A Comprehensive Survey of System Dependability for Real Time Embedded Software

Khaled Almakadmeh
Assistant Professor
Department of Software Engineering
Hashemite University
P.O. Box 330136 Zarqa (13115), Jordan
(00962) 797676367
khaled.almakadmeh@hu.edu.jo

Ghadeer Al Qahmouss
Master Student
Department of Software Engineering
Hashemite University
P.O. Box 330136 Zarqa (13115), Jordan
(00962) 53903333
ghadeer.raji@itc.hu.edu.jo

ABSTRACT

Context: system dependability is an explicit indicator that such system satisfies its critical assets; through helping customers in choosing trustworthy software that meets their requirements. This will increase the awareness of customers for increasing their dependability of software delivery without soft-failures that may lead to massive losses in their lives and economical crises. There are few attempts to review and classify the proposed dependability approaches and summaries the dependability major challenges.
Purpose: this paper is aimed to conduct a survey for previous research studies of system dependability of real-time embedded software in a variety of application domains and to suggest some dependability solutions for future research.
Plan: a comprehensive survey with a proposed methodology is used to conduct a review for previous research studies regarding system dependability for real-time embedded software.
Preliminary Results: a survey has been created for identifying and classifying some of the system dependability terms and approaches used by academia. Several papers are retrieved from the literature including conferences and articles to build a comparative study of system dependability approaches and techniques in order to highlight for dependability challenges in different contexts.
Conclusion: the results of this survey is to address different types of challenges and benefits from previous research studies also to emphasize the importance of using dependability strategies (i.e. fault avoidance, fault detection and correction and fault tolerance) while planning for systems dependability.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – *Methodologies.*

General Terms

Measurement, Design, Standardization.

IPAC '15, November 23-25, 2015, Batna, Algeria
© 2015 ACM. ISBN 978-1-4503-3458-7/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2816839.2816917

Keywords

Self Monitoring Architecture, System Redundancy, Dependability Engineering

1. INTRODUCTION

Recently, there is a wide effect of technology evolutions in our life and increasing demands force for having an efficient computer

based systems including education, medicine, industries, utilities and communication as well as transportation against traditional routine in daily works. The acceptance of such changes require convincing and satisfying the stakeholders toward any kind of technologies added to their work nature and weather the suggested system is dependable enough or not.

Software dependability attribute means: the ability of the software to deliver trusted functional services under the term of availability, maintainability, safety and reliability, while the customers are looking for reliable systems. At the contrary, users are not looking for systems that cause failures. Such failures are easy to deal with and correct; whilst the other failures may lead to defeat life such as failures that occur in life critical systems.

There are few published tools and techniques regarding system dependability engineering to enhance the software dependability of critical systems use a set of strategies such as fault avoidance, fault detection and correction as well as fault tolerance [19]. For instance, fault avoidance or prevention strategies can be used for avoid and reduce the amount of errors in design and programming code languages.

In addition, fault detection and correction approach performs while deploying a system at the operational level to detect the maximum number of specification, design and program errors and then correct them if possible, in order to give the stakeholders an evidence of software dependability level.

Finally, fault tolerance approach is used to discover the runtime errors during execution and manage them in order to prevent system failure in case of life critical systems where high level of dependability required; and therefore, a specific fault tolerance techniques are used (i.e. fault tolerance system architecture).

This does not mean that the system will be delivered free of errors; there are enduring faults stay at all software system, but a development team usually struggles to discover the maximum number of defects before delivery. There are attentions at the industry to have a dependable software system which may be extended to have a dependable process and a strong development team using appropriate development techniques that provide with promise dependability of current software. However, such confirmation is not enough for higher level of dependability especially in life critical systems.

This paper will put up a comprehensive survey for a common used approaches and techniques for dependability to explore some strength and weaknesses of such published work while there is still some software that affect our lives.

This paper is organized as follows: section 2 presents the research description; section 3 presents the review of the literature; section 4 presents preliminary results. Finally, section 5 presents conclusion of findings and future research directions.

2. RESEARCH DESCRIPTION

This section will present the research methodology, research dependability framework and the objectives of the comprehensive survey.

2.1 RESEARCH METHODOLOGY

This survey will use the most common research methods to investigate our objectives by provide a well-defined process for identifying and classifying some of the system dependability terms and approaches used by academia and to present a comparative study between system dependability approaches and techniques to highlight for some dependability challenges in different contexts. Figure 1 specifies the followed process in details.

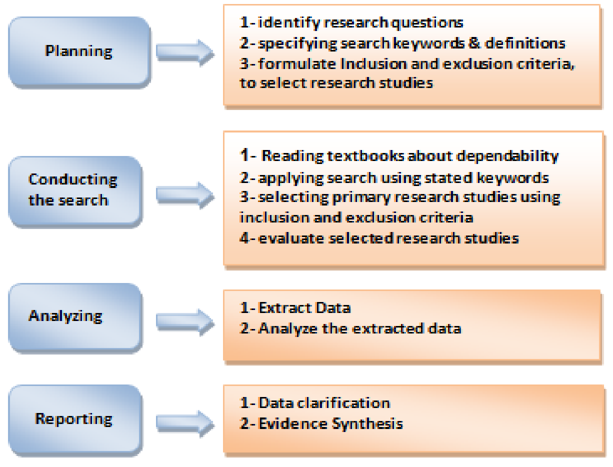

Planning
1- identify research questions
2- specifying search keywords & definitions
3- formulate Inclusion and exclusion criteria, to select research studies
Conducting the search
1- Reading textbooks about dependability
2- applying search using stated keywords
3- selecting primary research studies using inclusion and exclusion criteria
4- evaluate selected research studies
Analyzing
1- Extract Data
2- Analyze the extracted data
Reporting
1- Data clarification
2- Evidence Synthesis

Figure 1. Research methodology

2.2 RESEARCH FRAMEWORK

The framework for this survey consist of research studies, surveys or case studies contain new dependability approaches designed based on redundancy & diversity mechanisms, self-monitoring architecture, V-versioning programming, fault prevention, fault detection and correction, and fault tolerance. In addition, studies mentioned major dependability attributes (i.e. availability, maintainability, reliability, and safety) under these definitions in Table 1 [19]. It includes publications within the time period from 2010 to 2015.

Table 1. Definition of dependability attributes

Dependability attribute	Definition
Availability	Readiness of functionality [19].
Maintainability	The ability of system to tolerate modifications and repairs [19].
Reliability	Continuity of correct service [19].
Safety	The impact of catastrophic events on both of users and environment [19].

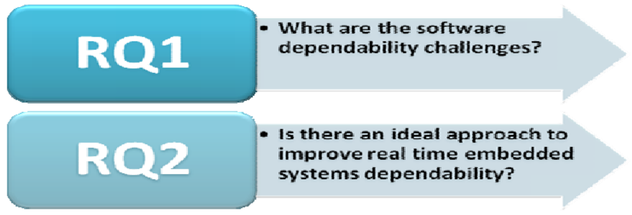
2.3 Research objective

The objective of this research is to analyze software system dependability approaches and techniques for soft real-time embedded software. Such analysis will help to build an approach that result on dependable real-time embedded software.

3. REVIEW OF THE LITERATURE

Step1: Research planning

In this survey, it is important to include most recent papers that proposed new mechanisms to improve dependability and analyze them by extracting the approach, the challenges and the benefits. Then, organize the extracted information in a way that answers the assigned research questions. We addressed two research questions as follows:


RQ1
What are the software dependability challenges?
RQ2
Is there an ideal approach to improve real time embedded systems dependability?

Establishment of selection criteria: formulating inclusion and exclusion criteria

Inclusion Criteria:

- The paper proposes an approach for enhancing dependability;
- The paper related to fault avoidance, fault detection and correction and fault tolerance strategies;
- The paper concentrates on critical systems;
- The paper focus on redundancy & diversity technique;
- The paper related to system dependability for real-time embedded software;
- The paper written in English language; and
- The paper mentions voting-systems.

Exclusion criteria:

- Every paper that is not written in English.

- Papers not related to the specified time period.
- Papers did not concentrate on availability, maintainability, reliability, and safety under the context of dependability.

Step 2: Conducting the Search

This survey is designed by following stated research strategy:

- Read the Software Engineering Body of Knowledge (SWEBOK) to highlight the main concepts of the research topic [22].
- Searching relative studies, surveys, articles, conference proceedings using these search terms: software dependability engineering; real time embedded systems; dependability by self-monitoring architectures; dependability based on redundancy & diversity.
- Contacting with some authors of papers Jose Merseguer to obtain further resources related to the research topic.

Step 3: Selection of the Primary Research Studies

Select the most recent studies between 2010- 2015 (See figure 2) to summarize the latest improvements to employ dependability attributes within a system. We have selected one-hundred (100) research papers. The selection of such papers was based on the inclusion and exclusion criteria presented in Section 3 earlier. After applying these criteria; the research papers that cover different areas are listed in Table 2.

This review and other secondary papers that assist in adding more details, full text reading and deep analysis has been conducted which give us a clear sight to construct our framework. Some papers were excluded as they are unrelated and discuss deeply software dependability concern.

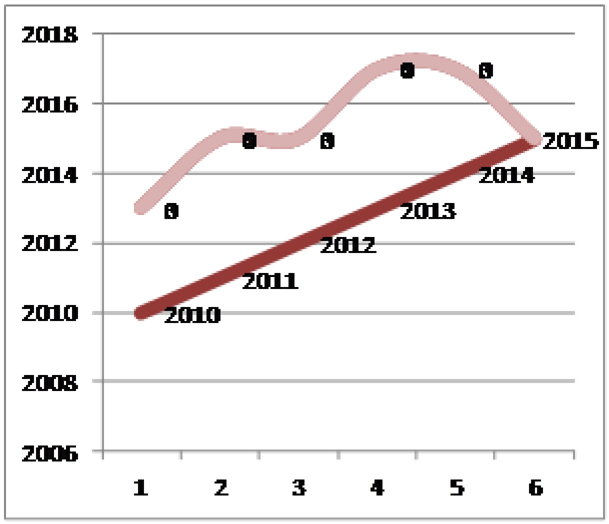


Figure 2. Number of selected papers published per year

4. PRELIMINARY RESULTS

4.1 DATA EXTRACTION

According to the chosen papers, we attempted to read the full text, remarks to main important in sequence like paper objectives, results of the proposed approach and their shortcomings. After that, we keep such papers in especial form for frequently check; this is to check there are a duplicate data to choose or not.

4.2 EVIDENCE SYNTHESIS

To answer the survey questions we analyzed the extracted data and used in three comparisons. The first comparison investigates the area, approach, shortcomings and results - see table 2. The second comparison shows the appearance of dependability sub-characteristics in proposed approaches (i.e. reliability, availability, maintainability, safety, security and performance - see Table 3). The third comparison classifies the studies based on which dependability strategy has been used (i.e. fault avoidance, fault detection and correction as well as fault tolerance) in order to construct significant approach - see table 4.

RQ1: What are the software dependability challenges?

In order to answer RQ1 this survey primarily summaries the main challenges that may prevent have a desired level of dependability sometimes the weakness stand for the nature of system and its environment especially critical systems or systems with dynamic nature.

Some approaches did success in providing dependable systems but they do not consider the importance of balancing between dependability attribute that may cause new requirements. In addition, they do not take into account the time constraints even in reliable systems. Several approaches fail to reach a maturity level; also none of them try to create a measurement to quantify dependability attributes (maintainability, safety, reliability or availability) in addition to the complexity of system structure.

RQ2: Is there an ideal approach to improve real time embedded systems dependability?

Still there is a negotiation about the best practice must follow to enhance a considerable level of dependability gains stakeholders satisfaction. actually there is no generic mechanisms to apply dependability on any type of software ,due to the diversity of fields employ the information technology in their works, also referred to nature of software itself (i.e. embedded systems, dynamic software), and the initial structure and the implementation. Software engineering science defined some of approaches, rules and guidelines that guide the process of ensuring dependability such as the Fault avoidance, fault detection and correction and fault tolerance approaches. Also, we recommend to ensuring the quality control at the beginning of development of software design and implementation phases .of course having a dependable system architecture, self monitoring architecture and V-versioning programming lead to dependable software.

5. CONCLUSION

In this paper, we presented a survey which addressed different kinds of challenges and benefits from past projects also emphasize the importance of using dependability strategies (i.e. fault avoidance, fault detection and correction and fault tolerance) while planning for systems dependability. Future work will focus on how to enhance dependability on embedded systems and systems with dynamic behaviors, mainly it's hard to apply dependability on dynamic systems according to their unexpected behaviors while execution which need a high forecasting to specify every single failure may happen in order to avoid them.

6. References

[1] Joseph P. Near, Aleksandar Milicevic, Eunsuk Kang, Daniel Jackson, A Lightweight Code Analysis and its Role in Evaluation of a Dependability Case, ACM, 2011.

[2] Gabor Karsai, Abhishek Dubey, Nagabhushan Mahadevan. Application of Software Health Management Techniques, ACM, 2011.

[3] Flouts Khomh, On Improving the Dependability of Cloud Applications with Fault-Tolerance ACM, 2014.

[4] Jörg Henkel, Lars Bauer, Hongyan Zhang, Semeen Rehman, Muhammad Shafique, Multi-Layer Dependability: From Microarchitecture to Application Level, ACM, 2014.

[5] Kiev Gama and Didier Donsez, Applying Dependability Aspects on Top of "Aspectized" Software Layers, ACM, 2011.

[6] Martin Randles, A.Taleb-Bendiab, Thar Baker, Towards the Automated Engineering of Dependable Adaptive Services, ACM, 2012.

[7] Kiev Gama, Didier Donsez, A Survey on Approaches for Addressi Dependability Attributes in the OSGi Service Platform, ACM, 2010.

[8] Alexander Romanovsky, John Fitzgerald, Giovanna Di Marzo, MetaSelf – An Architecture and a Development Method for Dependable Self- Systems, ACM, 2010.

[9] Leonardo Montecchi, Paolo Lollini, Andrea Bondavalli, A Reusable Modular Toolchain for Automated Dependability Evaluation, ACM, 2013.

[10] Armin Zimmermann, Dependability Evaluation of Complex Systems with TimeNET ACM, 2010.

[11] Cuauhtémoc Castellanos, Thomas Vergnaud, Etienne Borde, Thomas Derive, Laurent Pautet, Formalization of Design Patterns for Security and Dependability, ISARCS '13 Proceedings of the 4th international ACM Sigsoft, 2014.

[12] Jörg Henkel, Lars Bauer, Joachim Becker, et al., Design and Architectures for Dependable, Embedded Systems. Proceedings of the 9th International Conference, IEEE, 2011.

[13] J.-H Oetjens et al., Safety evaluation of automotive electronics using virtual prototypes: State of the art and research challenges. In Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE. IEEE, 2014. pp. 1-6.

[14] Kaliappa Ravindran, Self-Assessment and Reconfiguration Methods for Autonomous Cloud-based Network Systems. In: Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications. IEEE Computer Society, 2013. p. 87-94.

[15] José Luís Nunes. Improving the dependability of FPGA-based real-time embedded systems with partial dynamic reconfiguration. In: Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on. IEEE, 2013. p. 1-4.

[16] HANEN, Haouas; BOURCIER, Johann. Dependability-Driven Runtime Management of Service Oriented Architectures. In:PESOS-4th International Workshop on Principles of Engineering Service-Oriented Systems-2012. 2012.

[17] FEITOSA, Daniel. An architecture design method for critical embedded systems. In: Proceedings of the WICSA 2014 Companion Volume. ACM, 2014. p. 15.

[18] ESTEVE, Marie-Aude, et al. Formal correctness, safety, dependability, and performance analysis of a satellite. In: Proceedings of the 34th International Conference on Software Engineering. IEEE Press, 2012. p. 1022-1031.

[19] AVIZIENIS, Algirdas, et al. Fundamental concepts of dependability. University of Newcastle upon Tyne, Computing Science, 2001.

[20] MUSTAFIZ, Sadaf; KIENZLE, Jörg. A survey of software development approaches addressing dependability. In: Scientific Engineering of Distributed Java Applications. Springer Berlin Heidelberg, 2005. p. 78-90.

[21] Ian Somerville ,Ninth Edition, Software Engineering, 2009.

[22] Software Engineering Body of Knowledge (SWEBOK) ISO Standard 19759, 2014

Table 2. Dependability Approaches Comparison

Paper ID	Area	Approach	Results	Shortcomings
P1	Medicine (control software of medicine device proton therapy machine)	1. Achieved a considerable level of balance investment in analysis and design, & get some degree of confidence by applying more lightweight analysis. 2. Using a lightweight analysis affected by the nature of system structure and code. **[1]**	1. Propose an approach that identify the critical properties from the problem domain, build the structure of a property-part diagram then try to understand the relations between the component with both of system level requirements component and code. 2. Dependability evaluation technique by A lightweight code analysis based on symbolic execution. 3. Proposed a strategy for enhancing dependability by creates a list of side conditions reflect the environmental expectations after that review them due to modifying the design and implementation to exclude the non desired assumptions. **[1]**	1. Lightweight analysis cannot handle with extensive global state, Dynamic allocation or concurrency. 2. This approach does not consider the time constraints. 3. The soundness of analysis depends on the accuracy of domain knowledge provided by user. 4. The analysis itself does not provide sufficient evidence to improve dependability; it must be integrated to other helpful. **[1]**

P2	Real time embedded system	Propose an approach to enhance dependability by adapt system health management (SHM), it's a dynamic fault removal technique works at run-time which provide detection, isolation, and mitigation actions to discover and remove system faults ,this approach applied on tow level component and system level as a whole in general its attempt to identifying self-adaptive system. **[2]**	The approach has been evaluated by their experiment (SHM) that performs its successfulness in large-scale and industrial applications. **[2]**	1. The approach specialized only for industrial systems. 2. Applicable only in large-scale systems may that affect it efficiency. 3. Fault management systems need a verification to make sure it doesn't harm the safety rules while adaptive systems verification mainly a big challenge. **[2]**
P3	Cloud Computing	A general view discusses some challenges related to the implementation of most common fault-tolerance mechanisms usually used to enhance dependability to cloud computing Applications based on N-versioning programming (NVP) and self-checking programming). [3]	Suggest recovery mechanisms that are work effectively with honor the dynamic nature of cloud apps and platforms. [3]	1. It is hard to include fault tolerance mechanisms cloud applications because of its dynamic nature. [3]
P4	Embedded Systems	Create a multilayer design flow for a multi-layer dependability that attempt to reduce propagation of errors appear while adaptation and exchange information, a multi –layer concept means tow hardware or software abstraction layers adapted to each other. During design or implementation process. **[4]**	This approach could treats dependability in term of design constraint a prerequisite especially for complex and dependable future on-chip systems. **[4]**	
P5	Dynamic Platforms (i.e. Java & .Net)	Propose an approach that focus on improve dependability and monitoring by apply a cross cutting concern on dynamic platforms where system components can be stopped, updated ,installed, started, or uninstalled at the same time of execution . **[5]**	After deployed the approach on OSGi dynamic platform it provide a desired level of portability across different OSGi frameworks better modularity and maintainability of code, and flexibility into combination of different aspects. **[5]**	Some limitations refer to the technical solution introduced by the dependability and monitoring aspects implementation also issues related to their nature. **[5]**
P6	Cloud Computing	Perform an approach adapts the cloud base service composition to be safety in order to enhance the required level of maintainability by specifying a formal model that enact independently from the run time implementation to clarify the idea they used a calculus situation and software representation technique for the desired composition process such as intention description language **[6]**	1. The situation calculus has been represented effectively. 2. The cloud base service composition has been adapted successfully in such a safe and effective way. 3. This technique may need to integrate with and support established standards like (WSDL and WS-BPEL). **[6]**	1. Deploying this approach perhaps lead to loss in performance. 2. useful in wider applications where testing is necessary. **[6]**
P7	Dynamic Platforms	Survey on latest researchers efforts toward improving dependability stand for the context of fault tolerance mechanisms in the OSGi dynamic service plat-form **[7]**	Assessing the level of addressing dependability attributes in the well-known fault tolerance approaches used recently even the attributes never treated. **[7]**	1. Totally there are good approaches to support OSGI dependability, but it's not that mature and experimental enough. 2. None of remains approaches try to create measurements for each of safety, Reliability, maintainability and availability although they are usually quantified. **[7]**
P8	Self organizing Systems	A meta-Self approach that function as software architecture and development method as means to improve dependability and controller over self organization systems, to combine rules for self organization with dependability policies enforced at run-time while updating or retrieving metadata. **[8]**	The approach have been viewed by two examples	

P9	Model-Driven Engineering (MDE) techniques	This paper presented a Tool chain for state based- dependability analysis: it's a flexible tool easily adapted to other modeling languages and analysis tools it was implemented as a plug-in for Eclipse **[9].**	1. The Toolchain architecture allows plug-in elements can be developed in isolation, Also can be assigned to different persons or teams according to their skills. 2. Toolchain achieves reliability when a specific system evaluated automatically **[9]**.	No weakness points were mentioned it worked well as a plug-in with Eclipse platform. **[9]**
P10	Critical Embedded Real Time Systems (CERTS)	Proposed an approach to attach dependability and security automatically to after formalize them as design patterns then configure these patterns into model transformation, preconditions and post-conditions. **[10]**	The approach validated using the R/B architecture pattern, it's an architectural pattern used to create and recognize the system, in addition to behavioral pattern to define the modes and behaviors of components. **[10]**	Designing a pattern may affect the whole system (other non functional requirements, quality of services) **[10]**
P11	Embedded Systems	An overview of the recent research projects that focus on dependable embedded systems. **[11]**	1. Identifying dependability co-design its recommend that embedded systems must have different level of abstraction within design process. 2. Provide a modern classification to each of faults, failures and errors. **[11]**	1. It's hard to forecast the temporal behavior of most applications. 2. The innovative errors classifications impact positively application dependant, we can't say there is a specific classification combine all applications. 3. May required a huge amount of compiler generated data for making a run time decision during the program execution. **[11]**
P12	Embedded systems (Complex)	View the latest version of Time NET tool that, allows dependability modeling of complex systems with colored stochastic Petri nets, also help in model simulation that's necessary for highly dependable systems it's a well known tool help to measures of complex dynamic fault-tolerant systems. **[12]**	1. Has been successfully applied on projects for modeling and evaluation objectives. 2. It is commonly installed in different companies and universities. **[12]**	
P13	Embedded systems (Complex)	Proposed Virtual Prototype (**VPs**) approach . In order to investigate the safety and correctness of provided functionalities. Even in complex embedded systems.	1. VPs could observe and track the impact of faults and its propagation at embedded systems . 2. VPs can help in forecasting the unexpected errors . from early stages at design phase. **[13]**	1. a consistency between VPs and real system is a must. 2. sometimes it's hard to build a simulation using VPs for some scenarios . 3. the main challenge is how mapping abstractions level between real world faults and detected faults using VPs. 4. mainly its hard to apply stress testing using VPs ,aims to analyze the impact of observed faults in system safety. 5. investigating safety may increase the complexity problem. **[13]**
P14	Cloud Computing	Identifying fault tolerance techniques and tools. In order to create metrics that quantify dependability of cloud based applications. Illustrated by system level strategies to evaluate dependability of **S** ,which used to ensure safety requirements **[14]**	1.minimize the typical cost of development of distributed control software . which support system management through service level programming and reusing model. 2. the presented approach evaluated by a case study of replicated data service which corroborated the cloud infrastructure **[14]**	1. this approach support a model based evaluation of dependability. In cloud base systems especially Internet-type network systems **[14]**

P15	Embedded Systems	Introduce an approach describes the process of adopt partial dynamic reconfiguration prosperities of FPGA device. FPGA is a prototyping platform that let dynamic models to be reconfigured while therein another static models running continuously .aims to enhance the dependability of real-time embedded systems **[15]**	1.They could improve the resilience of FPGA-based systems by get the benefits of PDR. 2. Achieved a faster reconfiguration speeds. 3. Specified a detailed dynamic features of real time embedded systems by using (PRD). 4. improved critical to time sensitive application, the recovery speed, and minimized space overhead **[15]**	1. some problems related to the weakness of provided information from manufacturer **[15]**
P 16	service oriented application	A framework consist of service monitoring, decision-making reputation, evaluation, reconfiguration	1.The approach successfully evaluated and tested using a scenario of application based on service **[16]**	It is essential to evaluate framework it in different fields **[16]**

Table 3. Non-Functional Requirements Comparison

Paper ID	**Dependability attributes**							
	Reliability	**Maintainability**	**Safety**	**Availability**	**Integrity**	**Confidentiality**	**Security**	**Performance**
P1			√					
P2	√		√					
P3				√				
P4	√							√
P5	√	√						
P6	√		√					
P7	√	√	√	√	√	√		
P9	√			√				√
P10			√					√
P11							√	
P12	√							
P13	√		√	√				
P14			√					√
P15	√							
P16	√							
P17			√				√	√
P18			√					√

Table 4. Dependability strategies comparison

Paper ID	**Used Strategy**				
	Fault Avoidance (Prevention)	**Fault Detection and Correction**	**Fault Tolerance**		**Redundancy And Diversity**
			N-Versioning programming	**Self Monitoring mechanism**	
P1			√		
P2				√	√
P3			√	√	
P4					√
P5				√	
P7			√		√
P8				√	
P9					
P10			√		
P11					√
P12		√	√		
P14				√	√
P15		√		√	√
P16				√	