The RAISE Language, Method and Tools

Mogens Nielsen, Klaus Havelund, Kim Ritter Wagner
Dansk Datamatik Center
Lundtoftevej 1C
DK-2800 Lyngby, Denmark

Chris George
STC Technology Limited
London Road, Harlow
Essex CM17 9NA
United Kingdom

Abstract

This paper presents the RAISE software development method, its associated specification language, and the tools supporting it. The RAISE Method enables the stepwise development of both sequential and concurrent software from abstract specification through design to implementation. All stages of RAISE software development are expressed in the wide-spectrum RAISE Specification Language. The RAISE Tools forms an integrated tool environment supporting both language and method.

The paper surveys RAISE and furthermore, more detailed presentations of major RAISE results are provided. The subjects of these are (1) an example of the use of the RAISE method and language, and (2) a presentation of the mathematical semantics of the RAISE specification language.

1 Introduction

As described in [Prehn87], the starting point for RAISE\(^1\) is VDM ([Bjørner82],[Jones86]) – the Vienna Development Method – probably the most widely used “formal” method for software development. Experience from various applications has revealed a number of problems which seem to complicate the use of VDM in full scale industrial software development projects:

- VDM has until now been a largely paper-and-pencil approach. Real life software development requires a number of powerful, computerized tools supporting the development process.

- The VDM specification language does not have a satisfactory facility for the specification of concurrency. Many applications need to deal with concurrency, in specification, development, and implementation.

\(^1\)RAISE is an acronym for “Rigorous Approach to Industrial Software Engineering”.

• VDM does not have facilities for modularisation of specification and development in such a way, that the development of large software systems can be divided into blocks of a reasonable size, which can then be combined in a well-defined way.

• The VDM specification language has never been given a satisfactory mathematical semantics. Such a semantics is a pre-requisite for a thorough understanding of the language, and for a proof theory allowing reasoning about specifications written in the language.

• The VDM specification language lacks abstraction facilities. The developments of formalisms for property oriented specification of abstract data types ([Futatsugi85],[CIP85],[Guttag85]) have shown the feasibility and usefulness of employing more abstraction than the domain equations in the VDM specification language provide.

These problems have been the motivation for designing a "second generation" formal method for software development. The aim of the RAISE project is to construct a mathematically well-founded software development method, supported by a comprehensive computer based tools which forms an environment for the method and language. RAISE extends and improves VDM in the areas mentioned above. The outcome of the RAISE project is termed the RAISE product and it consists of the following components:

• The RAISE Method for software development,

• the RAISE Specification Language in which the stages of software development can be expressed,

• the RAISE Tools consisting of the tools supporting method and language, and

• the RAISE Documentation, including manuals and educational material for language, method and tools.

RAISE is intended to be used for industrial development of large and complex software systems, an area in which the need for the extensions and improvements of VDM, removing the above mentioned problems, are commonly acknowledged. RAISE is designed to be applicable to the development of a wide variety of software systems. Examples are: embedded real-time systems, network software, data base management systems, application generators, expert system generators, operating systems, compilers, and control and robotic systems.

As it is the case for VDM, the RAISE focus is on supporting the specification, design, and implementation stages of the software development process. There are, however, important implications on most of the remaining development stages from using RAISE, e.g. the maintenance stage will be improved considerably by the presence of a formal recording of the development of the software system to be maintained.

The development of software using RAISE is a stepwise process in which all stages are expressed in the RAISE Specification Language, RSL. Each stage in the process is called a specification and represents the knowledge of the problem and its solution at that stage. The number of steps may vary according to the nature of the problem to be solved and the project organisation.

A RAISE specification is often derived from the preceding specification by constraining the description (commitment), reflecting the fact that a degree of freedom or indeterminacy has been removed. A specification can also be constructed from the preceding specification by taking further requirements into account. In the last step the specification will be transformed into a program written in the programming language chosen for the project.

Within this framework, the use of RAISE can be varied according to the nature of the software project in question and the people involved in the development. RAISE allows the user