Abstract

In today's world, embedded systems are everywhere: from alarm systems to car machinery and everywhere in between. If you stroll the streets, you are bound to realize that you are surrounded by devices that connect people to the internet or computing power. Embedded systems have become a vital part of a growing range of automotive, aerospace, biomedical and military systems. Concurrently, these embedded systems have become much more complex, in part because more powerful processors are now inexpensive enough to be included in cars, cameras, kids toys and other products. These embedded devices, with more processing power and memory, typically perform a complex set of functions, or even several functions. As many of these applications are potentially life-threatening, the need for an appropriate design approach has never been more compelling.

The central state of embedded systems design is to develop concepts, methods and tools that can master their growing complexity and at the same time reduce development costs and time to market. In fact, it is becoming a trend to use the Unified Modeling Language (UML) to design embedded software.

We are convinced that software errors are likely to be introduced in the design phase and that these errors have a lasting impact on the reliability, cost and safety of a system. We propose a UML-based verification method to identify and remove errors, misconceptions, etc. during the design phase of embedded software development.

Requirements

A design is a preliminary sketch. It is an outline of the main features of the system's components or how the system is supposed to execute. Several diagrams are used for this activity.

Like any other embedded system, a TV is event-driven, meaning that it continuously waits for the occurrence of some external or internal event (power on or power off). After recognizing the event, such systems react by performing the appropriate computation such as changing to the appropriate state. E.g. switch to teletext.

Properties to check

- When the off event is being generated, the TV will be in state OFF and vice versa.
- An expectation phrase is transformed into temporal logic formulae, through an assistant that guides the user in writing the properties.

Verification

a) Transformation from State Chart to Extended Hierarchical Automaton (EHA)

b) Verifying embedded behavior through model checking and with(out) slicing

Symbolic model checking is a powerful formal verification technique for reactive systems, but sometimes slicing can be necessary for improving the time and space efficiency of symbolic model checking for systems specified as state charts. Using a property, slicing removes those parts of the EHA irrelevant for verifying the property.

From EHA, a formal representation in CadSMV is automatically obtained.

CadSMV (Cadence Symbolic Model Verifier) uses symbolic model checking for the verification. The means that the test is automatic, always obtains an answer and more importantly, should the property not be satisfied, it generates someone of identifying the originating error (counterexample).