AN APPROACH TO THE TESTING OF EMBEDDED SYSTEMS

Bart Broekman and Edwin Notenboom of Sogeti in The Netherlands have been involved since February 1999 in a European ITEA project on embedded systems. In this article they show the need to have a flexible approach and advises a method that can be used to fit the test to the product.

Embedded systems have spawned an industry that is huge and still growing. Software is becoming an increasingly important part of embedded systems and often the latest software technologies are applied, such as real-time Java and embedded Internet. The embedded systems that are being developed continuously increase in size, complexity and the level of integration with other products. This also increases the need for structured testing of such systems. So what does “the testing approach for embedded systems” look like?

Obviously the testing of mobile phones will be significantly different from testing video set top boxes or cruise control in cars. They each require specific measures in the test approach to cover specific issues of that system. There is therefore no point in trying to develop “the ONE test approach for embedded systems”, but instead we should focus on developing a *method* which assists in "assembling" the suitable test approach.

Fig. 1 illustrates the model called TEmb of such a method, which basically works as follows: In the initial stages of the project the system- and project characteristics are analysed. They are high level characteristics such as "technical-scientific algorithms", "mixed signals", "multiple-V development", "safety critical", etc. Just a limited set of characteristics is sufficient to "classify" the embedded system. For each characteristic specific measures (solutions) exist that apply to the specific situation and can be included in the test approach. Through this mechanism the suitable dedicated test approach can be assembled from "generic elements of any test project" and a set of relevant "specific solutions" related to the observed system characteristics. This will be explained in more detail in the rest of this article.

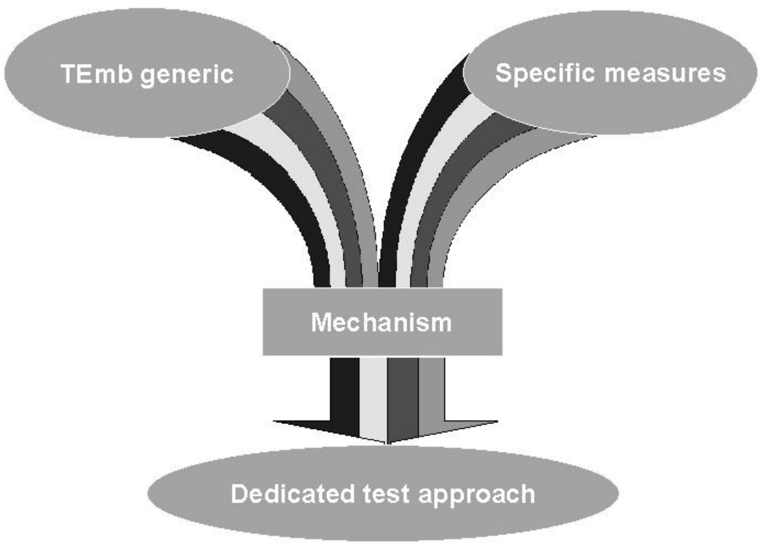

Fig.1 The TEmb model for assembling a dedicated test approach

Although many reasons exist why different embedded systems must be tested quite differently, there are also many similar problems and similar solutions that are part of any test approach. For instance: planning the test project according a certain life cycle, applying standardised techniques, dedicated test environments, organising test teams, formal reporting, etcetera. They are the generic elements, covering the four cornerstones of structured testing: Life cycle; Infrastructure; Techniques; Organisation. It can be considered the basis of any test approach.

For a particular embedded system, this basis test approach must be furnished with several specific measures to tackle the specific problems of testing this particular system. In the model this is called the “specific solutions” layer. A few examples of such specific solutions are:

- The British standard MOD-00-56 defines a specific life cycle for safety analysis activities.
- Dedicated tools exist for performing so called “threat detection”. They do not find “hard errors”, but they search for condtions in the input domain which *may cause* the software to fail.
- Specific test design techniques can be applied to test the state based behaviour of the system.
- In some development environments the system can first be modelled and then this model can be dynamically tested. This enables for instance to test the real time behaviour of the system *before* any code has been produced!
- “Evolutionary algorithms” (or “genetic algorithms”) is a special test design technique where the test cases are not derived from the system documentation but from the way the test cases themselves behave. It is an optimisation process which causes test cases to “evolve” into “better test cases”. It is based on the biological principles of “survival of the fittest”.
- When testing requires several expensive simulators, a specific controlling department can be installed to maintain and manage them. This is especially useful when these tools must be shared by many different teams at various overlapping stages.

The crux of the TEmb model for “assembling the suitable test approach” is the “mechanism”: How to find the specific solutions that apply to your specific embedded systems test project? The mechanism is based on the analysis of system-and project characteristics and will be explained with the following example:

Consider the following ten embedded systems:
- set top box
- cruise control
- weather forecast
- wafer stepper
- pace maker
- NMR scan
- infra-red thermometer
- rail road signalling
- telecom switches
- goal keeper (missile defense system)

They are very different systems indeed and it is natural to assume that the test approaches for those systems will be very different as well. But they don’t have to be *completely* different. What is it that makes each system so special to test? Is there a commonality in what makes them unique? For the sake of this example, look at the following two characteristics that may or may not apply to each system:

- *Safety critical*: Failure of the system can result in physical harm to a human being.
- *Technical-scientific algorithms*: The processing consists of complex calculations, such as solving differential equations or calculating the trajectory of a missile.

Now assess for each of the ten systems whether the two characteristics apply or not. This divides the ten systems into four groups, as shown in fig. 2.

Technical scientific algorithms \ Safety critical	***NO***	***YES***
NO	set top box; wafer steppers; telecom switches	rail road signalling; pace maker
YES	weather forecast; IR thermometer	cruise control; NMR scanner; goal keeper

Fig. 2 "What makes the system special" classified using system characteristics

The systems that belong to the same group possess similar qualities that can be tackled in a similar way. For instance, both the test approaches for rail road signalling and the pace maker will contain specific measures (or solutions) to address the safety critical nature of the system and will not bother to tackle technical scientific algorithms. Because a link is provided between the system characteristics and the suitable specific solutions, assembling a suitable test approach almost becomes an easy task. For the two characteristics of this example the following set of specific solutions can be proposed:

1. For *safety critical* systems the life cycle MOD-00-56 can be applied and a dedicated test level is defined to execute safety tests. The new roles of "safety manager" and "safety engineer" appear in the test process. Several techniques are proposed: Failure Mode Analysis (FMA), Fault Tree Analysis (FTA), model checking and formal proofing.

2. In order to tackle the *technical scientific algorithms*, several techniques can be helpful in assuring that the complex processing flow is sufficiently covered: evolutionary algorithms and threat detection. This also requires tools for coverage analysis and threat detection. An activity is defined to explicitly validate the used algorithm, early in the test (or development) process. In this specialised field, testing needs support from mathematical expertise.

This is illustrated in fig. 3 which also shows to which cornerstone (life cycle; infrastructure; techniques; organisation) the specific solution belongs.

system characteristic	Life Cycle	Techniques	Infrastructure	Organisation
safety critical	Master test plan, incl. MOD 00-56 Safety execution test (test level)	FMA / FTA Model checking Formal proof		Safety manager Safety engineer
technical scientific algorithms	Algorithm validation	Evolutionary algorithms Threat detection	Coverage analysers Threat detectors	Math. expertise

Fig. 3 For each system characteristic a set of specific solutions (related to the 4 cornerstones)

Other useful system characteristics, besides the two already mentioned in the example, are: multiple-V development (model; prototype; final product); state based behaviour; mixed signals; autonomous systems. The list of system characteristics and the related specific solutions is yet far from complete and I guess that completeness cannot be achieved at all. But I prefer to be pragmatic: just start with a useful set and let it evolve.

The advantage of focusing on system characteristics is that just a limited set is required to cover an enormous diversity in systems. The "uniqueness" of each embedded system test project is made more concrete and manageable by analysing which system characteristics are relevant. Attachment of specific solutions to all of the system characteristics then provides an easy transition to what is needed in your dedicated test approach.

The mechanism which is explained here can be seen as an extension to the risk based test strategy: The analysis of perceived risks (or usually "the agreement on", because it is rather subjective) leads to choices of what to do (and what *not* to do) in the test approach. The analysis of system characteristics (which is more objective in nature) leads to proposed solutions which can be adapted in the test approach to tackle the specific issues of your specific embedded systems test project.