *problem*, *forces*, *solution*, *examples*, and *related patterns*. The *name* element of a pattern is often a metaphor reflecting the nature of the *solution*, the *author* element gives the identity of the pattern author(s), the *context* element provides the situation or pre-conditions 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 *examples* element, and the *related patterns* element outlines any other pattern(s) to which a pattern is related to in some way. 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. A pattern is usually referred to by its *name*. In the following, the *name* of a pattern is listed in uppercase in order to distinguish it from the main text.

There are three characteristics of patterns that make them especially suitable for MWE: communicability, practicality, and reusability. A description of a pattern, when properly documented, aims to communicate (1) a practical *process* of arriving at an abstract, broadly applicable, 'best' possible *solution* specific to a *problem* in a given *context*, (2) the reasoning and trade-offs behind this *solution*, and (3) proven *examples* that illustrate conceptual reuse of the *solution* in different circumstances. This often makes patterns more practical in their applicability compared to, for instance, guidelines. Also, patterns often do not exist in isolation, and are part of an overall vocabulary (namely, a 'pattern system' or a pattern language') that attempts to solve a larger problem than that possible by an individual pattern. In due course, this vocabulary of related patterns can become the 'lingua franca' of project teams, thereby contributing to communicability across a team.

Indeed, patterns have been used during the development of certain mobile applications. A collection of patterns to develop Mobile Web Information Systems (MWIS) based on the AvantGo's Dynamic Mobility Model (DMM) has been shown (Risi & Rossi, 2004). It has also been shown empirically that the use of certain design patterns can reduce the cyclomatic complexity of programs aimed for wireless devices (Ihme & Abrahamsson, 2005). However, the relation of patterns to any development process or to a systematic approach towards the improvement of quality is not discussed.

2.5 Development of Mobile Applications and Patterns

There are a few initiatives that introduce process models specific to the development of mobile applications and the role of patterns in it, which we now discuss chronologically. An essentially linear process model for developing mobile applications has been proposed (Mallick, 2003) but it is inadequate in an uncertain environment where risk factors such as flexibility of schedules or changes to requirements need to be accounted for.

Based on a comprehensive literature survey and two development projects, a process model, which we give an acronym OBMP after its authors, for developing mobile applications incrementally has been described (Ocampo et al., 2003). OBMP considers usability in detail and security to some extent, but the approach to quality is not based on a quality model and it does not consider other quality attributes such as reliability or maintainability. OBMP supports the use of patterns in the design phase (only) but the details of actual selection and applications of patterns are not given.

Mobile-D (Ihme & Abrahamsson, 2005) is a process model specially designed for developing mobile applications, and is based on Extreme Programming (XP) (Beck & Andres, 2005), Crystal Methodologies, the Rational Unified Process (RUP), and Test-Driven Development (TDD). It supports the use of patterns during design but the details of the model are sketchy, and the details of actual selection and applications of patterns have not been provided.

Finally, inspired by RUP, a spiral model for developing mobile applications that takes the characteristics of wireless devices and the underlying network into consideration has been presented (Mahmoud & Maamar, 2006). This model recommends the use of patterns in the design phase (only) but the quality-related issues are not considered and the details of actual selection and applications of patterns are not given.

3 A Methodology for Integrating Patterns in Mobile Applications

In this section, we introduce POMWEM, a methodology for systematically addressing the quality of mobile applications from a stakeholders' perspective. POMWEM builds on past experience on integrating patterns in Web Applications (Kamthan, 2008).

The view of a systematic development methodology for the improvement of quality of mobile applications taken in this paper rests on the following interrelated hypothesis:

- **H1.** An improvement in the *process* for development can bring about improvement in the quality of the *product*, namely the mobile application itself.
- **H2.** For an understanding of quality of mobile applications from stakeholders' perspective or otherwise, the notion of quality needs to be decomposed into a manageable number of attributes.
- **H3.** A preventative approach to addressing quality (attributes) is *at least* as significant as a curative approach.

POMWEM is based on the characteristics of mobile applications outlined in Section 2.1 and H1−H3. It consists of the following interrelated and non-linear sequence of non-atomic steps:

1. Selecting the Development Process Model.
2. Identifying and Organizing Quality Concerns from a Semiotic Viewpoint.
3. Acquiring, Selecting, and Applying Suitable Patterns.

The non-linearity of the sequence of steps is crucial to enable revisitation of a step for any necessary refinements. From a practical standpoint, each of the steps 1-3 needs to be feasible. The feasibility study could be a part of the overall planning activity of the mobile application project.

Figure 1 illustrates the development environment of mobile applications from a POMWEM viewpoint. It is evident that pattern can be applied to all phases of the development of a mobile application, including process and product.

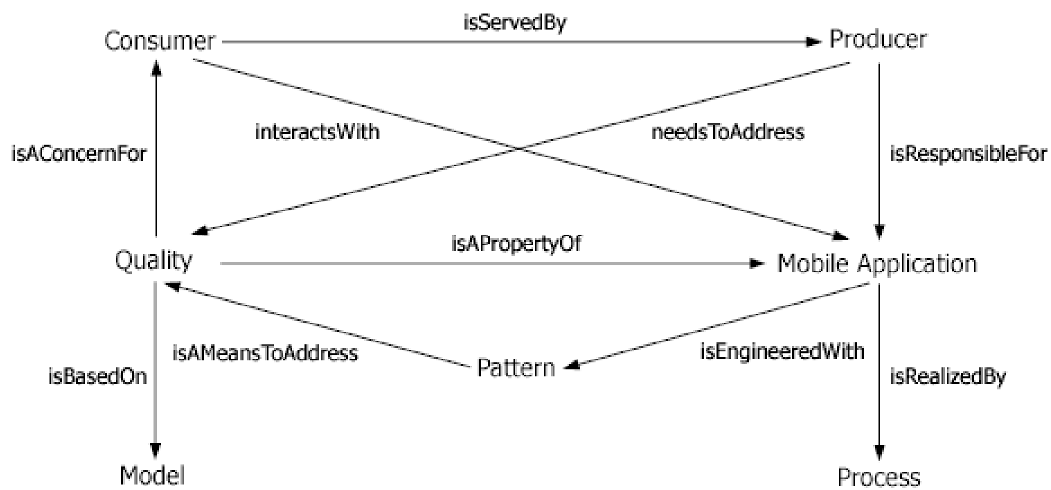

Figure 1. A model of a pattern-oriented and stakeholder-quality centered approach to the development of a mobile application.

4 POMWEM in Practice

In the following sections, we discuss the work involved in each of the steps of POMWEM, including feasibility issues, in detail.

4.1 Step One: Selecting the Development Process Model

A pattern-oriented approach does not mandate the use of a specific development process, however, certain process models can be more enabling than restricting. In order to keep the cost to a minimum, POMWEM is realized within the framework of an existing process environment.

In light of the unique characteristics of mobility discussed in Sections 2.1 and 2.2, the selection of a mobile application development process is based on the following factors: (1) it is flexible, (2) it is human-centered, (3) it is quality-sensitive, (4) it has broad community and tool support, and (5) it has low learning curve. To enable piecemeal growth, it is likely that such a process will typically be non-linear (specifically, iterative and incremental).

The process models from Section 2.5 do not satisfy most of the aforementioned criteria. Therefore, we recommend the adoption of one of the following process models. XP is a mature and broadly-used agile methodology for software development that is applicable to mobile applications (Kamthan, 2007a). XP places ‘lightweight’ requirements on resources and is suited for small-to-medium size projects. The Unified Process (UP) (Jacobson, Booch, & Rumbaugh, 1999) is an archetype of model-based and use case-driven process framework, of which RUP (proprietary), OpenUP and OpenUP/Basic (open source) are three notable customizations. RUP and OpenUP rely heavily on modeling and documentation, and are especially suited for large projects. OpenUP/Basic is agile and suited for small-to-medium size projects. Both XP and UP are likely to be familiar to software engineering students as well as professional software engineers. There is provision of the use of patterns during the design phase in both XP and UP, however, patterns can be integrated in other phases as well. As with any process model, XP, and UP have their own advantages and limitations, the discussion of which is beyond the scope of this paper.

Feasibility of Development Process Model

The adoption of the process model for the development of mobile applications will evidently depend on the constraints of H1−H3 and the organizational process maturity. This in turn involves several factors, including budget, availability of qualified personnel, project schedule, and available tool support. For example, a commitment to preventative quality assurance by means of patterns by an organization is unlikely in a process that is lower than Capability Maturity Model for Software (SW-CMM) (Paulk et al., 1995) Level 3.

4.2 Step Two: Identifying and Organizing Quality Concerns from a Semiotic Viewpoint

In this section, we propose a quality model for mobile applications that is relevant for our purposes. From a semiotic viewpoint, we can view a resource in a mobile application on six interrelated levels (Shanks, 1999): physical, empirical, syntactic, semantic, pragmatic, and social. In this paper, we shall limit ourselves to the discussion of pragmatic level, which is responsible for the relation of signs to their interpreters. The interpreters in our case are the stakeholders of a mobile application.

Based on a systematic approach for identification and refinement of stakeholder classes (Sharp, Galal, & Finkelstein, 1999), we state two broad classes of stakeholders with respect to their *roles* in relationship to a mobile application: (1) a *producer* (provider, manager, engineer, or maintainer) is the one who develops, deploys, or maintains the mobile application, and (2) a *consumer* (novice or expert user) is the one who uses the mobile application for some purpose. This 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 the mobile application are different. Further, more granular classification leading to a stakeholder hierarchy is possible, the discussion of which is beyond the scope of this paper.

For the sake of this paper, we view pragmatic quality as a *contract* between a mobile application and a stakeholder. We contend that pragmatic quality is a multi-dimensional concept and, based on H2, decompose it into granular levels that consist of known *external* quality attributes. The selection of a quality attribute is based on the following factors: (1) it is known to be relevant to stakeholders of a mobile application from earlier efforts in quality modeling; (2) is considered in a standard such as the IEEE Standard 1061-1998, the ISO 9241-11:1998 Standard, and the ISO/IEC 9126-1:2001 Standard; and (3) it can be addressed directly or indirectly (Fenton & Pfleeger, 1997). Finally, we assign patterns as means for improving the quality attributes. Table 1 summarizes this construction.

The relevance of quality attributes in Table 1 varies with respect to stakeholder types (Kamthan, 2008):

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

- **Pragmatic-Tier 2.** The quality attributes of direct concern to an end-user are comprehensibility, performance, and reliability. The quality attributes of direct concern to an engineer is comprehensibility.
- **Pragmatic-Tier 3.** The quality attribute of direct concern to an end-user is usability. For the sake of simplicity, we will view accessibility as a special case of usability. The quality attribute of direct concern to an engineer is maintainability. For the sake of simplicity, we will consider modifiability, portability, and reusability as special cases of maintainability.

<table>
<tr><th>Semiotic Level</th><th>External Quality Attributes of a Mobile Application</th><th>Preventative Means for Quality Assurance</th></tr>
<tr><td colspan="2">Social Quality Concerns</td><td rowspan="5">Patterns</td></tr>
<tr><td rowspan="3">Pragmatic</td><td>[Tier 3]
Maintainability, Usability</td></tr>
<tr><td>[Tier 2]
Comprehensibility, Performance, Reliability</td></tr>
<tr><td>[Tier 1]
Aesthetics, Availability, Efficiency, Familiarity, Readability</td></tr>
<tr><td colspan="2">Physical, Empirical, Syntactic, and Semantic Quality Concerns</td></tr>
</table>

Table 1. A model for the pragmatic quality of a mobile application.

A few remarks regarding the quality model proposed in Table 1 are in order. We contend that the quality attributes in Table 1 are necessary, but make no claim of their sufficiency. (For example, we have suppressed functionality as being one of the quality attributes even though it is present explicitly in the ISO/IEC 9126-1:2001 Standard.)

The quality attributes in Table 1 are not mutually exclusive. Indeed, the quality attributes in Tier 3 depend on that in Tier 2, and those in Tier 2 in turn depend on the quality attributes in Tier 1. For example, if a user cannot read, he/she cannot comprehend the information in a mobile application, and thereby cannot use it or maintain it as desired. Similarly, for a mobile application to be reliable, it must be available. Furthermore, the quality attributes within the same tier in Table 1 are also not necessarily mutually exclusive. For example, the steps taken towards improving reliability (say, fault tolerance) may lead to inclusion of redundant source code or data (that can be unfavorable to maintainability) but enable ease-of-use (that can be favorable to usability).

Finally, we note that the significance of quality attributes will vary across different types of mobile applications. For example, the quality needs of a mobile application providing stock quotes will have some similarity with an application providing currency conversion but will substantially differ from that facilitating text messaging.

Feasibility of Quality Attributes

The expectations of improving the quality attributes of a mobile application must be feasible in order to be practical. Not all the pragmatic quality attributes in Table 1 can (at least mathematically) be *completely* satisfied. For example, an a priori guarantee that a mobile application will be usable to *all* users at *all* times in *all* computing environments that the users deploy, is not realistic. Therefore, the quality requirements of a mobile application must reflect the fact that certain attributes can only be *satisficed* (Simon, 1996).

Furthermore, non-linear relationships between quality attributes necessitates trade-offs between them. (This is especially the case when the quality concerns at the social level are considered.) In such a case, the quality attributes can be prioritized using conventional means (Berander & Andrews, 2005).

4.3 Step Three: Acquiring, Selecting, and Applying Suitable Patterns

In the last decade, patterns have been discovered in a variety of domains of interest, including those that are applicable to the development of mobile applications (Ahlgren & Markkula, 2005; Ballard, 2007; Mazhelis, Markkula, & Jakobsson, 2005; Noble & Weir, 2001; Risi & Rossi, 2004; Roth, 2001; Roth, 2002; Tidwell, 2006; Van Duyne, Landay, & Hong, 2003). The availability of patterns varies significantly: while some patterns are only available commercially in print form, others are available free-of-cost via repositories on the Web.

In general, the relationship between the set of quality attributes and the set of patterns is many-to-many, which necessitates selection of patterns. The underlying *problem* at hand along with the *context* in which it occurs will play a crucial role in selecting desirable patterns. The selection of a pattern is based on the following factors: (1) its maturity (indicated by an associated confidence rating, if any), (2) extent of its availability as ‘open content’, (3) its parity to the goal (namely, relation to a quality attribute in Table 1), (4) quality of its description (Buschmann, Henney, & Schmidt, 2007b), and (5) reputation of author(s).

The main non-mutually exclusive concerns in the application of patterns are the understanding of the pattern description, the order in which patterns are applied, relationships (if any) between patterns, and finally, the result upon the composition of patterns. In particular, if the end result is unsatisfactory, the selection and/or the composition may need to be revisited and revised.

Even though patterns are applicable to any phase of a process and occur at different levels of abstraction, due to considerations of space, we will limit ourselves in this paper to addressing the role of patterns in the design phase of a mobile application. As evident from the discussion that follows, the patterns presented here form a skeleton sequence that traverses through several existing collections of patterns.

Macro-Architecture Design of Mobile Applications

The macro-architecture design is the place where high-level design decisions, independent of any implementation paradigm or technology, are made. To that regard, some general considerations are in order.

A mobile application will implicitly or explicitly target some domain such as education, commerce, entertainment, and so on. There are patterns available for certain common genres like

EDUCATIONAL FORUMS (for educational institutions) and NONPROFITS AS NETWORKS OF HELP (for non-profit organizations) (Van Duyne, Landay, & Hong, 2003). The application of such genre-specific patterns can increase user familiarity with the mobile application. Furthermore, the organization owning a mobile application may wish to serve (potential) consumers in diverse cultural and/or geopolitical situations (such as, in different countries and using different natural languages). This could be done using the LOCALE HANDLING pattern (Busse, 2002). However, the trade-off in deploying this pattern is that the maintenance responsibilities increase.

The macro-architecture design patterns suggested in the following are based on the fact that mobile applications are a class of distributed request-response-type interactive systems. They are selected on the basis of the number of views relevant to stakeholders they can satisfy. Specifically, the applicable patterns are (1) an extension of the CLIENT-SERVER pattern (Schmidt et al., 2000) by an intermediary (that includes a transcoder), followed by (2) the APPLICATION SERVER pattern (Manolescu & Kunzle, 2001), which in turn is followed by (3) the MODEL-VIEW-CONTROLLER (MVC) pattern (Buschmann et al., 1996; Buschmann, Henney, & Schmidt, 2007a).

The CLIENT-SERVER pattern supports modifiability and reusability. For example, a server or resources on the server-side could be modified without impacting the client. Also, a single server can support multiple mobile clients simultaneously, or a mobile client could make simultaneous requests for resources residing on multiple servers. For instance, an Extensible Markup Language (XML) document with normative information could be located on one server, which, upon request, would be transformed via an Extensible Stylesheet Language Transformations (XSLT) document located on another server into an Extensible HyperText Markup Language (XHTML) Basic document and presented using the Cascading Style Sheet (CSS) Mobile Profile style sheet located on yet another server.

The APPLICATION SERVER pattern also supports maintainability: it isolates the mobile application from other aspects on the server-side such that the communication between the application itself and the server takes place via the SINGLE POINT OF ACCESS (Yoder & Barcalow, 1997) pattern. This separation allows the mobile application to evolve independently.

The MVC pattern advocates the separation of structure from logic or presentation of a document, which leads to three semantically-different components: model, view, and controller. The (at least theoretical) minimization of the coupling between these components improves the modifiability of a mobile application. Since a model is normally not aware of the views and controllers attached to it, the *same* model in a MVC could be used with multiple views and multiple controllers, which in turn improves the reusability of a mobile application. For example, the same information could be adapted (repurposed or transformed) and delivered to different situations (like different user agent environments or consumer needs). There are several implementations of MVC available in a variety of programming languages such as Java 2 Micro Edition (J2ME)) and PHP Hypertext Preprocessor (PHP), and application frameworks like the Mobile Asynchronous JavaScript and XML (AJAX) or that accompanying Symbian OS.

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

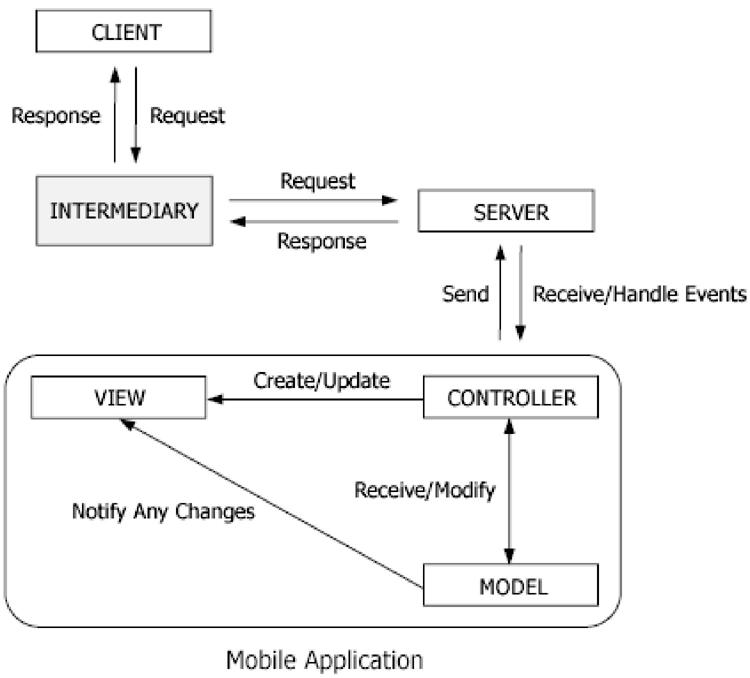

Figure 2. A view of the macro-architecture design patterns in the development of mobile applications.

Reliability Design

For addressing reliability (specifically, availability) concerns, the macro-architecture design of server-side components of a mobile application could use a number of patterns (Ahluwalia & Jain, 2006; Manolescu & Kunzle, 2001). For example, the INTRODUCE REDUNDANCY pattern could be used to include extra measures to support the availability of a mobile application. The FAIL-OVER THROUGH CLUSTERING pattern could be used to introduce redundancy in form of a cluster of multiple servers where if one (primary) server fails, another (secondary) server takes over the responsibility. However, including redundancy comes at the cost of increased maintenance.

If 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 server is available; the absence of such a message would then imply its unavailability.)

Micro-Architecture Design of Mobile Applications

The micro-architecture design is the place where low-level design decisions that are to be implemented are cast. In the following, the focus is only on the design aspects that impact pragmatic quality. As such, the attention is geared more towards client-side rather than server-side concerns.

Interaction design (Jones & Marsden, 2006) is perhaps the most crucial client-side concern among mobile applications. The goal of interaction design is to make both the content and the user interface useful, easy-to-use, and enjoyable. In this paper, we consider four of the most critical interaction design aspects of mobile 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. The details of every pattern for micro-architecture design introduced in the following are not discussed due to considerations of space.

Information Design

The delivery of information could either be static or dynamic. The user interface of a wireless device is usually constrained for space, which is a major concern during information design. The ONE TRUE WINDOW pattern (Noble & Weir, 2001) suggests invocation of a single window at all times within which the information is rendered.

The information being delivered also 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. The information can be further organized using the CATEGORIES pattern (Noble & Weir, 2001).

The USER CUSTOMIZATION pattern (Noble & Weir, 2001) suggests user-specific personalization of information. The APPLICATION STATE MANAGEMENT pattern (Ballard, 2007) and, specifically, the COOKIES MANAGEMENT pattern (Ballard, 2007) can help locally save state in various situations that involve repetitious tasks (like authentication, supplying e-mail address, and so on) and thereby improve performance. However, support for cookies varies across platforms, which can adversely affect portability of the mobile application. The HIDE AND SEEK pattern (Noble & Weir, 2001) can present information at different levels of abstraction, and thus can improve (space) efficiency. However, an overuse of menus to implement this pattern can be detrimental to performance. The ALPHABETIC LISTINGS pattern (Ballard, 2007) suggests a well-known way of organizing information, including search results.

There are patterns that can help if there is an interruption of a user task flow. By providing more details, the DIAL 'H' FOR HELP pattern (Noble & Weir, 2001) helps a user get informed about aspects of the mobile application. To err is human, and a mobile application must account for the possibility of exceptional behavior. This can be accomplished by using the ERROR MESSAGES pattern (Ballard, 2007).

Navigation Design

Any information not only needs to be structured, it also needs a systematic means for being discovered. Navigation is traversal in information space for some purpose such as casual or targeted browsing for information or complementing a reading sequence.

There are various patterns to devise a suitable navigation for mobile applications. The BREADCRUMBS pattern (Ballard, 2007) provides a navigation trail that could be used to inform the user of relative location. The PAGE FOOTER pattern (Ballard, 2007) suggests a horizontal navigation on all documents that are presented to a user and consists of entries that remain invariant during transition. The MENU LAYOUT pattern (Ballard, 2007) provides a classic gopher-style hierarchical navigation of resources.

The SCROLLING pattern (Ballard, 2007) takes advantage of user acquaintance with the scrolling functionality (and thus supports user familiarity). However, scrolling large amounts of data can be slow (which hampers performance). An alternate to scrolling is the use of the TABLE BASED LAYOUT pattern (Ballard, 2007).

Search Design

The goal of searching is finding information. The searching patterns, when used appropriately, aid comprehensibility and performance.

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

Presentation Design

It has been shown in surveys that users highly value the aesthetics of applications on the Web. The elements of presentation apply to all aspects of design discussed previously.

The CAROUSEL pattern (Ballard, 2007) suggests that the mode of information (say, text or graphics) being served should match the nature of information, thereby supporting comprehensibility. Specifically, A PICTURE IS SMALLER THAN A THOUSAND WORDS pattern (Noble & Weir, 2001) advocates the use of icons for well-known concepts to save space and reading time. However, it is not always possible to map concepts to graphical counterparts and icons are not always transferable across cultures.

The FAÇADE pattern, which is a classic object-oriented design pattern (Gamma et al., 1995), can be used to provide a single interface for the user agents of both mobile and for stationary clients. The weak coupling between subsystems and clients results in improved modifiability of the subsystems.

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 visually disabled. Using patterns like FEW HUES, MANY VALUES or COLOR-CODED SECTIONS (Tidwell, 2006) a mobile application could be given a unique 'identity.' There are of course other presentation issues that are crucial such 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.

Figure 3 presents an abstract illustration of the design of a resource in a movie rental mobile application that includes some of the interaction design patterns mentioned previously. The numbers pre-fixed to the pattern *name*s indicate the order of application of patterns, which indicated that the design follows a 'top-down' approach. The design is based on a combination of patterns for information design, navigation design, and presentation design. It shows the dependency among patterns emanating from their selection and use. For example, the selection and use of the ALPHABETIC LISTINGS pattern leads to the selection and selection and use of the SCROLLING pattern.

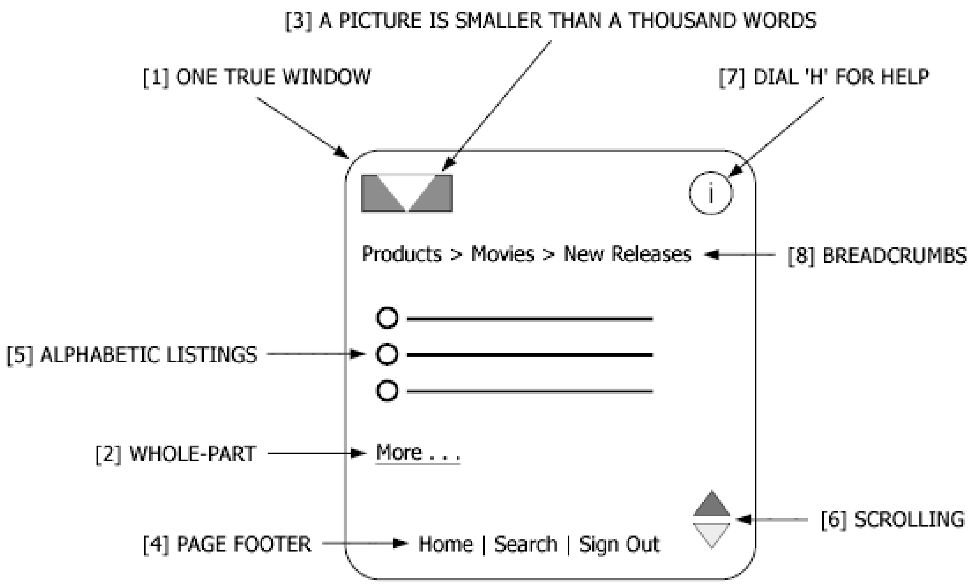

Figure 3. An assembly of interaction design patterns in the development of a mobile application on a PDA.

Table 2 extends Table 1 and provides a mapping between quality attributes and patterns discussed in this paper. A (+) symbol post-fixed to a pattern *name* reflects a positive impact on the corresponding quality attribute, whereas the presence of a (−) symbol reflects a negative impact. The list of patterns is by no means complete and is subject to evolution. The rating scheme can also evolve to become more granular, similar to a Likert scale. For example, (++) could be used denote a strongly positive impact.

Quality Attribute	Pattern(s)
Aesthetics	COLOR-CODED SECTIONS (+) FEW HUES, MANY VALUES (+)
Availability	FAIL-OVER THROUGH CLUSTERING (+) FAILURE NOTIFICATION (+) HEARTBEAT (+) INTRODUCE REDUNDANCY (+)
Comprehensibility	ALPHABETIC LISTINGS (+) CAROUSEL (+) CATEGORIES (+) SIMPLE SEARCH INTERFACE (+) STRAIGHTFORWARD SEARCH FORMS (+)

Efficiency	A PICTURE IS SMALLER THAN A THOUSAND WORDS (+) HIDE AND SEEK (+) ONE TRUE WINDOW (+) ORGANIZED SEARCH RESULTS (+) TABLE BASED LAYOUT (+)
Familiarity	ALPHABETIC LISTINGS (+) A PICTURE IS SMALLER THAN A THOUSAND WORDS (−) EDUCATIONAL FORUMS (+) NONPROFITS AS NETWORKS OF HELP (+) SCROLLING (+)
Maintainability	APPLICATION SERVER (+) COOKIES MANAGEMENT (−) CLIENT-SERVER (+) FAÇADE (+) FAIL-OVER THROUGH CLUSTERING (−) INTRODUCE REDUNDANCY (−) MODEL-VIEW-CONTROLLER (+) WHOLE-PART (+)
Performance	APPLICATION STATE MANAGEMENT (+) COOKIES MANAGEMENT (+) HIDE AND SEEK (−) SCROLLING (−) SELECTABLE KEYWORDS (+) SELECTABLE SEARCH SPACE (+)
Readability	MENU LAYOUT (+) ORGANIZED SEARCH RESULTS (+) TABLE BASED LAYOUT (+)
Reliability	FAILURE NOTIFICATION (+) INTRODUCE REDUNDANCY (+)
Usability	BREADCRUMBS (+) DIAL 'H' FOR HELP (+) ERROR MESSAGES (+) USER CUSTOMIZATION (+)

Table 2. The quality attributes of a mobile application and corresponding patterns along with their impact ratings.

Feasibility of Acquiring, Selecting, and Applying Patterns

The adoption and subsequent deployment of patterns in mobile applications needs to be viable, to which there are a variety of business, social, and technical challenges:

- **Availability of Patterns.** For an adoption of a pattern-oriented approach to the development of mobile applications, it is important that there exist design and implementation patterns that can sufficiently 'map' the solution space. However, there is no *a priori* guarantee that for a given quality attribute, there exist suitable patterns, if any.
- **Locatability of Patterns.** Even when it is ascertained that for a given problem a pattern does exist, that pattern needs to be located. There is currently no unique way of classifying or representing patterns, both of which can pose obstacles in locating desirable patterns. For example, even though

there are patterns available that are suitable for mobile applications, they are not explicitly classified as such. This makes the process of locating desirable patterns somewhat unsystematic.

- **Cost of Using Patterns.** There is cost in terms of time and effort in learning and adapting any reusable knowledge, including patterns. It is not automatic that all patterns are equally 'mature' and the quality of their descriptions varies. For example, as evident from Table 2, the *name*s of patterns in some cases are not evocative, which can make the task of identifying relevant patterns difficult. In our experience, patterns for interaction design published in certain books (Tidwell, 2006; Ballard, 2007) were more thoroughly described than those published in event proceedings. However, the likelihood of corrective maintenance of patterns published electronically tends to be higher than those published in print form. The cost of using patterns needs to be balanced against the (business and/or pedagogical) value that they provide.

4.4 POMWEM in Perspective

It is evident that POMWEM is neither absolute, nor universally applicable. In this section, we briefly discuss the scope and limitations of POMWEM.

First, the feasibility considerations at each step 1-3 determine the scope of POMWEM. This includes the possibility that for a given *problem*, there may not be any suitable candidate patterns available. Second, the quality model presented in Table 1 could benefit from a more rigorous definition. Third, the process of selecting patterns in our case is informal and largely a manual process. It could be formalized, for example, using multi-criteria decision support (MCDS) techniques such as conjoint analysis.

5 Directions for Future Research

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

It is our contention that any initiative towards addressing the quality of mobile applications in general and POMWEM in particular must be based on a common understanding of mobility. This, for example, would require the formulation of a meta-model (such as an ontology) for the domain of mobility. For that, a reference model (Stanoevska-Slabeva, 2003) and the characteristics of mobile applications outlined in Section 2.1 can be useful.

The Oregon Software Development Process (OSDP) (Schümmer & Lukosch, 2007) has been applied to the development of groupware. OSDP meets all the criteria for selecting a mobile application development process except that of maturity and broad community support. It would be of interest to investigate the applicability of OSDP for medium-size mobile application projects.

Table 1 of POMWEM can be extended in both 'horizontal' and 'vertical' directions. A horizontal extension can be achieved by increasing the granularity of the quality attributes in the pragmatic level. A vertical extension can be achieved by, for example, considering quality concerns at the social level. The Social Web, which is more commonly known by the pseudonym Web 2.0 (O'Reilly, 2005), is the apparent 'humanization' and 'socialization' of the Web as it evolves to a medium for participation and collaboration. Furthermore, one of the characteristics of mobile users is that they are 'sociable' (Ballard, 2007). This has evidently inspired the notion of Mobile Web 2.0 (Golding, 2008) or Mobile

Social Web. The quality attributes of interest at the social level are credibility, legality, privacy, and security. The potential benefits of 'collective intelligence' enabled by Mobile Social Web come with their share of negative side-effects (Kamthan, 2007b), and pose various quality-related challenges. Therefore, an extension of POMWEM with respect to quality attributes and the selection of patterns, although of research interest, is not trivial. For example, in its current state, the technological infrastructure of Social Web is not favorable to accessibility. The flexibility of anybody being able to post information raises the possibility for the distribution of inaccurate medical information from unqualified sources, which is a threat to credibility. The potential for unauthorized sharing of media files in a peer-to-peer mobile environment raises legal issues that need to be addressed. Finally, personalization is crucial for mobile applications but can come at the cost of loss of privacy and security. The aforementioned scenarios can resist the selection of certain patterns such as INTERSTITIAL ADS (Ballard, 2007) (unfavorable to credibility) and USER CUSTOMIZATION (Ballard, 2007) (unfavorable to privacy).

The treatment in this paper is limited to the deployment of patterns in the design phase of mobile applications. POMWEM can be extended to incorporate other phases including conceptual modeling, requirements specification, implementation, inspections, and testing. There are patterns available for these phases and have been successfully deployed in non-mobile contexts.

For patterns to continue being useful as a source of guidance and reference, they must be adequately described and documented, freely available and readily findable, and evolve with the needs of the Mobile Web domain. To that regard, there is much to be done in the area of offering a suitable representation of patterns.

Finally, POMWEM to date has been applied to small-to-medium size mobile applications as part of course projects and could benefit from experience outside the academic realm.

6 Conclusion

A MWE view towards the development of a mobile application is necessary for the productivity of engineers, for the longevity of the mobile application, and for the acceptance of the mobile application by its end-users. A lasting view towards quality and systematic means for addressing it are essential to this vision. POMWEM presents a direction to realize that.

At the same time, it is acknowledged that any commitment to quality involves social, organizational, technical, and even political trade-offs. POMWEM is no exception. Indeed, as shown in this paper, a pattern-oriented approach towards mobile application development is neither free, nor is it automatic.

In conclusion, the time- and technology-invariance of patterns enables us to duly plan a pattern-oriented approach that can lead to mobile applications with sustainable architecture and design. From a value-based perspective (Biffl et al., 2006) to MWE, an initial investment in a quality-centric approach to mobile applications using patterns is in interest of all stakeholders where the benefits can outweigh the costs in the long-term.

Acknowledgements

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

References

1. Ahlgren, R., & Markkula, J. (2005). Design Patterns and Organisational Memory in Mobile Application Development. The Sixth International Conference on Product Focused Software Process Improvement (PROFES 2005), Oulu, Finland, June 13-15, 2005.
2. Ahluwalia, K. S., & Jain, A. (2006). High Availability Design Patterns. The Thirteenth Conference on Pattern Languages of Programs (PLoP 2006), Portland, USA, October 21-23, 2006.
3. Ballard, B. (2007). Designing the Mobile User Experience. John Wiley and Sons.
4. Beck, K., & Andres, C. (2005). Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley.
5. Berander, P., & Andrews, A. (2005). Requirements Prioritization. In: Engineering and Managing Software Requirements. A. Aurum, & C. Wohlin (Eds.). Springer Verlag, 69-94.
6. Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., & Grünbacher, P. (2006). Value-Based Software Engineering. Springer-Verlag.
7. Buschmann, F., Henney, K., & Schmidt, D. C. (2007a). Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing. John Wiley and Sons.
8. Buschmann, F., Henney, K., & Schmidt, D. C. (2007b). Pattern-Oriented Software Architecture, Volume 5: On Patterns and Pattern Languages. John Wiley and Sons.
9. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). Pattern Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley and Sons.
10. Busse, D. (2002). Usable Web Design Patterns for World-Ready E-Commerce Sites. CHI 2002 Workshop on Patterns in Practice: A Workshop for UI Designers, Minneapolis, USA, April 21, 2002.
11. Chan, S., & Fang, X. (2001). Usability Issues in Mobile Commerce. The Seventh Americas Conference in Information Systems (AMCIS 2001), Boston, USA, August 5-8, 2001.
12. Chan, S. S., Fang, X., Brzezinski, J., Zhou, Y., Shuang, X., & Lam, J. (2002). Usability for Mobile Commerce across Multiple Form Factors. Journal of Electronic Commerce Research, 3(3), 187-199.
13. Coyle, F. (2001). Wireless Web: A Manager's Guide. Addison-Wesley.
14. Fenton, N. E., & Pfleeger, S. L. (1997). Software Metrics: A Rigorous & Practical Approach. International Thomson Computer Press.
15. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
16. Gao, J. Z., Shim, S., Mei, H., Su, X. (2006). Engineering Wireless-Based Software Systems and Applications. Artech House.
17. Ghosh, A. K., & Swaminatha, T. M. (2001). Software Security and Privacy Risks in Mobile E-Commerce. Communications of the ACM, 44(2), 51-57.
18. Golding, P. (2008). Next Generation Wireless Applications: Creating Mobile Applications in a Web 2.0 and Mobile 2.0 World. John Wiley and Sons.
19. Gong, J., & Tarasewich, P. (2004). Guidelines for Handheld Mobile Device Interface Design. The Thirty Fifth Annual Meeting of the Decision Sciences Institute (DSI 2004), Boston, USA, November 20-23, 2004.
20. Gorlenko, L., & Merrick, R. (2003). No Wires Attached: Usability Challenges in the Connected Mobile World. IBM Systems Journal, 42(4), 639-651.
21. Heyes, I. S. (2002). Just Enough Wireless Computing. Prentice-Hall.
22. Hjelm, J. (2000). Designing Wireless Information Services. John Wiley and Sons.

23. Ihme, T., & Abrahamsson, P. (2005). The Use of Architectural Patterns in the Agile Software Development of Mobile Applications. The 2005 International Conference on Agility (ICAM 2005), Otaniemi, Finland, July 27-28, 2005.
24. Jacobson, I., Booch, G., & Rumbaugh, J. (1999). The Unified Software Development Process. Addison-Wesley.
25. Jones, M., & Marsden, G. (2006). Mobile Interaction Design. John Wiley and Sons.
26. Kamthan, P. (2007a). Extreme Programming for Mobile Applications. In: Encyclopedia of Mobile Computing and Commerce, Volume 1. D. Taniar (Ed.). Idea Group, 277-282.
27. Kamthan, P. (2007b). Addressing the Credibility of Mobile Applications. In: Encyclopedia of Mobile Computing and Commerce, Volume 2. D. Taniar (Ed.). Idea Group, 25-31.
28. Kamthan, P. (2008). A Situational Methodology for Addressing the Pragmatic Quality of Web Applications by Integration of Patterns. Journal of Web Engineering, 7(1), 70-92.
29. Lyardet, F., Rossi, G., & Schwabe, D. (1999). Patterns for Adding Search Capabilities to Web Information Systems. The Fourth European Conference on Pattern Languages of Programming and Computing (EuroPLoP 1999), Irsee, Germany, July 8-10, 1999.
30. Mahmoud, Q. H., & Maamar, Z. (2006). Engineering Wireless Mobile Applications. International Journal of Information Technology and Web Engineering, 1(1), 59-75.
31. Mallick, M. (2003). Mobile and Wireless Design Essentials. John Wiley and Sons.
32. Manolescu, D., & Kunzle, A. (2001). Several Patterns for eBusiness Applications. The Eighth Conference on Pattern Languages of Programs (PLoP 2001), Monticello, USA. September 11-15, 2001.
33. Mazhelis, O., Markkula, J., & Jakobsson, M. (2005). Specifying Patterns for Mobile Application Domain Using General Architectural Components. The Sixth International Conference on Product Focused Software Process Improvement (PROFES 2005), Oulu, Finland, June 13-15, 2005.
34. Mennecke, B. E., & Strader, T. J. (2003). Mobile Commerce: Technology, Theory, and Applications. Idea Group.
35. Nguyen, H. Q., Johnson, R., and Hackett, M. (2003). Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems (2nd Edition). John Wiley and Sons.
36. Noble, J., & Weir, C. (2001). A Window in Your Pocket: Some Small Patterns for User Interfaces. The Sixth European Conference on Pattern Languages of Programs (EuroPLoP 2001), Irsee, Germany, July 4-8, 2001.
37. Ocampo, A., Boggio, D., Münch, J., & Palladino, G. (2003). Towards a Reference Process for Developing Wireless Internet Services. IEEE Transactions on Software Engineering, 29(12), 1122-1134.
38. O'Reilly, T. (2005). What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly Network, September 30, 2005.
39. Paavilainen, J. (2002). Mobile Business Strategies: Understanding the Technologies and Opportunities. Addison-Wesley.
40. Paulk, M. C., Weber, C. V., Curtis, B., & Chrissis, M. B. (1995). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley.
41. Risi, W. A., & Rossi, G. (2004). An Architectural Pattern Catalog for Mobile Web Information Systems. International Journal of Mobile Communications, 2(3), 235-247.
42. Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. (2008). Web Engineering: Modelling and Implementing Web Applications. Springer-Verlag.
43. Roth, J. (2001). Patterns of Mobile Interaction. Third International Workshop on Human Computer Interaction with Mobile Devices (Mobile HCI 2001), Lille, France, September 10, 2001.
44. Roth, J. (2002). Patterns of Mobile Interaction. Personal and Ubiquitous Computing, 6(4), 282-289.

45. Salmre, I. (2005). Writing Mobile Code: Essential Software Engineering for Building Mobile Applications. Addison-Wesley.
46. Schmidt, D. C., Stal, M., Rohnert, H., & Buschmann, F. (2000). Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects. John Wiley and Sons.
47. Schümmer, T., & Lukosch, S. (2007). Patterns for Computer-Mediated Interaction. John Wiley and Sons.
48. Shanks, G. (1999). Semiotic Approach to Understanding Representation in Information Systems. Information Systems Foundations Workshop on Ontology, Semiotics, and Practice, Sydney, Australia, September 29, 1999.
49. Sharp, H., Galal, G. H., & Finkelstein, A. (1999). Stakeholder Identification in the Requirements Engineering Process. The Tenth International Conference and Workshop on Database and Expert Systems Applications (DEXA 1999), Florence, Italy, August 30-September 3, 1999.
50. Simon, H. (1996). The Sciences of the Artificial (Third Edition). The MIT Press.
51. Spriestersbach, A., & Springer, T. (2004). Quality Attributes in Mobile Web Application Development. In: Product Focused Software Process Improvement. F. Bomarius & H. Iida (Eds.). Springer-Verlag, 120-130.
52. Stanoevska-Slabeva, K. (2003). Towards a Reference Model for M-Commerce Applications. The Eleventh European Conference on Information Systems (ECIS 2003), Naples, Italy, June 16-21, 2003.
53. Tarasewich, P. (2003). Designing Mobile Commerce Applications. Communications of the ACM, 46(12), 57-60.
54. Tidwell, J. (2006). Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly Media.
55. Trewin, S. (2006). Physical Usability and the Mobile Web. The 2006 International Cross-Disciplinary Workshop on Web Accessibility 2006 (W4A 2006), Edinburgh, Scotland, May 22, 2006.
56. Vanderdonckt, J. (1999). Development Milestones towards a Tool for Working with Guidelines, Interacting with Computers, 12(2), 81-118.
57. Van Duyne, D. K., Landay, J., & Hong, J. I. (2003). The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley.
58. Varshney, U., & Vetter, R. (2002). Mobile Commerce: Framework, Applications and Networking Support. Mobile Networks and Applications, 7(3), 185-198.
59. Weiss, S. (2002). Handheld Usability. John Wiley and Sons.
60. Yoder, J., & Barcalow, J. (1997). Architectural Patterns for Enabling Application Security. The Fourth Conference on Pattern Languages of Programs (PLoP 1997), Monticello, USA, September 3-5, 1997.