Research Paper

2002-12-11

Summary of Safety-Critical Systems, Formal Methods and Standards

Jimmy Persson (-)
Gustav Evertsson (-)

Blekinge Institute of Technology, Sweden

Index

Introduction

Overall description

The purpose of this paper is to document the research conducted in the field of verification and validation.

The research was performed during the course “Verification and Validation” at Blekinge Institute of Technology in Sweden.

Problem description

This paper is divided into three parts. The first is a summary of the report “Safety-Critical Systems, Formal Methods and Standards” by Jonathan Bowen and Victoria Stavridou. We present what we see as the main points from the report. The second part is a critical review of the report from our own view on the subject. The last part contains an evaluation from Fenton, Kitchenham, Tichy and Hamlet’s point of view.

Research

Summary of the report

When this report was written, there was lots of safety critical systems and hence, safety critical software systems. Such systems were greatly affected by software crisis. To gain reliability in these kinds of systems, formal methods are recommended. This report reviews the industrial use of these technologies, recommendations about formal methods, usability and problems when developing safety critical systems in industrial scale. Some future directions are suggested as well.

See it as one system

To achieve ultra high reliability one must consider all parts of the system, including both software and hardware components. For instance, an error that seems to be located inside a software component when it actually is located in a hardware component. Because of this, it is hard to measure the reliability in a system when just looking at one component. Further more, safety should not be added to a system, it should be designed into it from the beginning. In the fifties, software passed hardware when considering production costs. This was because hardware became more reliable, thanks to the transistor and integrated circuits. Hardware became more reliable since lots of issues were solved. Software on the other hand still had lots of problems, and therefore more instable. The gap was increased between what was possible to achieve and the requirements. It is not until lately that research has been performed to minimise the gap. The industry has been reluctant to adapt new methods. Just recently, case studies and experiments have been performed that shows the benefit of formal methods. Thanks to this the industry are now more willing to accept formal methods.

There are four ways of achieving high dependability. They are: Fault avoidance, fault tolerance, fault removal and fault forecasting. To get ultra high dependability, a combination of these all four methods must be used. Today it is not possible to get as high dependability ad may be wanted, because of the praxis used. N-version and ordinary software testing is used today to get high dependability.

Formal methods

Formal methods are the most cost effective technology, if it is used correctly. It is however not always motivated to use these methods every where in the software process. Sometimes it is sufficient to use them in parts of the process, and some other technology in the rest. It is however important to make correct quantifications, so that one can be certain what parts to verify and validate in the formal method. One widely spread misunderstanding is that it is very expensive to use formal methods. It is not, because this misconception is built on the fact that maintenance is mostly not included in the development planning. Later on, it often turns out to be a huge part of the total time used for maintenance.

Formal methods can be used in different areas

There are lots of standards when it comes to security. Some think that formal methods can help increase dependability in such standards. Here follows short information about some of the standards mentioned in the report.

Requirements capture is a very important phase in the process, and an error in this stage would be carried through the entire developer process. A correction of such error would cost approximately a thousand times more than correcting it in the beginning of the process. Statistics shows that about two thirds of all errors are made in this stage. It is therefore important to find as many errors as possible as early as possible, and not to proceed before a requirement is correct. It is hard to formalise because there is nothing to use for evaluation but reality. Timing is important for many safety critical systems, but it is hard to formalise this in a usable manner. It is important to be able to prove that the desired response time will be met under all circumstances. By using formal methods for proving that the design is correct, such assurance can be retrieved already in the design phase.

It is important to test the compiler that is used. The resulting machine code that it produces is virtually impossible to verify that it is correct. The compiler in it self can be unreliable, and therefore adds a new variable of insecurity into the developing process. Before, software was written in assembler to avoid these problems. Nowadays, high level languages are used and have been proven to produce more reliable software, which are reliable and stable.

Another important part to consider is the human computer interface. This is a difficult task to formalise, since there are a lot of factors involved.

Standards

There are lots of standards for handling security. Some say that formal methods can help in increasing dependability in such standards. There are perhaps too many standards these days, perhaps because they are specialised to fit specific tasks. In the beginning there was just the ISO9000 standard. Some standards that support formal methods are the following: RTCA DO-178, UK HSE, IEC, ESA, UK RIA, MoD 00-55 and 00-56, AECB Canada and finaly IEEE P1228.

Education and knowledge

Software engineers do not always have the required knowledge and experience to be able to use formal methods. A reason for this problem is that universities often think this is a too specialised topic to be included in the education. A possible solution, which is discussed, is to start using certification of developers. This can perhaps secure that the developers have the required knowledge. Another solution is to create laws, so that in case of disaster due to system error, the person responsible can either be forced to pay a large fine or even put to jail.

Problems with formal Methods

We have already discussed some of the problem that can be found with formal methods. Examples of such are the problem with requirement and design of real time systems. It can therefore be hard to apply formal methods on a new project before researches are done to investigate how the formal method will be used.

Measuring befits of formal methods can be hard, especially in terms of how much the dependability increases. It is even harder when it comes to measuring global dependability because we do not know how to combine the data with data collected from other techniques such as fault tolerance.

Future Research

The last part of the report is dedicated to discussion about future research. Formal methods are a new area within software development and are still under development. More case studies must be performed. This is because it is a fact that a highly trained expert can show that formal methods are good in a laboratory. It can not however be shown that one will get the same befits when using it in the industry.

Review

General

The authors of the report write that it should be seen more as a snap shot of the ongoing resource than a definite guide to formal methods. This is something that we consider a problem with this report. It is now more than ten years old and the numbers do not feel up-to-date. The conclusions drawn from them may still be correct but it is hard to trust them completely. They also write about, for example, object oriented software development as something new, but OO is widely used and accepted today. Another issue that they mention is that it is hard to formalise the process of developing HCI (Human Computer Interface). Today the knowledge in this area is extensive, and there are lots of studies about how people react on colours, shapes, location and such[1].

Today security, safety and reliability are seen from another perspective today than when this report was written. Today we have Internet, which has given engineers new problems to handle in the areas mentioned above. For instance, there is no time for planned downtime. Before this might not have been a big issue, because often there could be planed downtime during night time to perform maintenance. Today these are very present issues, and are taught at universities.

We encountered a contradiction when reading the report. First the authors say that it is cost effective to use formal methods. When reaching the end of the report, they say that it is more expensive to use formal methods. This can be because in the first statement they calculated the cost of a total project and the whole life cycle of it. In the second statement though, they just calculated the cost of finding and removing defects, without minding anything else. We think that this issue should have been more clearly described.

Generally we think that the report is well written, and easy to understand. It is pretty high level and gives a general overview on the subject. Some standards are mentioned, but not discussed in detail, which we think is good. It is good to know that standards exist and perhaps something about what their purpose is. It is impossible to keep track of all standards, so just a little information can be sufficient. If one wants to know more, they are specified in detail in other reports referred to in the references list in the end of the report.

Dependability

The report is built around dependability, which is defined as the trust in the fact that a service is doing what it is intended to do. Almost everywhere the conclusion in the reasoning when using formal methods is about how to achieve high dependability in a cost effective way. They also discuss what it would be like if formal methods were not used at all, from a dependability point of view. One thing that concerned us was that the authors talk a lot of how to achieve dependability by using formal methods, but they do not say what it is that they compare to. They just define it as some kind of standard method or technology to use. They do not compare it to other process models. If the report was written today, another process to compare formal methods to could be for instance extreme programming.

[1] Preece, J, Human Computer Interaction.

Certification

There are both pros and cons about certification. One negative side are as we see it, the cost for a company to be certified. It may be very expensive, and small companies may not have enough funds to be able to be certified. They may however have the right knowledge and experience. Hence, such companies can not be certified although they should be. Certification is positive for customers to companies because they will be guarantied a certain quality level.

Evaluation

Case studies

We know by now that the authors are positive to use formal methods, and they give some real world examples where such methods have been used. They do not say how the methods are used, or what people used them[2]. Formal methods obviously require a great deal of knowledge and competence in the area of the subject, to achieve the best result. When one are about to make a decision about what method to use, one should consider who performed the study, and the experiment, company or other, being the subject of the study. In the case of this report, we think it is very positive that they use examples from the real world and not made up studies invented to figure as examples. If the examples were performed by students in a university, the result would probably not be valued as high as if it were performed by professionals. The correctness of results are always questioned, but probably more so in the case of students, or less competent persons, are the ones that have performed or participated in the examples, since they are not as experienced as professionals.

Data manipulation

Data can always be manipulated[3]. The way data is chosen qualifies as manipulation with data. For instance, the choice of the specific examples used, may have been done with the purpose of making the reasoning about formal methods look good. In this case, the authors have picked examples from different areas. This can be positive in that one gets some perspective. One will however not be able to compare the result from many examples in same area, which could be preferable to get some more detailed contrasts on how well formal methods work.

Lack of information

So far we have discussed, amongst other things, if this report is sufficient to base a choice of using formal methods or to not use them. We believe that it does not contain sufficient information for basing such decision on it. According to the report, the area of formal methods is pretty new. As a consequence there is not much information on the subject. Not many case studies have been performed. The more case studies[4] and reports that can be used as a base for the conclusion, the higher the credibility will be. We therefore make the conclusion that this report is possibly not very accurate. Future studies show if this assumption is true or false.

In the case of this report, where there is a possible risk of faulty or even bad information, one still has to somehow filter the information to retrieve the correct one. To be able to do so, one will need to use intuition[5], and common sense. It will not be enough to use as grounding for a

[2] Davis, From the Editor, IEEE Software, p. 4.

[3] W. Tichy, Should Computer Scientist Experiment More?, p 39.

[4] Kitchenham, Pickard and Pfleeger, Case Studies for Method and Tool Evaluation, p 53.

[5] Davis, From the Editor, IEEE Software, p. 4.-7.

decision about the use of formal methods, but it will tell if it may be useful to enter deeply intro the subject. Then enough information can be collected to base a real decision on[6].

[6] W. Tichy, Should Computer Scientist Experiment More?, p 39.

Conclusion

We thing that the authors of the report we have studied, use the right detail level when discussing the subject of formal methods. That makes the report good for introduction on the subject. However, we recommend reading their report critically, and perhaps even getting other references to check on the information, since everything in the report might not be true.

The authors base most of the report contents on issues about dependability. It was an important factor then, and it still is, but perhaps in another sense. Today it is hard to have planned downtimes in critical systems, like Internet for instance. Another issue are embedded systems like mobile phones, systems in cars and so on.

Industry is more positive to use formal methods today than before, but other methods as well, like extreme programming. Perhaps the use of formal methods and their like's, has increased because the methods are better tested than before, and there is more information on the subject. We think that the more information about methods available, the more likely it is that the method will be used.

Source

Written

B. Kitchenham, L. Pickard and S. L. Pfleeger, Case Studies for Method and Tool Evaluation, IEEE Software, July 1995

Davis, From the Editor, IEEE Software, March 1996.

Jonathan Bowen and Victoria Stavridou, Safety-Critical Systems, Formal Methods and Standards, Oxford University Computing Laboratory, University of London, December 1992

N. Fenton, How Effective are Software Engineering Methods?, Journal of Systems and Software, Vol. 22.

Preece, J, Human Computer Interaction, Cambridge: Addison, Wesley, 1994.

W. Tichy, Should Computer Scientist Experiment More?, IEEE Computer, May 1998.