Innovative Approach for Requirements Verification of Closed Systems

Jose Reis, Brett Bicknell, Michael Butler, John Colley

Innovative Approach for Requirements Verification of Closed Systems

Jose Reis, Critical Software Technologies

Brett Bicknell, Critical Software Technologies

Michael Butler, University of Southampton

John Colley, University of Southampton

Keywords: Closed Systems, COTS, Event-B, Rodin, formal methods, integration, verification, model checking, simulation, safety case, assurance

Requirements quality has long been an issue in safety-critical development. Programme pressures and lack of understanding of user needs, combined with inadequate methods and tools, leads to incomplete and inconsistent requirements specification. This often results in poor-quality software and significant overspending on programmes.

Requirements are generally verified early in the lifecycle using manual inspections. However, while the manual approach is valid for less complex systems, it is not adequate for systems with greater complexity and a complicated hierarchy of requirements. Another trend within UK Defence is the adoption of Closed Systems for mission- and safety-critical systems. As the name suggests, a Closed System is a solution which is available as a black box, where access to internal modules and design artefacts is not possible. In the context of this paper, we will refer to a Closed System as a software-based system which can refer to either of the following contexts:

1. Systems developed by a third party that are commercially available to the public domain as readymade solutions; also known as COTS solutions.

2. Systems developed by a third party that are not commercially available but, due to legal, export controls or other commercial reasons, are not open to the system integrator.

The use of Closed Systems raises several challenges from a safety certification perspective: limited information, limited test coverage and non-deterministic behaviour of the Closed System in use. When the system under test is closed, a classic isolation testing approach is inadequate to demonstrate the reliability of safety-critical software, because proper coverage cannot be measured. Lacking proper coverage during the testing phase, the scenarios used may only test certain functions within the Closed System software. When the software is then used, functions that have not been tested may generate faults that could cause the system to behave in unexpected ways.

A system architecture based on Closed Systems is particularly sensitive to obsolescence management issues. When a component (software, hardware or firmware) is upgraded, the system needs to go through regression testing. The key issues are to determine which tests need to be run to ensure that the integrated system

operates correctly with the new part, the impact of the new version of the software/hardware on the rest of the system and how to validate assumptions made about the new version of the software/hardware.

Verification and Integration of Closed Systems through Formal Methods (VICS-FM), a new approach developed by Critical Software Technologies and the University of Southampton, uses a rigorous approach to extend the evidence provided by software verification processes[1]. The approach involves generating formal models to help provide further evidence relating to the behaviour of architectures with integrated Closed Systems, before having to run the Closed Systems in a test environment. This saves time and costs, increases confidence in the system and allows for any proposed updates to the Closed System to be assessed before they take place.

The additional evidence provided by the formal modelling includes:

- The ability to precisely define functional coverage and to measure the functional coverage of test sets.
- The generation of test sets using the tools available in the formal approach.
- Strong validation of models using proof and simulation tools, which validate the definition of functional behaviour and the functional coverage of tests.
- The ability to support the evaluation and impact assessment of changes when a new version of the software/hardware is released.

The VICS-FM approach offers a rigorous method to:

- Strengthen the requirements and design of the system.
- Validate assumptions about the solution.
- Conduct ‘what-if’ analysis.
- Identify which scenarios, if any, lead to the system being left in an unforeseen or undesired state, such as deadlock.
- Generate an animated model and visualisations of the system using a graphical user-friendly interface, to help test and validate the formal models.

The VICS-FM approach is based on the use of a formal modelling language (Event-B) which is based on set theory and supported by mathematical proof and model checking technology. Due to the formal method used, the responses provided by VICS-FM are unambiguous and trustworthy. VICS-FM is best used at earlier phases of the lifecycle, because it can identify problems with the requirements and can be used to confirm the suitability of the Closed System. The approach can also be used once the Closed System has been procured and installed.

VICS-FM complements more traditional verification & validation approaches. In particular VICS-FM can be used to verify elements on the left-hand side of the V-Model. System, integration and acceptance tests are still required, but their scope can be reduced. When the VICS-FM approach is used with bespoke applications, unit tests are not normally required.

INNOVATIVE APPROACH FOR REQUIREMENTS VERIFICATION OF CLOSED SYSTEMS

Innovation

The VICS-FM approach is described below in the context of Closed Systems verification but it could also be applied to a bespoke system.

The innovation in VICS-FM stems from the application of an existing method (Event-B) plus a toolset (Rodin) to solving the problem of how to verify and integrate Closed Systems. The model of the Closed System is sufficiently generic to be reusable in different contexts. The specific VICS-FM innovation did not involve any significant developments to the Rodin toolset.

The approach proposed represents a considerable change in the use and successful integration of Closed Systems, using formal methods to guarantee their integration and functionality.

The VICS-FM approach could be used in:

- Verification and Validation of complex systems that use Commercial Off-The-Shelf (COTS) solutions.
- Verification of safety properties in mixed risk environments where COTS solutions are used by the underpinning platform. Typically in these environments, single failure scenarios can be verified through classical testing approaches but multiple failure scenarios are more difficult to verify. The VICS-FM approach will increase the assurance of these systems.
- Supporting the integration of legacy systems with COTS solutions (for example, virtualisation middleware). Legacy systems do not always integrate directly with COTS solutions, therefore bespoke solutions are required. These solutions can be modelled and proven correct using this approach.
- Requirements and design specification: VICS-FM can be used to verify properties and highlight gaps in requirements and design early in the development process.

Solution

The VICS-FM approach requires that the input specifications are translated to an Event-B model. This is done manually, using the Rodin toolset[2]. The Event-B models are constructed of variables, events – which perform actions on variables – and invariants, which allow the designer to specify the properties of the system that should be upheld. The proving and model checking capabilities of Rodin help indicate whether these invariants, and hence the associated properties, are maintained over the possible behaviour of the system that is encapsulated by the modelled events. The approach not only provides an indication of the changes that need to made in the case of the violation of any of the properties, but it also provides examples – i.e. particular sequences of events – where the properties are violated; from which the associated scenarios can be identified to help inform areas of uncertainty in the system behaviour as soon as possible.

A case study used to evaluate the VICS-FM approach involved the integration of a closed COTS solution within an end-user system with particular safety and critical operational constraints. The COTS solution is a virtualisation middleware (VMWare) and the end-user system is a military system (mission critical) which, due to the security classification level, cannot be disclosed in this paper. Issues were found with standard testing approaches due to some inherent, non-deterministic behaviour of the COTS solution. In an attempt to combat this, some additional bespoke software was created to constrain the behaviour of the COTS solution, although it was not fully understood how the combination of the COTS solution with the end-user system and bespoke

software element affected the safety and operational constraints; partly due to the closed nature of the COTS solution, but also because of the complexities of the interaction. The VICS-FM approach was used to assist in the verification of the behaviour of the composed system in terms of the identified constraints. The Event-B model produced consisted of several parts:

1. A model of the relevant behaviour of the COTS solution, based on documentation of the COTS solution and discussion with domain experts. It was not a requirement to model the entire COTS solution, only the subset of the behaviour which was relevant to the properties to be verified. As mentioned in the limitations later in the paper, it must be ensured that the assumptions and exclusions made during this process are recorded and agreed with the safety and engineering teams. As the verification involved the non-deterministic behaviour of the COTS solution, this was reflected by considering all possible combinations and sequences of the defined events for the model of the COTS solution in an abstract way. This meant that, not only was it possible to construct a suitable model of the COTS solution without requiring in-depth knowledge of the exact process sequences followed in different scenarios, but also that during model checking, any of the potential paths of the COTS solution could be encountered.

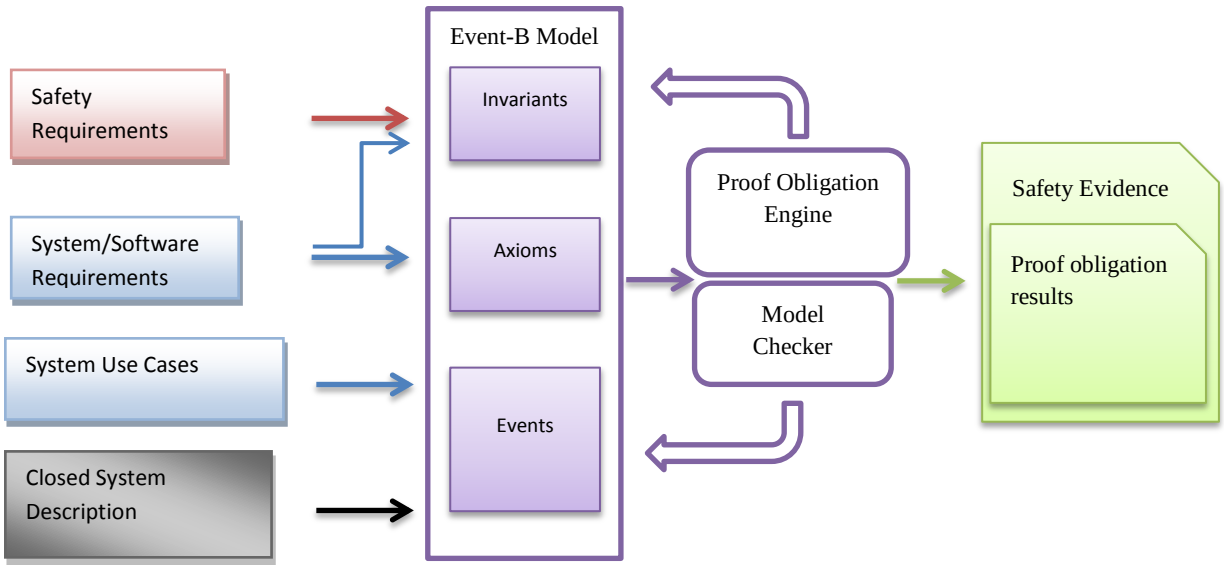

Figure 1: VICS-FM Approach

2. A model of the applicable elements and available actions of the end user system, including the bespoke software mentioned above, and how these interact with the COTS solution.

3. Global properties of the system, specified as invariants, corresponding to the constraints that require verification. In the case study identified above, the invariants corresponded to the properties of the system that the behaviour of the COTS solution should maintain due to the limits imposed by the bespoke software solution.

To facilitate the generation of the models, especially for users who are less experienced in the approach, UML-like diagrams and state machines can be generated in Rodin, from which the tool automatically generates the underlying Event-B model. In a simulation of the model, the user can step through the transitions (depicted in the example state machine, Figure 2 below) in an interactive fashion, whilst the Rodin toolset checks whether any of the transitions causes an invariant violation and flags them for the user. The UML-B tool in Rodin[3],

which provides the capability to produce and translate these diagrams, was used to generate the models of the COTS solution and end-user system mentioned above during the case study.

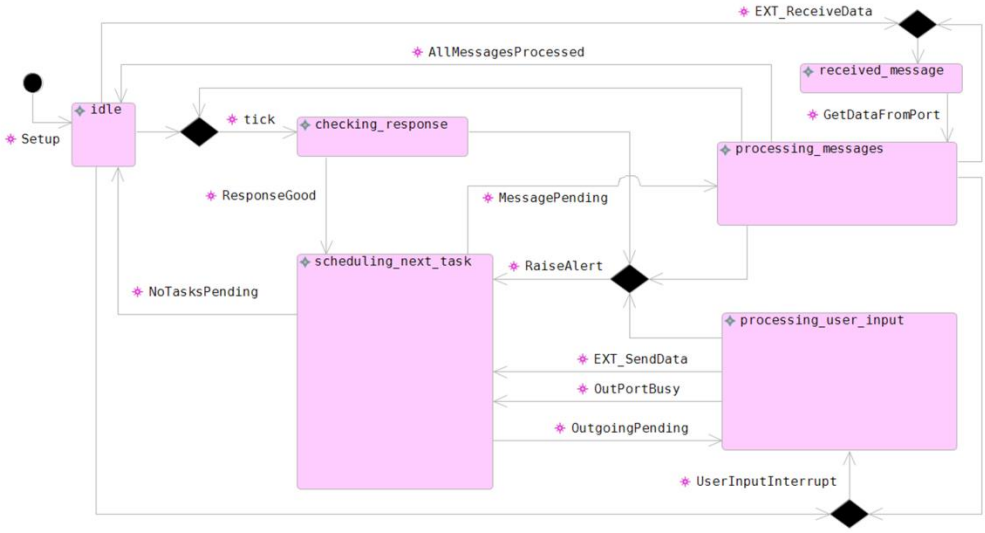

Figure 2: Example of Event-B State Machine

After generating the Event-B model of the system the following actions can be conducted automatically using the Rodin toolset:

1. A model consistency check, run using simulation and model checking (using the ProB tool[4]).

2. Proof obligations are generated which, when proven, can demonstrate whether any of the defined actions invalidate any of the invariants regarding the variables being modified. In other words, proof obligations are generated to demonstrate that all possible outcomes of each action maintain the properties specified by the invariants. There are automatic proving capabilities within Rodin which assist in completing these proof obligations.

3. The model checker can be run to find scenarios where particular invariants are violated, if the proof obligations cannot easily be discharged. This is useful for more complex behaviour and invariants, such as those encountered during the case study. These scenarios can be identified and typically correspond to areas of uncertainty, where it is important to have feedback as soon as possible. During the case study mentioned above, a scenario was highlighted which violated some of the safety and operational constraints. By examining the trace of events corresponding to this scenario – provided by the model checker – it was possible to understand which limitation or specific behaviour of the bespoke software caused the violation of the constraint. Figure 3 below gives an example of the use of the model checker; the violated invariant is shown in red and the trace of events that led up to the violation is displayed in the history tab on the right-hand side.

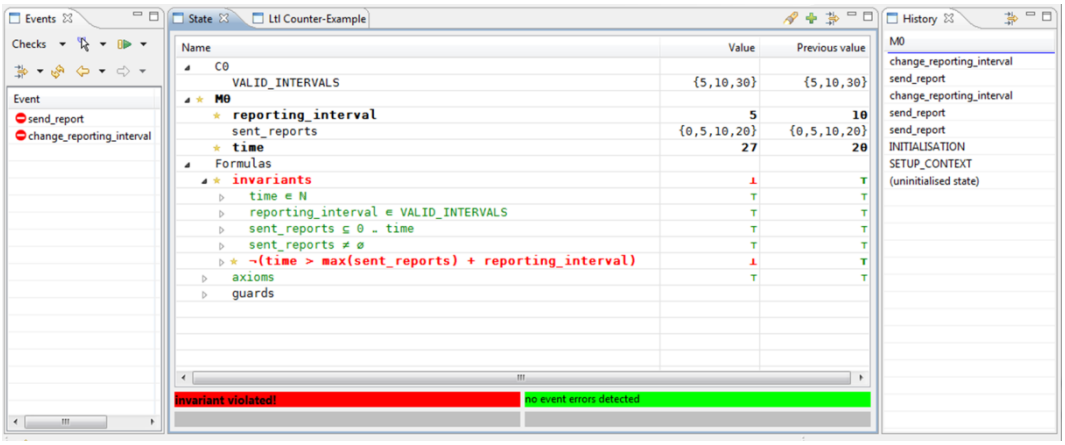

Figure 3: Example of the model checker in use

4. The Event-B model can be animated within Rodin using the BMotion Studio tool, which is part of ProB. Using the tool, it is possible to generate an animated front-end to the simulation of the model in the form of a user-friendly graphical interface, which corresponds to the system's GUI (see Figure 4). The user can interact directly with this animated front-end, while the tool continues to run the formal analysis in the background, reacting to user choices and checking the model and invariants at each step. This was utilised during the case study, to produce a representation of the GIU provided as part of the end-user system. As this graphical representation is tied to the underlying Event-B model, it can not only be used to run through the model and confirm that the model is the correct representation of the system, but can also be used to explore further scenarios. This graphical representation of the system can be used without necessarily requiring any experience with the Event-B language or the toolset.

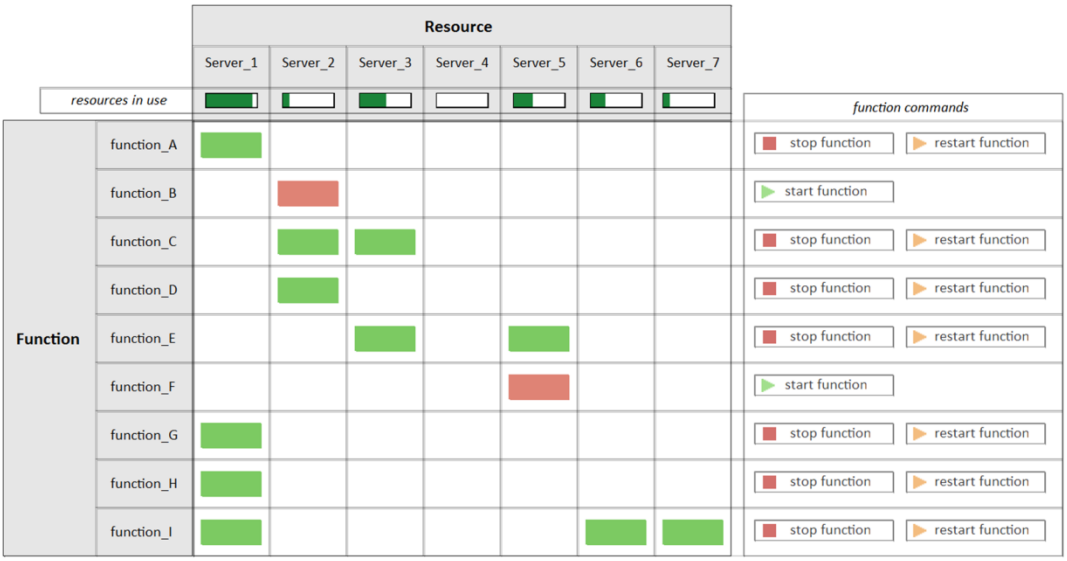

Figure 4: Example of a BMotion Studio visualisation

Benefits

The VICS-FM method lends itself to finding inconsistencies in the requirements specification. The set theory that supports the method and the proof obligation mechanism forces the engineer to think further and to specify properties that are implicit, but not necessarily specified in the requirements, as inputs. Any inconsistencies between the requirements are identified and fed back to the requirements owner.

Another useful mechanism offered by the Event-B method is refinement. This allows the model to start in a very abstract form, and gradually be refined by adding further properties and detail at each step, whilst ensuring that consistency is maintained with all previous refinement levels of the model. In the context of Verification and Validation (V&V), this mechanism enables the tailoring of the Event-B model to include only that which is strictly required to prove specific properties. This mechanism is critical in the context of Independent Software Verification and Validation (ISVV) activities, because the time available for conducting an ISVV activity is often limited and therefore it is not possible to translate the entire system into an Event-B model.

Another advantage of this approach is that is promotes model reuse. The model of the Closed System can be considered separate and therefore reused and applied to validate the integration of other systems that interface with the same Closed System. In addition, if further properties need to be specified for the system in the Event-B model, this can be factored into the process by refining the existing system model and adding the relevant extra behaviour in this additional refinement. Adding extra properties (for instance, an extra requirement) does not necessarily involve modifying the existing model, or having to re-prove any of the existing proof obligations, as these extra properties are introduced and tested in an additional refinement level. This adds flexibility as to when the formal verification is used: the main model can easily be picked up at later stages should the customer be unsure about a particular, perhaps newly-developed, property.

The project in which the VICS-FM Approach was used had a duration of 5 months and for that reason the chosen case study and the Event-B model generated had to be tailored. Considering these constraints it was possible to identify one scenario where the real system would be left in an undesirable state. This finding was achieved with an effort of approximately 3 man-months, which includes analysis of the requirements and discussion, generation of the Event-B models and “execution” of models. The provider of the case study came to the same conclusion using a classical testing approach but with an effort of approximately 12 man-months. The reason for the extra effort using the classical testing approach is mainly because of the non-deterministic nature of the virtualisation platform and the use of the manual approach in testing. Using the VICS-FM approach the identification of scenarios where the system could be left in an undesirable state requires less time and effort. In terms of scalability, it is known that the underlying toolset has been used with models of considerable size [5, 6, 7]. The key issue when the size of the models increases relates to proof generation, which is not fully automated and sometimes requires manual intervention to discharge. This is not an issue with the VICS-FM approach but rather an issue with the Rodin toolset and the theorem provers used.

In summary, the VICS-FM approach:

- Provides a sound model of the requirements selected for modelling, so that any ambiguity within these can be removed.
- Improves the integration of Closed Systems, supporting the relevant safety cases and reducing costs throughout the V&V cycle.
- Enables precise definition of functional coverage.

INNOVATIVE APPROACH FOR REQUIREMENTS VERIFICATION OF CLOSED SYSTEMS

- Enables the measurement of the functional coverage of test sets.
- Strengthens the requirements and design of the system.
- Validates assumptions about the solution and facilitates 'what-if' analysis, in particular around safety and availability requirements.
- Identifies which scenarios, if any, lead to the system being left in an unforeseen or undesired state, such as deadlock.
- Includes strong validation of the models using proof and simulation tools, which validates the definition of functional behaviour and the functional coverage of tests.
- Allows for the formal model of the Closed System(s) to be reused in different industrial contexts.
- Is flexible in facilitating changes to the requirements and software versions due to the inherent refinement mechanism.
- Indicates if these changes affect previously defined properties.

Limitations

1. The resulting Event-B model is only a representation of a subset of the end system and Closed System. To address any potential shortfalls of the Event-B Model, a strategy has to be identified and agreed with the safety and systems engineering team with regard to which properties need to be formally verified. This will then define which aspects of the Closed System are modelled and which elements of the rest of the system remain to be modelled. The level of confidence in the evidence collected using other approaches (classic testing), plus scenarios which are hard to verify or simply cannot be verified by running the system over longer and longer timescales, should be taken into account to decide what will be formally modelled using Event-B.
2. Due to the fact the Event-B model generated covers only a subset of the real system, the outputs of the proofs and simulations run are limited to the elements modelled in Event-B. Depending on the issues that need to be verified, this could become a limiting contingent on the expected outcomes of the verification activity and the dependencies within the inner elements of the Closed System. Using abstractions, it is possible to focus on and expand higher-level elements of the system, if that makes the model more representative.
3. Any assumptions about the underlying behaviour of the Closed System must be documented and discussed with the safety and engineering team to ensure that they are justified.
4. The Event-B method and the Rodin toolset are not yet capable of verifying certain performance requirements such as confirming the time it takes to perform particular tasks. Therefore, the required availability of certain resources used by the system must be ascertained using a different approach.
5. VICS-FM is fundamentally geared towards the verification of safety-critical software. It is a sound approach that provides rigorous evidence to support the safety case. Nevertheless, the approach still requires specialist knowledge in set theory and the Event-B method, hence it is not straightforward to learn and must be practised by an experienced engineer.

References

1. B. Bicknell, J. Reis, M. Butler, J. Colley, C. Snook, A Practical Approach for Closed Systems Formal Verification Using Event-B, *10th International Conference, SEFM 2012*, pp. 323-332.
2. J.R. Abrial, M. Butler, S. Hallerstede, T.S. Hoang, F. Mehta, and L. Voisin. Rodin: An Open Toolset for Modelling and Reasoning in Event-B. *International Journal on Software Tools for Technology Transfer (STTT)*, 12(6):447–466, 2010.
3. Colin Snook and Michael Butler. UML-B and Event-B: An Integration of Languages and Tools. In *The IASTED International Conference on Software Engineering - SE2008*, February 2008. Event Dates: 12-14th February 2008.
4. M. Leuschel and M. Butler. ProB: An Automated Analysis Toolset for the B Method. *International Journal on Software Tools for Technology Transfer (STTT)*, 10(2):185–203, 2008.
5. F. Yuan, S. Wright, K. Eder, D. May. Managing Complexity through Abstraction: A Refinement-Based Approach to Formalize Instruction Set Architectures, *ICFEM 2011*, pp 585-600.
6. Satpathy, Manoranjan, Snook, Colin, Arora, Silky, Ramesh, S and Butler, Michael, Systematic Development of Control Designs via Formal Refinement, *International Conference on Model-Driven Engineering and Software Development,* 2013.
7. Alexander Romanovsky, Martyn Thomas, *Industrial Deployment of System Engineering Methods,* 2013 Springer Berlin Heidelberg.

The technology underlying the VICS-FM approach continues to be developed as part of the FP7 ADVANCE Project.