Generalization of an Active Electronic Notebook for Teaching Multiple Programming Languages

Mohsen Torabzadeh-Tari, Peter Fritzson, Adrian Pop, Martin Sjölund
PELAB – Programming Environment Lab, Dept. Computer Science
Linköping University, SE-581 83 Linköping, Sweden
{mohto, petfr, adrpo, marsj}@ida.se

Abstract — **In this paper we present a generalization of the active electronic notebook, OMNotebook, for handling multiple programming languages for educational purposes. OMNotebook can be an alternative or complementary tool to the traditional teaching method with lecturing and reading textbooks. Experience shows that using such an electronic book will lead to more engagement from the students. OMNotebook can contain technical computations and text, as well as graphics. Hence it is a suitable tool for teaching, experimentation, simulation, scripting, model documentation, storage, etc.**

OMNotebook is part of the open source platform OpenModelica. It is already used for the course material DrModelica in teaching the Modelica language but can easily be adapted to other programming languages which is also shown in this paper. The notebook can also be adapted to other areas, such as physics, chemistry, biology, biomechanics etc., where phenomena can be illustrated by dynamic simulations within the notebook.

The idea behind this paper is to show that by using a standardized interface the notebook can be extended to any computer language, i.e., being language independent. This is shown in the form of an implementation and adaptation of the notebook to support the Scheme language.

Keywords- OpenModelica, Scheme, DrModelica, Eletronic Notebook

I. INTRODUCTION

In this paper we introduce a research project for generalizing the modern object-oriented equation-based modeling and simulation environment OpenModelica towards also supporting other programming languages than Modelica, [1]. In this way the active student has a common platform for learning programming languages as well as given the opportunity to experiment with physical phenomena by using interactive electronic book, OMNotebook, [2].

This kind of interactive courses based on electronic books allows experimentation and dynamic simulation as well as execution of computer programs. The OMNotebook can contain program code, text, links, pictures, video, virtual and scientific visualizations. OMNotebook is an active electronic book that makes it is possible to integrating applied sciences in physics, human biology [3], mathematics and computer science.

This work was supported by the EU project Lila and by the ITEA2 OPENPROD project.

II. TECHNOLOGY

A. The Modelica Language

In the late 90th people from both the academic community and industry joined forces to define and standardize a multi-domain modeling language, Modelica, [1]. The object-oriented approach for modeling the dynamic behavior of engineering systems was adopted for Modelica, partly based of earlier prototype languages by the designers. In September 1997, version 1.0 of the Modelica Languagewas published on the web. Since then, the Modelica design group has had many meetings, resulting in new versions of the Modelica language.

The freely available Modelica Standard Library (MSL) contains many model components and examples from different application domains. Such components can be easily adopted by the modeler and integrated in his/her own application model. The inherent acasual capabilities built into the language lets the user express relationship between variables which gives more reusability compared to software based on assignment statements and related constructs as in conventional programming languages. A general type system including a general class construct unifies object-orientation, multiple inheritance, components/connectors, and templates/generics.

Furthermore, models in Modelica are described mathematically by Hybrid Differential, Algebraic Equations Equations (HDAEs). The hybrid is referring to the fact that both continuous-time and discrete-time models are handled, [4].

B. The OpenModelica Platform

In 2002 an initiative was taken by PELAB, [2], to develop an open source platform for the Modelica language based on more than 20 years of in-house experience of research in compiler construction. This effort was gradually expanded, and in 2007 an international consortium was formed to support the open source effort. Previously, the only option for the Modelica community, i.e., Modelica users, for simulating Modelica models was to use commercial tools.

The main goal for this open source platform is to create a complete environment for modeling, compiling and simulating Modelica models based on free software. Both the source code and the binaries are freely available and supported for a variety of intended uses in research, teaching as well as in industry.

The platform was originally written in a language called RML (Relational Meta Language), which is a popular

formalism for compiler semantics. This formalism allows efficient compilation combined with optimized C code. This was later (2006) replaced by an extension to Modelica itself, MetaModelica, and the whole compiler was migrated to MetaModelica

The OpenModelica environment compiler translates the Modelica model into a flat Modelica code first and then into C code after a couple of more steps. Also an interactive command handler, i.e., a shell, for executing Modelica scripts and functions etc., is also provided in the environment.

The OpenModelica environment, shown in Fig. 1, consists of several interconnected subsystems. The debugger currently provides debugging of an extended algorithmic subset of Modelica, MetaModelica.

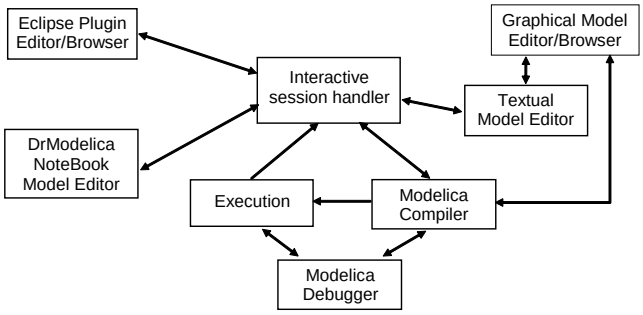

Fig. 1. Illustration of communication between different parts of the OpenModelica platform.

The interactive session handler interface is used in this paper for communication between the OMNotebook and the Scheme interpreter.

C. OMNotebook – the Active Electronic Notebook

The OpenModelica Notebook editor, OMNotebook, provides an active electronic notebook including an editor. The notebook is active in the sense that models inside the book can be changed and executed, it is not just a passive textbook or html page. This is one of the first open source efforts that makes it possible to create interactive books for educational purposes in general, and more specifically for teaching and learning programming. This functionality allows the usage of interactive hierarchical text documents where underlying chapters and sections can be represented and edited. OMNotebook supports functionality for Modelica model simulations, text, images and interactive linking between those. Furthermore, via the external interface, Scheme programs and other codes can be evaluated.

The notebook is currently being used for course material (DrModelica) in teaching the Modelica language and object-oriented modeling and simulation, (see Fig 2), but can easily be adapted to electronic books teaching other programming languages which is demonstrated in this paper. OMNotebook can also easily be used in other areas such as physics, biology chemistry, biomechanics etc., where phenomena can be illustrated by dynamic simulations within the book.

Traditional teaching methods with lecturing and reading a textbook are often too passive and don't engage the student. The option presented in this paper with an active notebook, however, facilitates the learning process, e.g. to run programs and exercises within the book, and mix lecturing with exercises and reading in the interactive book.

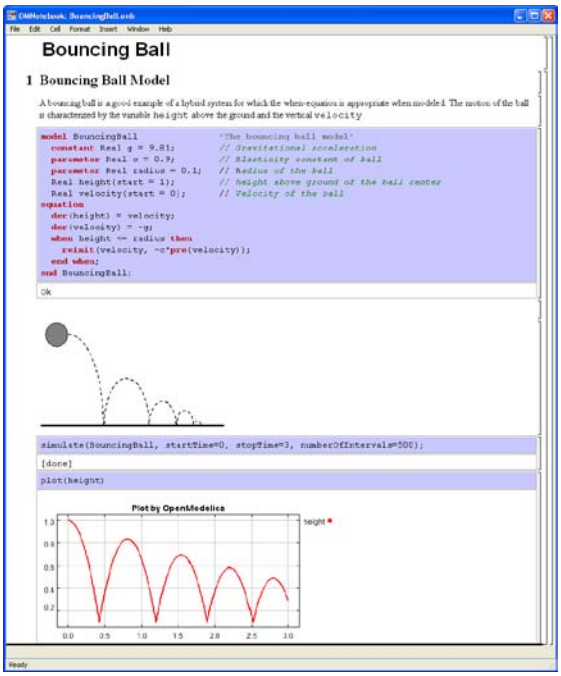

Fig. 2 Bouncing ball example with movement animation in OMNotebook

The hierarchical structure of traditional documents, e.g. books and reports, can also be applied to the notebook which means basically that the book is divided into sections, subsections, paragraphs, etc. This makes the navigation in the book sections easier.

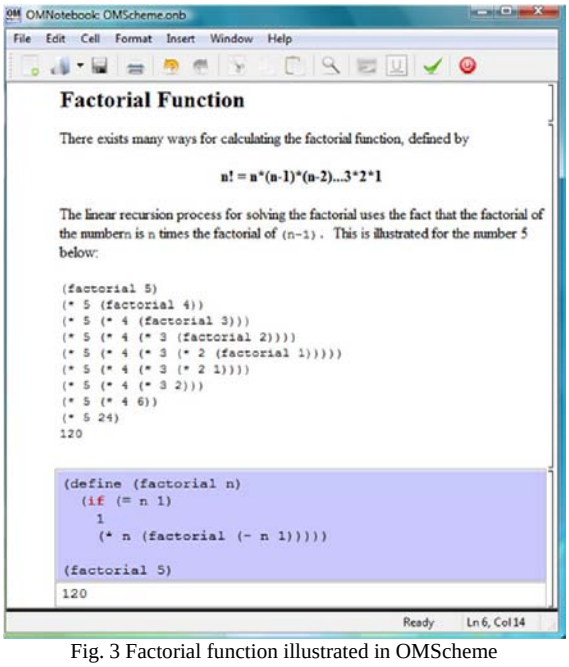

Fig. 3 Factorial function illustrated in OMScheme

In the OMScheme extension of the OpenModelica platform the interactive notebook has been further developed and adapted to the Lisp dialect, Scheme, for creating a suitable educational environment, where students don't have to focus

on different tool and technology but instead focus on the information. An implementation of the factorial function using OMScheme is shown in Fig 3.

D. PLT DrScheme

The PLT Scheme is a dialect of the Scheme programming language, [5]. MzScheme is the interpreter behind the PLT Scheme for compiling syntactically valid programs into an internal bytecode representation before evaluation. The graphical user interface toolkit is named MrEd. DrScheme is an integrated development environment based on MzScheme, i.e., a MrEd application, with support for embedding third-party extensions. DrScheme provides developers with modular development tools, e.g. syntax or flow analyzers. The C API provided by this environment is embedded in OMNotebook and used in this paper for evaluating Scheme programs.

III. CONCLUSIONS

In this paper we outline the basic ideas of an active notebook for educational purpose intended for handling multiple programming languages. An early prototype is being developed for the Lisp dialect Scheme. This interactive E-book, OMNotebook, has been used successfully in both graduate and workshop courses for the Modelica language. OMScheme takes this idea further, combining an additional programming language with innovative teaching concepts.

ACKNOWLEDGMENT

This work was supported by the EU project Lila and the ITEA2 OPENPROD project.

REFERENCES

[1] Modelica Association. The Modelica Language Specification Version 3.1, May 2009. www.modelica.org

[2] Peter Fritzson et al OpenModelica Users Guide and OpenModelica System Documentation, www.ida.liu.se/projects/OpenModelica, 2009.

[3] Anders Sandholm, Peter Fritzson, Varun Arora, Scott Delp, Göran Petersson, and Jessica Rose. The Gait E-Book - Development of Effective Participatory Learning using Simulation and Active Electronic Books. In *Proceedings of the 11th Mediterranean Conference on Medical and Biological Engineering and Computing (Medicon'2007)*, Ljubljana, Slovenia, June 26 - 30, 2007.

[4] Bernhard Bachmann, Peter Aronsson, and Peter Fritzson. "Robust Initialization of Differential Algebraic Equations" In Proc. of (Modelica '06), Vienna, Austria, 2006.

[5] PLT Scheme. Inside: PLT Scheme C API, Release 4.2.2, URL: http://docs.plt-scheme.org/inside/index.html, October 2009.

[6] Anders Fernström, Ingemar Axelsson, Peter Fritzson, Anders Sandholm, Adrian Pop. OMNotebook – Interactive WYSIWYG Book Software for Teaching Programming. In *Proc. of the Workshop on Developing Computer Science Education – How Can It Be Done*?. Linköping University, Dept. Computer & Inf. Science, Linköping, Sweden, March 10, 2006