CLAIRE: An Event-Driven Simulation Tool for Test and Validation of Software Programs

Adriana Carloganu and Jacques Raguideau
CEA Saclay, DTSI/SLA, 91191 Gif sur Yvette Cedex, France
adriana.carloganu@cea.fr and jacques.raguideau@cea.fr

Abstract

Malfunctions of systems in domains such as medicine, avionics, traffic control, defense and nuclear applications can cause human injuries. Test and validation of such systems is a difficult task, because many situations cannot be safely reproduced. Simulation makes possible to assess the correctness of a safety-critical system, even in dangerous situations.

This paper presents CLAIRE, a purely software simulation tool with graphic facilities for system modelling, designed for test, validation and non-intrusive dynamic analysis of real time applications.

1. Introduction

CLAIRE was developed at CEA (www.cea.fr), the French research institute for nuclear applications. It was designed to perform validation and test of the specifications and executable code for real time distributed systems. It can also be used to observe and provide dynamic control of the simulation process and to record data information for off-line analysis. The remainder of this paper summarizes the principles, roles and facilities of the tool and gives some practical results of the experience in its use.

2. Tool Design and Facilities

The tool design is based on the following concepts: event, data flow model and purely software simulation.

Event-driven simulation was chosen to fulfil the requirements for code validation: an event is associated with an action, which sets the system into a predefined state. Such events can be the execution of a particular instruction, the access to memory address, etc. With such events, the user can associate reactions like: sending a message, stopping the execution, etc.

With CLAIRE, can be created a description of the system using a top-down hierarchical technique, with no limit on the number of levels. The low-level boxes can contain user actions. The data flow is represented by the lines connecting boxes. The exchanged data information can be continuous (accessible at any time) or discrete (associated with an event). An event carries information about data variables, the time of assignment and triggered functions.

The tool relies on a purely software simulation; this means that the environment and the hardware on which the binary code under test will run are entirely simulated. Therefore, the tool-library includes microprocessor simulators (Motorola 68000 Family, Intel 586, etc) and allows the user to add components. Each simulator drives the passage of time using instruction execution-time information. The lines connected to a microprocessor simulator can contain variables used to inspect the binary code running on the processor, without any instrumentation being necessary.

The user can describe test strategies in low-level boxes, dynamically modify input data, microprocessor memory or registers, inject faults in order to assess the system behaviour in different conditions.

During simulation, the user can enable/disable trace files to store modified variables, created events, microprocessor internal states, binary code coverage, and user actions. CLAIRE offers facilities for trace file display and analysis.

CLAIRE was written in the C language and is currently available for Unix and Linux machines.

3. Practical Experience

The French Institute for Nuclear Safety and Radioprotection (IRSN) uses CLAIRE to perform safety analysis and evaluations of large real-time distributed systems used for instrumentation and control in nuclear power plants. Systems of up to 8 microprocessors were modelled and submitted to consistency and robustness checks.

CLAIRE is currently being evaluated in different application domains: avionics, smart card, transport.