ABS: A CASE tool for real time systems design

F. Bernaudin¹, C. Feliot² & L. Pelhate³
¹CSEE Transport, France.
²DUONS-SYSTEMES, France.
³RATP, France.

Abstract

Railways, as well as other industries such as aircraft manufacturers, are involved in the safety of passengers. Nevertheless tools, methodology and practise are often very different from one domain to another domain. In this paper we will explain how we can take the best practises in different domains in order to increase confidence in the produce and reduce the development cost.

1 Introduction

We present a Computer Aided Software Engineering Tool (ABS) based on two different tools: Atelier B from Steria and SCADE from Véritog. The B language supported by the Atelier B is a model oriented formal language which mathematically guaranties the consistency of the software code with regards to its specification. The SCADE tool which supports the synchronous declarative language LUSTRE is well adapted for real time systems design. It provides the system designer with a user friendly graphic interface for modelling. It automatically generates a code from the model. ABS results from a smart merge between formal methods (Atelier B) and synchronous languages (SCADE).

2 In the aircraft domain

For reactive and critical real time equipments design, usual software languages are slowly replaced by specific synchronous declarative languages, which are easier to understand for the final users. The declarative languages consider the software as a set of equations that must be always verified by the program variables. This approach is very close from the one adopted by designers of real-
time control systems using models like systems of differential equations or synchronous operator network.

LUSTRE, which is one these declarative synchronous languages, is the basement of the SCADE™ tool. SCADE™ is the result of a partnership between AEROSPATIALE, SCHNEIDER and CS VERILOG. With LUSTRE, the software specification can be modelled with some finite sets of data and some Boolean equations that establish how the data have to evolve at each clock signal edge.

3 In the railway domain

People are working on methods to tackle with the safety design issue of critical system. The B-method (with its corresponding tool the Atelier-B™) has been heavily used for the design of the new Parisian underground line (METEOR). B is a model-oriented language. According to the B theory, software can be fully described by its data, by some invariant properties (static properties) which express how data are allowed to evolve, and by the operations on these data (dynamic properties).

First at all, B allows the software designer to translate the software requirements into a mathematical model of the system data and of the static and dynamic properties. High abstract level properties such as "a collision between two trains never occurs" can be expressed as well as non deterministic solutions, e.g. "Signals and Switches are controlled in order to prevent collision".

This model includes some Abstract machines which encapsulate variables - whose static properties are expressed through an invariant clause - and operations describing the dynamic aspects (the process at work).

From the abstract model, the tool Atelier-B™ automatically generates the associated proof obligations required by the B theory. This is the key concept of the B method. Proof obligations guaranties that all the static properties remain true whatever are the operations performed on the data.

For example, if one of the static properties is a safety property expressing that a catastrophic situation never occurs (e.g.: “Routes authorised by the interlocking system do not have intersection”), then it should be proved that all operations protect the trains from such a catastrophic situation.

The B language supports incremental specification through “refinements” of the earlier mentioned abstract model. The “refinement” technique provided by B, allows the designer to change the abstract model into a more and more concrete model of the problem until its formalisation has a direct counterpart in terms of executable code.

To ensure the conformance between an abstract machine and its refinement, it must be proved that all the static properties from the abstract specification are fulfilled by its refinement. About the earlier mentioned safety property, it should be proved that the refined specification protects at least from catastrophic situations as the abstract specification does. Such proof obligations are also automatically generated by the Atelier-B™.
Thanks to a mathematical formalisation of the requirements, and with the help of *refinements* and of the associated *proof obligations*, using B in the development of a system provides the following advantages:

- clearing all ambiguities straight from the interpretation of the need,
- writing a specification (the abstract model) coherent and compliant to the need,
- design of the software system, which realises the specification, in successive stages.

The coherence of the model and the conformity of the target code generated with regards to this model are guaranteed by mathematical proofs. The proofs of these obligations in a concrete case can only be considered with the use of automatic proof tools. With Atelier B™, the average automatic proof rate is superior to 80%.

### 4 Merging best practises

The idea of merging B and SCADE, is based on the following observations:

- B is well adapted for high-level properties description such as functional and safety ones through *invariant* clauses.
- B supports specifications refinements, and mathematical proof which guaranties conformance to the requirements.
- a SCADE model can be considered as a refinement of a B abstract machine as SCADE deals with concrete data only.
- SCADE is very close to the equations used to define the critical real time system and allow non real time software experts to design such a system.
- SCADE uses LUSTRE, a synchronous language:
  - It is based on Boolean equations
  - It guaranties the sorting of the equations for implementation on the target computer.

So the aim of this merging is to take benefit of Atelier B

- *system invariant* modelisation,
- proof properties issue from B or SCADE,
- refinement and generation of proved code,
- B mathematically guaranties consistent code generation from the specification.

SCADE

- software specification tool well designed for non software people,
- safety properties specification,
- code generator certified compliant with regards to DO178B.
5 Process description

5.1 Process overview

The ABS tool offers two processes to generate independent soft codes that fulfil the same requirements. The specifications are proved thanks to the abstract model including the safety properties of the system.

5.2 Translation approach (SCADE to B)

The merging is based on the translation between the LUSTRE language (formal language used by SCADE) and B language.

The correspondence between LUSTRE and B language is as simple as possible, in order to reduce drastically the errors generated by this translator tool. So it has to be a syntactic translator. To succeed in merging the synchronous model with the B model some specific criteria have been defined.

To make the translation from SCADE to B, the SCADE node consistency is kept. Thus, each node can be proved thanks to the B abstract machine. So we have as many B abstract machines as SCADE nodes. This correspondence is very useful in term of modularity and reuse of library components.
5.3 Automation of the proof

The development of a project with the B method is based on two activities: writing of the machines and proving of these machines.

The writing of the formal texts implies the specification of each Abstract machine. A B specification includes data (integer, Boolean, combination of data...), services to initialise and to process these data and properties on these data that must always remain true. The way to prove a B specification consists in demonstrating the integrity of the model checked. This activity raises the fact that a feedback is necessary to improve the specification and then the SCADE model. This reinforces the SCADE design safety.

5.4 To summarise

The coupling of LUSTRE and B gets its main interest in the obligation of proving the specification thanks to B. The purpose of proving is double. The first one consists in checking that all the equations, generated from LUSTRE, fulfil the safety requirements whatever are the ranges of the data used. The second one consists in sorting efficiently and in a safety way all these equations.

The part done with the Signalling expert is: (Cf. figure 2)
- "Component Library" which lists all the signalling components (points, track circuits, signal ...)
- "Informal Signalling Description ", which is built from the geographical layout and defines the interlocking system.

The part done with the B-method expert is:
- "Abstract Formal Description", which results from the safety analysis which defines the Safety properties specific to the design system. This part is done in collaboration with the signalling expert.
- "Refinement Proof Rules" is written by the B expert in order to get an automatic proof between the High level description and B1. This is system dependant.
- "Implementation Proof Rules" is written by the B expert in order to get an automatic proof between B1 and B0. This part is not system dependant. It is the full complete theory to prove and translate expressions written with set theory in Boolean expressions.
Figure 2: ABS Computer Aided Software Engineering Tool.
6 Conclusion

As a conclusion we can state that a railway expert can fully do the creation of a computerised interlocking system. The ABS tool is in charge of software generation and safety analysis. The need of a software or safety expert is seldom required. So the cost of design can be reduce drastically without dodging any safety/software issue.

Acknowledgement

This project has been funded by the Ministère de l'éducation nationale, de la recherche et de la technologie MENRT. Companies and research centres involved in this project: CSEE transport, RATP, Steria, Verimag, Verilog.