BOUNDARY CONDITIONS FOR OPENETCS DEVELOPMENTS
– POSITION PAPER ON THE PREREQUISITES FOR CERTIFIABILITY –

JAN PELESKA AND CÉCILE BRAUNSTEIN
UNIVERSITY OF BREMEN
{JP,CECILE}@INFORMATIK.UNI-BREMEN.DE

Abstract. The main objective of the openETCS project, as it is currently discussed among the project members, is to develop models and software for the European Vital Computer (EVC) without fixing a dedicated hardware platform, perhaps not even a dedicated operating system. In this position paper the authors argue that certifiability of software cannot be achieved without at least specifying the capabilities and boundary conditions to be fulfilled by the underlying operating system and hardware architecture. Instead of selecting a concrete operating system (OS) and a concrete EVC hardware (HW) platform – this would certainly not be advisable in the current phase of the project – we propose to specify the boundary conditions for both OS and HW. Certification – or, at least, adequateness for later certification – may then be achieved under the prerequisite that the OS and HW to be selected for the EVC fulfils the conditions specified.

1. Why Software is not Certifiable Without OS and HW Specifications

When developing (or automatically generating) software from an openETCS model it is mandatory to prove (or at least justify by comprehensive testing) that the actual behaviour of the software, when executed, will be consistent with the behavioural semantics of the model, that is, with the *intended* behaviour captured in the model. Behaviour is observable in

- the data domain (are inputs correctly transformed into outputs?),
- the time domain (how much time passes between an input to the EVC and its required reaction?),
- the causal domain (do these observation occur in the intended order?).

All of these behavioural aspects depend on both the OS and the underlying HW:

- Data transformations fail if, for example, the word length of the HW registers is inappropriate for the calculations involved.
- Expected reactions miss their deadlines if, for example, the OS scheduler does not allocate appropriate amounts of CPU time to the task processing the input.
- Events occur in the wrong order if the OS does not provide adequate mechanisms for critical section management.

As a consequence an application software layer like the openETCS EVC control software can only be validated and certified in relation to certain OS and HW capabilities.

Date: October 12, 2012.

We therefore expect that the EVC code generated from the openETCS models will reference the interface of some OS application program interface (API) providing the services necessary to ensure the proper behaviour of the software to be executed. Moreover, a list of hardware capabilities to be fulfilled has to be specified (e.g., is it mandatory to run the software on a multi-core platform?). The OS and HW capabilities specification should be part of the openETCS-project process as shown in figure **??**.

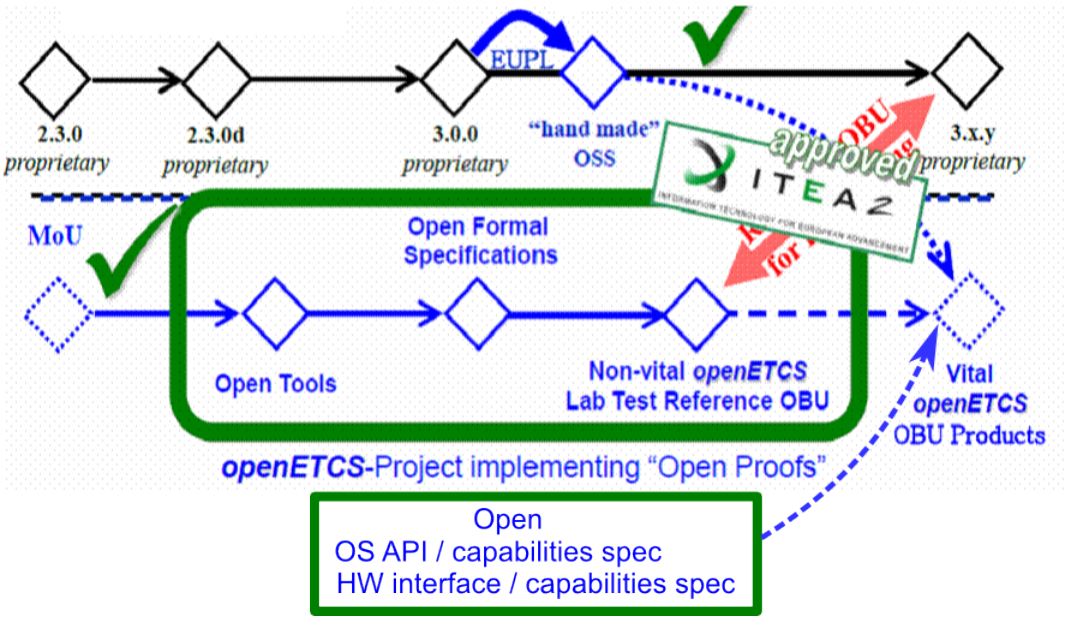

FIGURE 1. OpenETCS-Project process

2. Suggestion About How to Proceed

We propose to elaborate specification documents about OS and HW capabilities required as part of WP2 and WP3. The OS specification document should specify

- the API required by the openETCS application,
- the scheduling capabilities (e.g., reaction to interrupts, partitioning in the time domain [i.e., no process may prevent another one from getting the CPU with the intended period and duration]),
- the communication interfaces supported,
- validated compiler/linker to be used for generating executable code for the target HW.

Instead of specifying OS properties from scratch we suggest to adopt the well known OS specification from the ARINC 653 standard[1] for which OS have been developed and are applied successfully for safety-critical tasks in the avionic domain.

The HW specification should include, for example,

[1]Avionics Application Software Interface, Part 1, Required Services. AERONAUTICAL RADIO, INC., 2551 Riva Road, Annapolis, Maryland 21401-7435, (2005).

- the required HW interfaces and their performance characteristics,
- the CPU architecture (number of cores, word length etc.),
- partitioning requirements (do we need a memory management unit to prevent processes from corrupting each others' memory state?),
- existence of real time clocks.

With this information at hand, certification credit in the form

Provided that operating system and hardware platform fulfil the boundary conditions specified, the openETCS code is suitable to run on such a platform without alterations.

Observe, however, that this certification credit does not render the verification and validation of the proper HW/SW integration superfluous. It just saves time in the way that the *logical* correctness of the algorithms and software interfaces involved have been verified before and do not have to be be re-verified completely during HW/SW integration.