Chapter 7
Development of Automated Systems using Proved B Patterns

Olfa Mosbahi
University of Carthage, Tunisia

Mohamed Khalgui
University of Carthage, Tunisia

ABSTRACT
This chapter proposes an approach for reusing specification patterns for the development of automated systems composed of two components: the controller and the controlled parts. The first is a software component controlling the second one that models the physical device and its environment. Specification patterns are design patterns that are expressed in a formal specification language. Reusing a specification pattern means instantiating it and the proofs associated. This chapter shows through a case study how to define specification patterns in Event-B, how to reuse them, and also how to reuse the proofs associated with specification patterns.

1. INTRODUCTION

The idea of design patterns is not restricted to the field of computer science or object oriented software. Design patterns were actually introduced by Christopher Alexander in the field of architecture. In 1977 he spoke of patterns as, “each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” (Alexander, 1977). Related to Event-B, by design patterns we understand former developments that can be reused in the current development. Of course not every development is a good choice for reuse. What we are interested in are generic solutions of common problems. Although reusability is a good reason for having patterns, in Event-B, there is another important point. We

DOI: 10.4018/978-1-4666-3922-5.ch007
aspire the reuse of proofs. A substantial part of
the work when developing in Event-B is proving
the correctness of the models. When reusing an
already proved development, why should one do
the same proofs again? The goal of design pat-
terns in Event-B is therefore: The reuse of already
proved solutions, to refine the problem at hand,
without doing any proofs.

In our research work, we are interested in
the formal development of automated systems.
These systems are composed of two components:
the controlled (operative part) and the controller
(control part). The behavior of the first is continu-
ous and modeled by a discrete model, whereas the
second is software which has discrete behavior
by nature and its goal is to restrict the controlled
component behavior. Controller and controlled
interact to form the automated system (Figure
1) as a closed-loop control one in which control
actions are dependent on its outputs. Modeling
continuous processes can be performed in design
patterns. Indeed, formal specifications are increas-
ingly used in industry and it becomes interesting to
use some of these specifications in new projects.
Reuse a formal specification means first defines
a formal specification pattern and also the way
to combine these patterns in the construction of a
new application. B is a powerful notation that can
make it hard for the newcomer to decide how to
structure and develop a B specification, and hard
for a reviewer or implementer to comprehend a
specification written in an unfamiliar style.

Our motivation is a desire to make B more
usable by commercial non specialist developers
and our reason for investigating patterns comes
from experience in the industrial use of B. B
textbooks introduce the mathematical bases of B,
the notation, and essential elements of the use of
B. However, few books provide advice on how to
“do” B in practice. Illustrations clearly show how
a feature was used by the author, but context and
intent are implicit, and there is rarely any advice
on how to reuse or adapt the B text. The work in
this paper is a new presentation of the concept of
pattern applied to enhance the “semantic structure”
of B, thereby helping the writing; reading and
presentation of B. Formalisms such as B have a
role to play in general software development. So
the patterns should enable:

- Writing of formal texts by generalists, be-
  cause the patterns present formal solutions
to common problems,
- Development of tools to support the use
  of formal methods by generalists, by rec-
  ognizing and assisting in the application of
  patterns, and by breaking down the formal
  concepts into merchandisable or tool-sup-
  portable components.

We have chosen in this paper the B language
to formally specify the notion of specification
pattern for the following reasons:

- Where B is already being used, then there
  is no need to learn a new formalism to de-
  fine and reuse specification patterns,
- B is supported by tools that validate the
  specification. We will use them to validate
  the definition of specification patterns and
Related Content

www.igi-global.com/article/towards-semi-automatic-transformation-process/47385?camid=4v1

www.igi-global.com/article/an-energy-efficient-trust-aware-opportunistic-routing-protocol-for-wireless-sensor-network/199001?camid=4v1

Service-Driven Computing: Challenges and Trends
www.igi-global.com/chapter/service-driven-computing/115421?camid=4v1

Mining Software Specifications
www.igi-global.com/chapter/mining-software-specifications/29406?camid=4v1