C-Sim - The C Language Enhancement for Discrete-Time Simulation

Jan Hlavicka, Stanislav Racek
Czech Technical University in Prague, University of West Bohemia in Pilsen
hlavicka@fel.cvut.cz, stracek@kiv.zcu.cz

Abstract

The paper presents the C-Sim simulation environment, which enables the execution of several processes in an interleaved mode using the global simulation time concept. C-Sim was used within the EU/IST project Fault Injection for Time Triggered Architecture (FIT) to build a simulation model of TTP/C protocol based real-time embedded computer system in order to verify its dependability through fault injection.

1. C-Sim characteristics

C-Sim is a simple program enhancement of the C language (ANSI specification) used for creating discrete-time simulation models based on the method of pseudo-parallel processes. C-Sim kernel has the form of a library of basic object types and operations on them. The goal of its design was to create a portable and to a certain degree machine independent tool for computer implementation of discrete-time systems simulation. The idea was taken from the programming language SIMULA. The C language, unlike SIMULA, has no tools for object-oriented programming but some of the basic principles were implemented using predefined macro commands.

2. C-Sim portability

C-Sim kernel is ANSI-C portable, so C program using only the kernel can be made ANSI-C portable as well (it assumes only the use of I/O capabilities of the standard C libraries). Most of C++ compilers accept such program as well. It is possible to create customized run-time environments in order to manage a class of simulation experiments. Simple run-time environments for MS-DOS, X-Windows and NT Workstations have been created so far. A complex application-dependent run-time environment (including e.g. model visualization) can be constructed as a separated SW layer.

3. C-Sim application area

C-Sim can be used in all the traditional areas of discrete-time simulation, e.g. in the area of queueing networks modeling. Moreover it can be used for non-traditional applications of discrete-time modelling, e.g. verification of properties of parallel programs and systems, communication protocols, fault-tolerant systems and algorithms, algorithms of real-time safety-critical applications, etc. C-Sim based simulation program can describe both the functions of system HW (at a chosen level of abstraction) and SW (here the C-coded parts of an application source code can be directly used as a part of C-Sim based simulation program code). C-Sim is freely accessible at www [1] including a user manual.

4. C-Sim case study

Within the EU/IST project Fault Injection for Time Triggered Architecture (FIT) we used C-Sim to build a simulation model of TTP/C protocol based real-time embedded computer systems in order to inject faults and to follow their influence [2]. The source code of the model has two basic SW layers:

- C-reference model of TTP/C protocol: contains C-language coded TTP/C specification, including data types (CNN, MEDL) and functions (TTP/C services).

- C-Sim code of a program of (abstract) TPC controller activity: this program serves as a template to create the required number of TTP/C processes (i.e. abstract controllers) as a part of the TTP/C cluster model.

The process-oriented form of the simulation model enables us to include easily other processes, like e.g. application processes (a node host processor computation including its operating system TTP0), processes of controlled object, etc. As the main testing application we used BBW-4W (Brake-By-Wire for 4 wheels) that was developed by Volvo company, which is one of participants of the FIT project. Another case study implemented in C-Sim was presented in [3].

5. References