Open Source Software for Scientific Computation

SCILAB

Research, Development and Applications

Shi-Yin Qin, Bao-Gang Hu,
Shi Li, Claude Gomez (Eds)

Tsinghua University Press Springer

Open Source Software for Scientific Computation

SCILAB

Research, Development and Applications

Edited by
Shi-Yin Qin, Bao-Gang Hu,
Shi Li, Claude Gomez

Selected Papers from 2005 International Workshop
on SCILAB and Open Source Software Engineering
Oct. 27-29, 2005, Wuhan, China

Tsinghua University Press
Beijing

Springer

图书在版编目(CIP)数据

SCILAB研究、开发与应用/秦世引等编. - 北京：清华大学出版社，2005.10
ISBN 7-302-11981-3

Ⅰ. S… Ⅱ. 秦… Ⅲ. 数值计算－应用软件－研究－英文 Ⅳ. O245

中国版本图书馆CIP数据核字(2005)第117559号

出 版 者：清华大学出版社　　地　　址：北京清华大学学研大厦
http://www.tup.com.cn　　邮　　编：100084
社 总 机：010-62770175　　客户服务：010-62776969
责任编辑：陈国新
封面设计：吴　华
印 刷 者：北京嘉实印刷有限公司
装 订 者：三河市春园印刷有限公司
发 行 者：新华书店总店北京发行所
开　　本：165×235　印张：18.5
版　　次：2005年10月第1版　2005年10月第1次印刷
书　　号：ISBN 7-302-11981-3/TP·7759
印　　数：1～600
定　　价：56.00元

Preface

With the development of computer and its popularization, scientific computation has evolved into one of the three magic weapons of scientific research with its own right. In addition, industrials in various domains have considered scientific computation as a key technology. In the light of this ongoing development, it is expected that scientific computation software will play an important role for enlarging wide applications. Scilab, as a free open-source software package for scientific computation, provides a powerful and interactive environment with extensive mathematical capabilities, sophisticated graphics, and high-level programming language for rapid prototyping. Due to its distinguished features of "open source", Scilab has been widely used by more and more engineers and scientists in both academia and industries.

As a matter of fact, since it was developed in 1990s by INRIA (Institut National de Recherche en Informatique et en Automatique) and ENPC (École Nationale des Ponts et Chaussées), France,Scilab has been distributed freely along with the source code via the Internet and used in educational and industrial environments around the world. Meanwhile,Scilab is also being disseminated in all over the world by various introductory documentations in several languages such as French, English, Spanish, Portuguese, Chinese and so on.

Since 2002, an annual event of software development contest based on Scilab platform and environment has been carrying on in China, which is no doubt to promote and advance a worthy cause for the development of open source software engineering, especially for the research, development and applications of Scilab. In order to promote an in-depth research and development in this field and enrich scientific achievements, LIAMA and Beihang University, China as well as Wuhan University of Technology, Wuhan, China jointly organize 2005 International Workshop on Scilab and Open Source Software Engineering on October 27-29, in Wuhan, China under the sponsorship of Chinese 863 Program, INRIA-France, LIAMA, French Embassy in China, CAAI (Chinese Association for Artificial Intelligence). During the workshop the Award Ceremony of 2005 SCILAB Contest is celebrated so that some young and promising prize winners

can take a good opportunity to exchange knowledge and experience with participants in the workshop. It is indispensable and very important to boost up the strength of R&D and further popularize the applications of scientific achievements in this field.

In this book, 22 papers are included, which are selected from papers submitted to 2005 International Workshop on Scilab and Open Source Software Engineering. According to their corresponding topics the papers are classified as 4 different parts which may be introduced as follows.

Part 1 is about some researches on Scilab and Open Source Software Engineering, which contains 5 papers and the major contents deal with the SCICOS code generator; the in-depth analysis of Modeling Capabilities of SCICOS; From Modeling/Simulation with SCILAB/SCICOS to Optimized Distributed Embedded Real-Time Implementation with SYNDEX; Parallel Computing Environment Design Under SCILAB; and Port the toolbox of Scilab image processing for Windows.

Part 2 focuses on the design and development of toolbox for Scilab, in which 5 papers are collected. The development and applications of 3 toolboxes based on Scilab are introduced in 3 papers respectively, which may be listed as a Video and Image Processing Toolbox; a Toolbox for Blind Signal Separation and Independent Component Analysis and a 3D Reconstruction Toolbox. Meanwhile the design and implementation of Cellular Automata Simulation Platform with SCILAB are studied in depth in one paper, and the migrating method from Matlab to Scilab is discussed in another paper.

Part 3 consists of various applications based on Scilab,which are distributed in 7 papers and include numerical studies of nonlinear PID controllers; design and implementation of EDA tool; simulation platform for networked control system; applications in image detection; cultivating virtual plant in Scilab ; fuzzy control of a mobile robot; application of hybrid multi-agent techniques in robotic mapping terrains, and so on.

Part 4 concerns some other related applications with Scilab in 5 papers. It deals

with chaotic pattern search; parameter estimation of nonlinear systems; analysis of thermal effects on modal characteristics; multi-agent cooperation in an E-business intermediation; and simulation and computation in circuit systems as well. Even though these applications do not merely depended on Scilab, it indicates that more extensive applications exist in diverse domains for Scilab after all.

We are grateful to all contributors for making the workshop a success. We would like to express our sincerely thanks to France Telecom, R & D Beijing; Thomson Broadhand R & D Beijing Co., Ltd; and Bull Information Systems Beijing Co., Ltd. for their support. We also thank Mr. Hua Wu for his elaborately composing and design in the publication of this book; and Ms. Hongzhou Chen, Mr. Hongding Liu, Ms. Qian Zhang and Mr. Yongfei Zhang for their beneficial helps in organizing the workshop.

Shi-Yin Qin, Bao-Gang Hu

Shi Li, Claude Gomez

Beijing, Sept. 28, 2005

CONTENTS

Part 1. Researches on Scilab and Open Source Software Engineering

Targeting the Scicos Code Generator: The Linux RTAI Example 3
Roberto Bucher

A Tour of Modeling Capabilities of SCICOS 19
Ramine Nikoukhah

From Modeling/Simulation with SCILAB/SCICOS to Optimized Distributed Embedded Real-Time Implementation with SYNDEX 33
Yves SOREL

Parallel Computing Environment Design Under SCILAB 47
Yi-Min Li

Port SIP for Windows 55
Cheng Zhang, and Ping Yu

Part 2. Design and Development of Toolbox for Scilab

A Video and Image Processing Toolbox for Scilab - V.I.P.Scilab 65
Qi-Ying Wu, Guo-Peng Chen, Zhi-Ming Yuan, Fang-You Zheng, Yan-Zhen Wu, and Xuan-Da Huang

BSSLAB: A Toolbox for Blind Signal Separation and Independent Component Analysis 77
Ding Liu, and Fei Wang

The Design and Implementation of SCILAB 3D Reconstruction Toolbox 85
Yi Huang, Jun-Fei Wu, Ben-Bo Hou, and Ke-Jun Wang

Building Cellular Automata Simulation Platformwith SCILAB 97
Wei-Guang Chen, and Long-Hua Ma

How to Migrate from Matlab to Scilab 107
Farid Belhacene, Vincent Couvert, and Serge Steer

Part 3. Applications Based on Scilab

Numerical Studies of Nonlinear PID Controllers Using Scilab/Scicos 125
Bao-Gang Hu

Design and Implementation of EDA Tool Based on Scilab 141
Dong Zhang, Zhi-Li Zhang, Cai Kang, and Zhen-Zhong He

Simulation Platform for Networked Control System Based on Scilab 151
Hong-Bo Li, Zeng-Qi Sun, Yan Wang, and Ba-Dong Chen
Application of SCILAB in Image Detection 161
Rui Xiao, Xiao-Li Niu, and Ding Liu
Cultivating Virtual Plant in Scilab 167
Meng-Zhen Kang , Rui Qi , Philippe de Reffye, and BaoGang Hu
A Fuzzy Control Based Obstacles Avoidance Strategy with SCILAB for a Mobile Robot 181
Qian Zhang, Yong-Fei Zhang, and Shi-Yin Qin
The Application of hybrid multi-agent techniques in robotic mapping terrains 193
Hua Wu, Shi-Yin Qin, and Yu-Lan Zhang

Part 4. Other Related Applications
Chaotic Pattern Search Algorithm and Its Application 221
Xiao-Fang Yuan, Yao-Nan Wang, and Liang-Hong Wu
Parameter Estimation of Nonlinear Systems Model Based on Hybrid Model PSO Algorithm 231
Liang-Hong Wu, Yao-Nan Wang, and Xiao-Fang Yuan
Analysis with Scilab of Thermal Effects on Modal Characteristics of a Mechanical Structure 241
Maurice Goursat, and Laurent Mevel
The Study on Multi-Agent Cooperation in an E-Business Intermediation Platform 257
Jun-Ping Du, and Wen-Sheng Guo
SCILAB Simulation of Impedance Transform TCXO 267
Ling-Bei Zong, Qian Zhang, and Shi-Yin Qin

Part 1

Researches on

Scilab and Open Source Software Engineering

Targeting the Scicos Code Generator The Linux RTAI Example

Roberto Bucher

Scuola Universitaria Professionale della Svizzera Italiana (SUPSI),
Dipartimento Tecnologie Innovative,
CH-6928 Lugano-Manno,
roberto.bucher@supsi.ch,
WWW home page: http://www.dti.supsi.ch/~bucher

Summary. This paper shows how the original Scicos code generator has been modified to obtain a code for a specific real-time target. The executable code is automatically generated for Linux RTAI, a hard real-time extension of the Linux Operating System. The generated code runs as a normal user space hard real-time application on a standard personal computer, with the RTAI extension of the Linux Operating System. The COMEDI project provides the necessary drivers to integrate acquisition cards into the Scicos scheme. All the stages of the control system design, from analysis to code generation and implementation, can be performed in the same Scilab/Scicos environment. The environment is completed by a set of new Scicos blocks, a library which integrates the RT API and a soft real-time monitor application, to interact with the generated hard real-time task.

A didactic application is presented to demonstrate the capabilities and the performances of this environment.

1 Introduction

The realization of a Rapid Controller Prototyping (RCP) environment for a specific target requires different components:

- The target with a hard real-time operating system.
- A development suite for the specific target (compiler, linker, libraries).
- A Computer Aided Control System Design (CACSD) environment.
- A code generator integrated in the CACSD environment which can translate a block diagram to program code
- A set of drivers to interact with the data acquisition boards on the target.
- A soft real-time application to operate on the generated hard real-time task (monitoring, parameter tuning, start/stop).

The suggested solution exploits a normal x86 PC as target. The Linux operating system with the RTAI add-ons (Real Time Application Interface) [1]

provides the hard real-time performances to the controller task. The access to the hardware (I/O) is performed using the drivers from the COMEDI project [2]. The development tools are already included in the normal Linux OS. The Scilab/Scicos suite [3] with a modified Scicos code generator performs all the CACSD tasks. Finally, Linux RTAI also provides the needed software to connect, monitor and interact with the real-time task (RTAI-Lab).

Section 2 gives an overview of the Linux RTAI system and of the RTAI-Lab framework. Section 3 explains how the original code generator has been modified to generate codes for the Linux RTAI environment. A laboratory example is presented in Section 4. Finally, the main results are briefly discussed in section 5.

2 The target

2.1 Linux RTAI - A real-time extension to Linux

The work presented in this paper exploits the RTAI extension to the Linux Operating System (OS) developed at the Dipartimento di Ingegneria Aerospaziale of Politecnico di Milano (DIAPM). This extension adds hard real-time features to a Linux OS, allowing sample frequencies up to several thousand of cycles per second, with a jitter of few microseconds. Professor Paolo Mantegazza started the RTAI (Real Time Application Interface) project in 1999. Since the version 3.1 Linux RTAI is based on the Philippe Gerum's Adeos (Adaptive Domain Environment for Operating Systems) nano-kernel [4], [5].

The RTAI extension was created as an environment for implementing low cost data acquisition and digital controller systems [6].

Different projects and industrial applications take advantage of Linux RTAI:

- Primaindustrie developed the numerical control of the 2D laser cutting machine "PLATINO" [7].
- The Orocos project: this project aims to develop an open robot control environment for generic robot devices (manipulators, mobile robots, humanoids, ...) [8].
- The ALMA project (Atacama Large Millimeter Array [9], where a cluster of 16 Linux RTAI nodes performs data filtering, data windowing, fast Fourier transforms and phase corrections at a processing rate of about 1 GFLOPS.

One of the most important features of Linux RTAI is the possibility to implement hard real-time code in both kernel and user space area, using a single RTAI scheduler. With modern CPU's, the loss of performances of a user space hard real-time task is negligible compared to the same task in kernel space, but it offers a lot of advantages (for example code debugging).

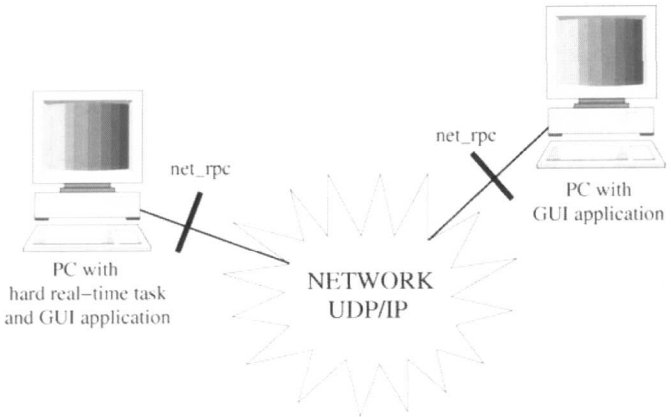

Fig. 1. RTAI-Lab with remote monitoring of the hard real-time task

2.2 RTAI-Lab

Linux RTAI features a tool called RTAI-Lab, providing a structured framework for the integration of RTAI into CACSD environments [10]. The current implementation includes support for the commercial MATLAB/Simulink/Real-Time-Workshop suite [11] and the open source Scilab/Scicos suite. Its internal architecture allows easy porting to other CACSD software. Basically, RTAI-Lab relies completely on the CACSD software for control system design and code generation. It only provides some specific blocks and building options. The generated code is embedded in a RTAI framework and can be executed in soft or hard real-time and monitored by an external GUI application.

At present, two GUI applications exist. Both can be run locally or remotely on the network.

xrtailab

This application is distributed with the Linux RTAI package. This tool allows the monitoring and tuning of real-time tasks from any PC in the LAN (see Fig. 1).

The GUI application *xrtailab* provides three types of display instruments: digital scopes, LEDs and meters. Each instrument is defined by a specific block in the Scicos library.

Fig. 2 shows the running GUI application.

xrtailab has been implemented using OpenGL (Mesalib) and the Extended Fast Light Toolkit (EFLTK).

ARTIST

"A Real-Time Interactive Simulink-based Telelab" is a project developed at the Dipartimento di Sistemi e Informatica of the University of Florence [12].

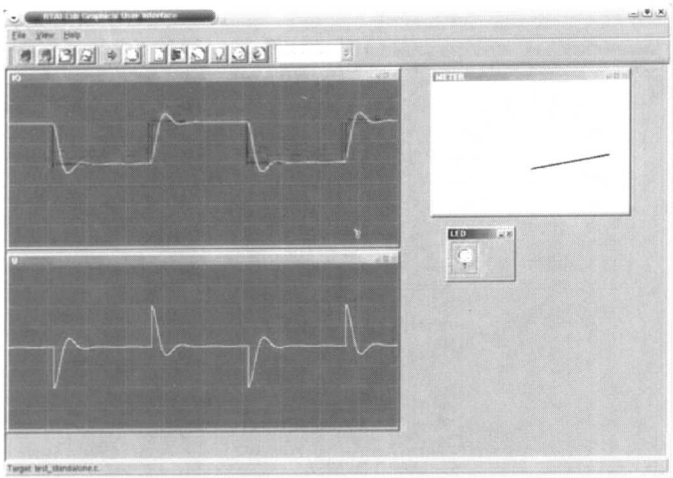

Fig. 2. RTAI-Lab graphical user interface

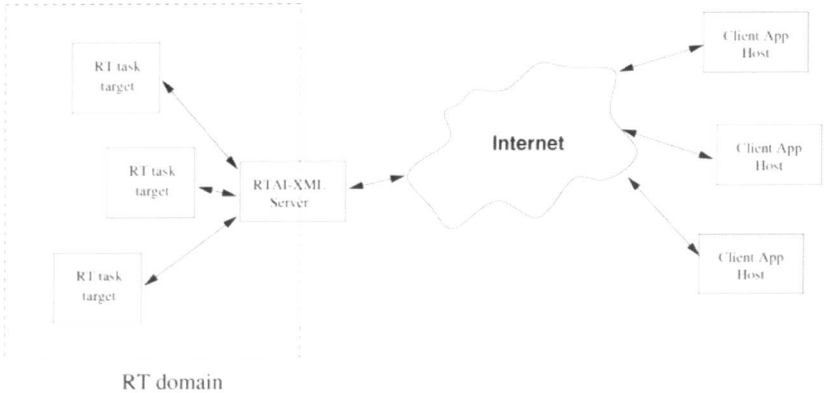

Fig. 3. RTAI-XML server

One of the limitations of the original xrtailab application is that the net_rpc protocol used between the hard real-time task and the GUI application requires Linux RTAI OS be installed on the client. ARTIST introduces a server between the hard real-time task and the network to route all messages from net_rpc to xml-rpc and to the socket protocol. The client application can be now implemented under any operating systems. Fig. 3 shows the architecture of the RTAI-XML server.

RTAI-XML implements an RPC server based on XML; inside the RTAI domain, RTAI-XML connects directly to the target, using RT calls.

A Java applet called *jrtailab* is provided as an example for a client application. It can be started using a standard web browser (see Fig. 4).

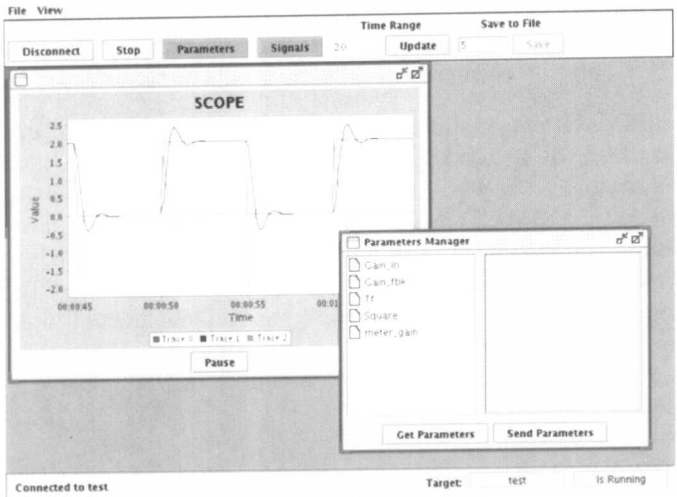

Fig. 4. The jrtailab application

2.3 Accessing data acquisition boards

Data acquisition boards can be accessed through the drivers provided by the COMEDI project. A set of drivers for a variety of common data acquisition plug-in boards has been implemented. They are implemented as a core Linux kernel module providing common functionality and individual low-level driver modules.

In addition to the low-level acquisition boards drivers, COMEDI provides two basic modules:

Comedilib

it is a user-space library which provides a developer-friendly interface to COMEDI devices. Included in the Comedilib distribution are documentation, configuration and calibration utilities and demonstration programs.

Kcomedilib

it is a Linux kernel module which provides the same interface as Comedilib in kernel space, suitable for real-time tasks.

2.4 Installing Linux RTAI and the Scilab/Scicos add-ons

A detailed description about installing the full system is given in [13]. This document describes step by step how to install Linux RTAI, RTAI-Lab, COMEDI and the Scilab/Scicos add-ons. At present, this is the most complete document about installing the RTAI-Lab environment.

2.5 RT targets

At the SUPSI laboratory there are different kinds of target systems:

- Pentium IV systems, installed with a full Linux distribution (most Debian based), which can be used to develop, run and monitor the real-time tasks. These systems can access external hardware using data acquisition boards covered by the COMEDI projects, or with driver implemented by the users.
- Embedded systems, where the full OS requires about 10 MBytes, including LAN support. These systems are based on PC-104 or Compact-PCI hardware, and at present they are used to implement high precision controllers for voice coil motors (nanocontrol). Design, code generation and monitoring tasks are performed on a second PC.

3 The RTAI Code generator

3.1 The starting point - CodeGeneration_.sci

In spite of an excellent C-code generation, the original Scicos function "CodeGeneration_.sci" presents some disadvantages:

- The code of the "main" procedure of the "stand-alone" executable is directly implemented in the code generator, and is not suitable for real-time tasks.
- The code of the "Makefile" code is encoded in the code generator. Different targets require a different "Makefile", thus a new approach is needed.
- Access to the block parameters is quite complex because they are not identified by a name in the generated code.
- Input signals (sine wave, square wave, step) from the original Scicos palette cannot be used as input signals for the generated code.
- Sensors and actuators must be integrated by hand in the generated code.

3.2 The new generator - RTAICodeGen_.sci

In order to fit the generated code to a specific target, a more flexible code generator has been implemented by the author. The new code generator contains the original procedures needed to translate the Scicos block diagram into C-code, but introduces some improvements to integrate the code into a specific target environment:

- The Makefile is generated starting from an external "template". The user can implement different "template makefiles" for different targets. This also allows new libraries to be integrated in the project with a user specific code.

- The Scicos "Identification" fields has been used to identify a block and his parameter with a unique name. This approach simplifies the tuning of the parameters from an external application (for example "xrtailab").
- The "main" procedure is implemented as external file, and can be fitted for the chosen target. In Linux RTAI, the main file "rtmain.c" contains all the procedures needed to run the generated code in hard real-time and the procedures needed to connect the real-time task to the monitor application.
- Basically, all the sensors and actuators are realized directly using Scicos blocks. In case the Scicos "superblock" contains input or output ports, the <MODEL>_io.c file has a better readable and modifiable code.
- The code generator directly compiles and links the real-time task.

3.3 The template Makefile

The directory .../macros/RTAI/RT_TEMPLATES contains the template makefile "rtai.mak", specifically for the Linux RTAI target. In this file, some keywords (for example the keyword "$$MODEL$$") will be dynamically substituted by the code generator with the needed values.

3.4 The rtmain.c file

The core of the project is represented by the *rtmain.c* file. This file contains the "main" procedure of the project and integrates all the functions needed to run the generated code in real-time.

The main procedure starts two threads:

The real-time thread

the thread *rt_BaseRate* performs the initialization of the generated code and then switches to hard real-time to trigger the periodic task (ISR). When the real-time task is terminated, this thread switches to soft real-time again and performs the termination activities of the task. The ISR procedure contains a call to the *WaiTimingEvent* function, which is responsible for guaranteeing the respect of real-time performances.

The communication thread

the *rt_HostInterface* thread is used to exchange information between the real-time task and the monitor application:

- Start/Stop of the real-time task.
- Upload of the parameter values.
- Modification of the task parameters (parameter tuning).