Software Verification and Validation

Ms. Vinita Rohilla & Ms. Shaily Malik
Vinita.rohilla@gmail.com, shaily_malik984@yahoo.co.in

EXTENDED ABSTRACT
In the development of an embedded system, it is important to be able to determine if the system meets specifications and if its outputs are correct. This is the process of verification and validation (V & V) and its planning must start early in the development life cycle. Both aspects are necessary as a system meeting its specifications does not necessarily mean it is technically correct and vice versa. There are many different V & V techniques which are applicable at different stages of the development life cycle. The results of V & V form an important component in the safety case, which is a document used to support certification. Certification is usually pursued due to either legal reasons or economic advantages. The certification process also starts from the beginning of the life cycle and requires cooperation between the developer and regulatory agency from the very start. Thorough V & V does not prove that the system is safe or dependable, and there is always a limit to how much testing is enough testing. In addition, certification does not prove that a system is correct, so it does not eliminate the developer's legal and moral obligations. Therefore, extreme care should be taken in the development of embedded systems to make sure that the right amount of time is spent on V & V, and also that certification not be used to prove that a system is correct.

Verification Techniques
There are many different verification techniques but they all basically fall into 2 major categories - dynamic testing and static testing.

• **Dynamic testing** - Testing that involves the execution of a system or component. Basically, a number of test cases are chosen, where each test case consists of test data. These input test cases are used to determine output test results. Dynamic testing can be further divided into three categories - functional testing, structural testing, and random testing.

• **Functional testing** - Testing that involves identifying and testing all the functions of the system as defined within the requirements. This form of testing is an example of black-box testing since it involves no knowledge of the implementation of the system.

• **Structural testing** - Testing that has full knowledge of the implementation of the system and is an example of white-box testing. It uses the information from the internal structure of a system to devise tests to check the operation of individual components. Functional and structural testing both chooses test cases that investigate a particular characteristic of the system.

• **Random testing** - Testing that freely chooses test cases among the set of all possible test cases. The use of randomly determined inputs can detect faults that go undetected by other systematic testing techniques. Exhaustive testing, where the input test cases consists of every possible set of input values, is a form of random testing. Although exhaustive testing performed at every stage in the life cycle results in a complete verification of the system, it is realistically impossible to accomplish. [Andriole86]

• **Static testing** - Testing that does not involve the operation of the system or component. Some of these techniques are performed manually while others are automated. Static testing can be further divided into 2 categories - techniques that analyze consistency and techniques that measure some program property.

• **Consistency techniques** - Techniques that are used to insure program properties such as correct syntax, correct parameter matching between procedures, correct typing, and correct requirements and specifications translation. [Andriole86]

Validation Techniques
There are also numerous validation techniques, including formal methods, fault injection, and dependability analysis. Validation usually takes place at the end of the development cycle, and looks at the complete system as opposed to verification, which focuses on smaller sub-systems.

• **Formal methods** - Formal methods is not only a verification technique but also a validation technique. Formal methods mean the use of mathematical and logical techniques to express, investigate, and analyze the specification, design, documentation, and behavior of both hardware and software.

• **Fault injection** - Fault injection is the intentional activation of faults by either hardware or software means to observe the system operation under fault conditions.
• **Hardware fault injection** - Can also be called physical fault injection because we are actually injecting faults into the physical hardware.

• **Software fault injection** - Errors are injected into the memory of the computer by software techniques. Software fault injection is basically a simulation of hardware fault injection.

• **Dependability analysis** - Dependability analysis involves identifying hazards and then proposing methods that reduces the risk of the hazard occurring.

• **Hazard analysis** - Involves using guidelines to identify hazards, their root causes, and possible countermeasures.

• **Risk analysis** - Takes hazard analysis further by identifying the possible consequences of each hazard and their probability of occurring. [Kopetz97]

The IEEE Standard for Software Verification and Validation (IEEE Std 1012-1998) contains information on software integrity levels, the V & V process, the Software V & V reporting, administrative, and documentation requirements, and an outline of the software verification and validation plan.

**CONCLUSION**
An explanation is given of software verification and validation (V&V) and how it fits in the development life cycle. How to apply V&V is also discussed. Evaluations of its effectiveness are summarized

**FUTURE SCOPE**
Verification and validation are part of the long certification process for any embedded system. There are different reasons why a product needs certification. Sometimes certification is required for legal reasons. For example, before an aircraft is allowed to fly, it must obtain a license. Being certified would also be important for commercial reasons like having a sales advantage. One of the main reasons for certification is to show competence in specific areas. Certifications are usually carried out by government agencies or other organizations with a national standing.

**REFERENCES**

