# Chapter 21
# A Generic Architectural Model Approach for Efficient Utilization of Patterns: Application in the Mobile Domain

**Jouni Markkula**
*University of Oulu, Finland*

**Oleksiy Mazhelis**
*University of Jyväskylä, Finland*

## ABSTRACT

*A software pattern describes the core of the solution to a problem that tends to (re-)occur in a particular environment. Such patterns are commonly used as a means to facilitate the creation of an architectural design satisfying the desired quality goals. In this chapter, the practical challenges of efficient usage of patterns in domain-specific software development are presented. The specific domain considered here is the mobile domain, for which is given a sample collection of potentially useful patterns. After that, a novel generic architectural model approach for organizing patterns is presented. In this approach, the identification of relevant patterns is considered as the process of reducing the set of candidate patterns by domain-implied constraints. These constraints can be incorporated in a domain-specific generic architectural model that reflects the commonalities in the solutions of the particular domain. This approach has been validated with a real company application development case.*

## INTRODUCTION

Patterns have been introduced and widely adopted by the software community to support the systematic design of software with desired qualities. In a general sense, a pattern presents a known solution to a recurring problem in a context (Gamma et al., 1995). A pattern in software engineering has been defined as a description of communicating objects and classes that are customized to solve a general design problem in a particular context (Gamma, Helm, Johnson, & Vlissides, 1995). Patterns range in their

DOI: 10.4018/978-1-5225-3422-8.ch021

abstraction level, varying from abstract architectural styles to design patterns and low-level programming idioms (Buschmann, Meunier, Rohnert, Sommerland, & Stal, 1996). They can be divided into general (or domain-independent) patterns (Gamma et al., 1995; Buschmann et al., 1996) and domain-specific patterns. Domain-specific patterns, such as Core J2EE Patterns, are tailored to specific application areas, technologies, development tools, or platforms (Alur, Crupi, & Malks, 2001; Fowler, 2003).

In the efficient usage of patterns, domain knowledge plays a significant role. The domain presents a specific context, which may limit the usage of some general patterns but also calls for domain-specific patterns that are not necessarily valid in other domains. One such specific, highly challenging, present-day domain is the mobile domain, encompassing both technological and application specificities. The mobile domain represents a challenging software design environment caused by technologically imposed and dynamically changing constraints, and by the complexity of applied business models, among other factors.

A large number of patterns have been elicited and distilled during the last decades. Nowadays, patterns are available through a variety of media, e.g., books, journal articles, conference papers, Internet catalogs, and software framework documentation. However, finding the pattern particularly suitable in a specific problem setting and domain has become more difficult. Even if patterns are available that document proven and successful solutions to the problems faced, the practical application of these patterns is often highly challenging. Identifying an appropriate pattern suitable to the problem in the present domain requires experience and often takes too long. The designer would merely avoid taking the risk of searching, learning, and applying a pattern (especially if it is not known beforehand) if considerable time needs to be spent locating the pattern. Better approaches are needed for the efficient use of well-established design knowledge embedded in patterns.

In this chapter, we present, at first, the practical challenges of utilizing patterns in domain-specific application development, focusing on the mobile domain. After that, we give a sample collection of mobile domain patterns and illustrate their description and usage. Finally, we introduce the generic architectural model approach for organizing patterns as a way to solve the challenge of efficient pattern usage. The validation of the approach with a company case of commercial software product development is also presented.

## BACKGROUND

Patterns are now widely used method for documenting and sharing verified design knowledge in the software engineering discipline. Their essential elements include (Gamma et al., 1995): the pattern name, the description of the design problem and its context, the solution describing elements, their responsibilities and collaborations, and the consequences of applying the patterns, e.g., the benefits and trade-offs. There is exhaustive number of patterns available from different sources, in varying quality. These sources include:

- Widely known pattern catalogs published as books. These patterns can be expected to have gone through the review process, as the published patterns above, and/or their authors are known experts in the field. Usually, these patterns are widely acknowledged and used, and, hence, can be useful for documentation and knowledge transfer.
- Patterns and pattern catalogs published in journals, conferences, or workshops. These patterns have been shepherded and then peer-reviewed by others.