Methods for Enhanced Safety Wrapper Design

Luping Chen, John May

Safety Systems Research Centre, Faculty of Engineering,
University of Bristol, Bristol, BS8 1UB, UK
L.Chen@bristol.ac.uk, J.May@bristol.ac.uk

Abstract: A key requirement for safety-critical systems that use Commercial Off-The-Shelf (COTS) components is that the COTS components do not compromise the reliability, safety and security of the overall system. Safety wrappers can be seen as a means to control the integration of the (often disparate in nature) components to ensure fault tolerance, quality of service and security. This paper discusses new strategies to enhance safety wrappers using a nuclear protection system example. It also illustrates a Perturbation of Interface Parameters (PIP) technique to aid and assess the new designs.

1 Introduction

Component-based software engineering (CBSE) offers the potential of economic and speedy system design and production. At a coarse grain this can be based on the reuse and integration of high-level COTS components and bespoke components to form a new system[BW98]. A key issue for such hybrid systems is to show that the use of COTS components (which will be considered as 'black boxes') does not compromise the safety, reliability and (perhaps) security of the overall system, since the reliability of COTS software components cannot be fully assured prior to integration [CL02]. Furthermore, even if such pre-assurance was a theoretical possibility, it would seldom be available, since COTS components are commonly developed to unknown standards or standards aimed at general use, which are insufficient for safety applications [Pr96]. These difficulties are sometimes compounded by the inaccessibility of some COTS code. Where examination of code is not permitted, traditional assurance techniques (with the exception of black box testing) cannot be applied to COTS components post-purchase, to supplement the supplier's verification and validation (V&V) activities. In general it is necessary to use 'middleware', possibly based on standard infrastructure technologies, to integrate disparate components [Ma02]. This middleware offers an important opportunity to include component adaptation and monitoring strategies, to help ensure fault tolerance, quality of service and security. The overall integration process can be complex, perhaps involving both syntactic and functional adaptation but this study will have the simple focus of assessing the value of safety wrappers (diagnostics in the middleware) to prevent the propagation of faults or unexpected behaviour from the COTS component to the rest of the system. It will be assumed that the system will support an appropriate recovery strategy, or selection of a safe state, following fault detection. We will consider the particular case of integrating a COTS component of uncertain integrity into a bespoke system of accepted safety integrity level (justified by the design, development and V&V processes). The objective of our research was to focus on how to protect the bespoke system from failure due to COTS faults/uncertainties.

2 Measuring safety wrapper performance using PIP

Some earlier works have developed the use of fault injection to test the diagnostic fault coverage provided by safety wrappers [Vo98]. The proportion of faults in the COTS component detected by a wrapper can be used as a metric to evaluate if a wrapper design fulfils its safety specification, and if a new wrapper strategy can offer further improvements. One approach in [NM00] investigated the diagnostic value of executable assertions using Software Fault Injection (SFI) inside the COTS component to cover a range of fault sizes (footprint in the input space) and locations. However, when a COTS component is truly treated as a black box, the COTS code is inaccessible, and fault injection inside COTS components will not be possible. To try to overcome these limitations of traditional SFI, we developed a technique of Perturbation of Interface Parameters (PIP) of a COTS component to simulate faults contained in the COTS component [CHM02]

3. New safety wrapper designs

There are many different 'styles' of diagnostics that could be used in wrappers. For example, straightforward executable diagnostics checking the results of *normal* COTS inputs vs. diagnostics checking *special* inputs constructed specifically for diagnostic purposes, and diagnostics checking COTS input/output relations (including reverse computation diagnostics) vs. diagnostics checking conditions on output variables alone. The issue of the effectiveness of these different wrapper types is complex. It seems a reasonable conjecture that the different techniques will each have their strengths. One technique will probably not be sufficient to cover all failure types, and conversely, different techniques will overlap to some extent in their failure detecting abilities. Furthermore, different techniques will result in different levels of diversity (lack of coincident failures) with respect to the original COTS functions. The aim of the research programme reported in this paper is to find methods of assessing wrapper effectiveness, and use them to identify powerful (highly effective) wrapper designs. Furthermore, effectiveness is not the sole criterion of interest. It is also important that wrappers are practical in the sense that they can be built with reasonable effort (and therefore will be used in practice). The following are key aspects of the approach:

- Assessment of high wrapper coverage must consider all potentially critical failures (these might, for example, be based on failure modes and effects analysis (FMEA) of COTS components and the surrounding system).
- Wrapper design should use techniques that maximise the possibilities for diversity between COTS components and wrappers.
- *Efficient* wrapper design will use multiple diagnostic techniques, and balance the functions to maximise orthogonality (i.e. reduce overlapping) of their failure-trapping functions.

We investigate the opportunities for, and design implications of, optimising the above factors associated with wrappers for a real COTS component, and using PIP to estimate the effectiveness of the new designs.

3.1 Design for high fault coverage

Since the objective of full failure coverage is usually unachievable, good fault/failure coverage in a normal safety wrapper is usually sought by designing against all potentially *critical* system failures based on a FMEA of a COTS component within its surrounding system. The specification of such a wrapper is therefore based on an understanding of what constitutes a critical violation of the requirement specification of the COTS component, and the normal test cases to verify and validate such wrappers are fundamentally based on this understanding.

If a latent fault in a COTS component has a failure format outside the chosen design specification for the safety wrapper, the specifically designed wrapper could not be expected to catch such failures from the COTS component. *There remain possibilities for a COTS component to introduce faults that are not predicted by the FMEA, and a corresponding test process based on that FMEA may miss such unpredicted failure modes.* It can be a non-trivial task to predict all possible system-critical COTS component failure modes. There is therefore a case for more arbitrary testing of a COTS wrapper, based on arbitrarily generated COTS failures, to see if the wrapper defends against those. Furthermore, in this case the effects of the propagation of undetected failures from the COTS to the wider system can be simply observed.

Our proposed technology for enhancing wrapper coverage is based on generation of a wide range of potential COTS component failures, by exploring the possible formats of anomalies in its interface parameters. Since the interface parameters as outputs of a COTS component to a bespoke system are the only route for the COTS component to display its failures, a complete set of the anomalies describes all possible failures of COTS component irrespective of their root cause. Thus, the difficulty of identifying a complete set of critical failures of a COTS component is shifted to the problem of producing a sufficiently rich set of simulated interface anomalies through perturbing the interface parameters. At the very least this is a valuable supplementary approach to V&V of a safety wrapper, which should probe it in a quite different way from traditional approaches.

3.2 Maximising diversity between wrappers and COTS components

As in any software, it must be assumed that both a wrapper and an associated COTS component will contain some faults, and reduction of common failures between them will improve system reliability. Diversity is an important feature of fault tolerant code; without it the original code and the added code will fail together and failures will not be trapped. In the design of normal multi-version (N-version) software, different structures in the two versions can produce diversity (structural diversity). However, for the development of safety wrappers, a COTS component can be a ‘black box’*, and without the internal details of a COTS component, it is difficult to assert diversity between a designed wrapper and a COTS component by lack of knowledge of original structural design*. One aid to designing diversity between a COTS component and its wrapper is that the latter only has to check rather than compute solutions which introduces a certain 'natural diversity' from the former. Therefore where possible, *the design of a safety wrapper could emphasize 'checking' rather than 'computing' data, to enhance diversity (functional diversity)* using simple logical condition checking as far as possible. An alternative would be to use safety wrappers based on reverse computation of the COTS code functions, to try to achieve function diversity.

3.3 Multiple diverse wrappers

In 3.2, we were forced to abandon structural diversity between a wrapper and a COTS component as a technique for achieving diversity. However, structural diversity can be used in a multi-version approach to wrapper design. A potential strength of a multi-wrapper approach is that the diversity/reliability improvement issues are judged on the basis of wrapper design and there is less emphasis on the COTS design (about which little may be known). Based on the new design strategies, the reliability of a system with a COTS component will depend on the failure coverage of all safety wrappers and the diversity between those wrappers. There is still a need for diversity between wrappers and the COTS component, which is a black-box factor and hard to judge, but it seems reasonable to expect that if wrapper A is not diverse with the COTS component and wrappers A and B are diverse, then wrapper B *will* be diverse with the COTS component (this assumption needs to be subjected to experimental or other examination). We have considered two kinds of diversity in new designs.

Structural diversity in wrappers

Previous results on structural diversity of multi-version software can be used directly to enhance the wrapper design [CMH02]. The central design idea is to build orthogonal functions in multiversions (in this case the wrappers). Thus wrappers are built with common requirement specifications but different software structures, and this can improve their diversity and hence their reliability as safety middleware.

Functional diversity in wrappers

There are various methods to implement functional diversity in design multiversion software, which can also be employed for safety wrappers. In our demonstration system, we built a 'checking style' wrapper for an original existing wrapper — a 'wrapper's wrapper.'

4 Observations and Conclusions

This paper mainly introduces strategies for high coverage/diversity wrapper designs. An empirical PIP test technique has been used to assess and help to identify new strategies. In this test technique, the proportion of faults in the COTS component detected by a wrapper can be used as a metric to evaluate if a wrapper design fulfils its safety specification, and if a new strategy can offer further improvements. Meanwhile, diversity has been demonstrated as another index relevant to a wrapper's tolerance to potential defects inside a COTS component. We tested three new strategies based on a nuclear protection system [QW91]. The experiments show that each strategy demonstrates prospects for enhancing the performance of safety wrappers. Our main observations were:

1. In the design for high fault coverage, PIP testing revealed a failure mode that was not considered in the previous design specifications of a wrapper, and the wrapper was redesigned to incorporate new assertions to defend against the failure mode, resulting in improved system safety(failure coverage was increased from 74% to 100% for the simulated fault set by PIP).
2. In designs for increased diversity between a wrapper and a COTS component, we used check-style and reverse functions in wrappers to increase their functional diversity. The failure probability of the whole system (under our simulated fault conditions) was reduced by 12% when the safety wrappers is used.
3. New multi-version safety wrapper designs, based on different software structures showed encouraging diverse failure behaviours under various faulty conditions. PIP tests conducted on the whole system incorporating two wrappers, demonstrated the safety improvement of the system against potential faults both in the COTS component and in individual safety wrappers. Since the wrappers are 'glass box', structural diversity can be used to enhance their design: our previous SSRC work on improving multi-version software diversity can be applied directly to multi-version wrappers [LC02b].

Our experiments on diversity designs did not distinguish functional and structural diversity: in terms of performance they were similar. But it was clear that check-style wrappers can be considerably more succinct than the code they check. This is not surprising; it is well known that checking a function can be a less complex task than computing it [Ha92] This effect was sometimes so pronounced that it was difficult to select plausible fault modes for injection into the check-style wrappers. We observed some points that could influence practice:

- Taking the view that a 'wrapper' can be built from multiple smaller complimentary wrappers can be very effective and easy to implement
- Functional diversity is easier to design than structural diversity in multi wrappers
- The application of check-style wrappers can reduce remained faults because they are usually simpler modules than other kinds of functional wrappers.

This is particularly useful in the context of black-box COTS components, and improves the fault tolerant ability of the whole system. COTS wrappers, in common with all forms of fault tolerance, have an intuitive worth based on normal engineering judgement. Use of PIP to help design and assess wrapper designs can be applied to any systems containing COTS components. The arbitrary nature of the fault injections can be seen as an advantage, supplementing and complementing the normal approach to wrapper design based on the COTS specifications and an FMEA in the surrounding system.

Acknowledgements

The work presented in this paper comprises aspects of a study (NewDISPO2) performed as part of the UK Nuclear Safety Research programme, funded and controlled by the CINIF together with elements from the SSRC Generic Research Programme funded by British Energy, Lloyd's Register, and the Health and Safety Executive.

References

[BW98] Brown, A.; Wallnau, K: The Current State of CBSE. In *IEEE Software*, 15(5), (Sept/Oct 1998); pp.37-46.

[CI02] Crnkovic, I.; Larsson, M.: Building Reliable Component-Based Software Systems, Artech House Books, June 2002.

[Ha92] Harel, D.; *Algorithmics: The Spirit of Computing,* Addison-Wesley, 1992.

[Ma02] May, J.: Testing the reliability of component-based safety critical software. In (S. Thomason, editor): *20th International System Safety Conference*, PO Box 70, Unionville, Virginia 22567-0070, August 2002; pp. 214—224.

[NM00] Napier, J.; May, J.: Empirical Assessment of Software On-Line Diagnostics Using Fault Injection. SAFECOMP 2000; pp.14-267.

[Pr96] Profeta, J. et. al.: *Safety-Critical Systems Built with COTS*. IEEE Computer, Volume: 29 Issue: 11 , Nov. 1996; pp.53-59.

[Vo98] Voas, J.: Certifying Off-The Shelf Software Components. IEE Computer, 31, June, 1998; pp.53-59.

[CHM02] Chen, L.; Hughes, G.; May, J.: The application of SFI in safety assessment of systems embedded with COTS components, Deliverable Report on Workpackage 4 (D6) of NewDISPO , Produced for British Energy, November 2002.

[CMH02] Chen, L,; May, J.; Hughes, G.: Assessment of the Benefit of Redundant Systems, Lecture Notes in Computer Science 'Computer Safety, Reliability and Security', volume 2434, Springer, Sept 2002. pp.151-162.

[QW91] Quirk, J.; Wall, N.: *"Customer Functional Requirements for the Protection System to be used as the DARTS Example"*, DARTS consortium deliverable report DARTS-032-HAR-160190-G supplied under the HSE programme on Software Reliability, June 1991