TESTING A SAFETY-CRITICAL APPLICATION*

John C. Knight, Aaron G. Cass, Antonio M. Fernández, Kevin G. Wika

Department of Computer Science, University of Virginia
Charlottesville, VA 22903

As part of a case study, we are developing software for an experimental safety-critical application. The case study is *The Magnetic Stereotaxis System* (MSS), an investigational device for performing human neurosurgery being developed in a joint effort between the Department of Physics at the University of Virginia and the Department of Neurosurgery at the University of Iowa.

The system operates by manipulating a small permanent magnet (known as a "seed") within the brain using an externally applied magnetic field. By varying the magnitude and gradient of the external magnetic field, the seed can be moved along a non-linear path and positioned at a site requiring therapy, e.g., a tumor. The magnetic field required for movement through brain tissue is extremely high, and is generated by a set of six superconducting magnets located in a housing surrounding the patient's head.

A key element of the device is the imaging subsystem. It uses two X-ray cameras positioned at right angles to detect in real time the locations of the seed and of X-ray opaque markers affixed to the patient's skull. The X-ray images are used to locate the objects of interest in a canonical frame of reference, and this information is used to display graphic representations of the seed and skull markers on pre-operative magnetic resonance (MR) images. The MR images are the primary source of information used by the surgeon for making control decisions. Clearly, failure of the MSS control software could have serious results.

We intend to exploit a variety of approaches to verification and we have begun consideration of how the software should be tested. Many testing issues are raised by this application because:

- Although it adds undesirable complexity, the system uses a distributed architecture and depends upon significant amounts of "off-the-shelf" software that is not under our control.
- The X-ray imaging system and other high-energy electrical devices are not available for testing on a routine or extended basis.
- The system is interactive and requires extensive operator direction via a graphic user interface. The "operator" in this case is a physician.
- Correct operation of the system is very difficult to determine because a variety of complex calculations are required to perform the necessary control. The application has all the appearances of being untestable [1].

We have addressed the problem of device unavailability in a conventional manner by developing synthetic devices. For example, high fidelity X-ray images are produced by the test harness. Similarly, the requirement for interaction is eliminated by the provision of a "pseudo-user" that accepts high-level commands from the test harness as part of a test case and injects them into the interface.

We have addressed the problem of correctness determination by the use of *reversal checks* on the entire system. A reversal check computes a program's input from its output and compares this with the actual input. The current calculations for the superconducting coils, for example, begin with a required force and are very complex. Computing the force resulting from the coil currents, however, is simple and provides the exact inverse of the current calculations. Thus the input can be computed and compared.

A variation on the idea of a reversal check is used by the imaging subsystem. Using the images provided, its complex forward computation determines the locations of objects within the field of view in a canonical frame of reference. However, the test harness sythesizes these images from prescribed locations in the exact same reference frame. Thus the expected output of the imaging system is the starting point for image synthesis and the two can be compared to check functional correctness. By combining reversal checks, the majority of the critical software's operation can be checked. Naturally these checks will remain during operation.

The result that verification of safety-critical systems by testing is infeasible due to the number of tests required implies that testing has no real value for the MSS [2]. That result, however, applies to functional correctness. For safety-critical applications, we believe that testing to demonstrate selected system properties, no matter how limited, *is* valuable and feasible because what amounts to exhaustive testing can be employed for these properties.

Testing of this application employs a technique we call *specification limitation.* By this we mean that the specification for the application is deliberately limited in several areas to restrict the total number of test cases. For example, the angles entered by the operator for the required direction of motion are rounded to 1/10 of a degree. In practice, this is not a significant functional restriction but it permits exhaustive testing of the angles used for setting direction. The same approach is used with distance.

[1]Weyuker, E. J., "On Testing Non-Testable Programs," *Computer Journal* Vol. 25-4, November 1982.

[2]Butler, R. W. and G. B. Finelli, "The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software," *IEEE Trans. on Soft. Eng.*, Vol. 19-1, pp. 3 - 12, January 1993.

* Supported in part by the NSF under grant number CCR-9213427 and in part by NASA under grant NAG1-1123-FDP.

ISSTA 94 - 8/94 Seattle Washington USA
© 1994 ACM 0-89791-683-2/94/0008..$3.50