Noname manuscript No.
(will be inserted by the editor)

Dynamically Reconfigurable Systems: a Systematic Literature Review

Gabriel Fornari · Valdivino Alexandre de Santiago Júnior

Received: date / Accepted: date

Abstract Reconfigurable systems have evolved as a more comprehensive and better known area in the last years. Reconfigurability is strictly related to the ability to change: the more flexible a system is, the greater is its reconfigurability. Reconfiguration can provide the systems characteristics as self-adaptation, allowing their resources to be used according to the environment in which they are found and, consequently, extracting a better use of these resources. Unmanned Aerial Vehicles (UAVs), mine hoists, mobile robots, and balloon systems are some applications where self-adaptation and reconfiguration are important. Some reconfigurable systems are able to plan their reconfiguration at runtime, i.e., the system sets its new configuration while running. These systems are called Dynamically Reconfigurable Systems (DRSs). This paper aims to investigate DRSs seeking to answer four specific questions: (i) how the different kinds of DRSs are classified in the literature and what is their definitions; (ii) what are the hardware and software platforms, methodologies and techniques engaged in DRSs; (iii) what are the domains of application of DRSs; and, (iv) which countries lead the number of publications in DRSs. To do that, a systematic literature review was conducted, where, at the end, 85 articles between 1995 and 2017 were completely read.

Keywords Reconfigurable Systems · Dynamically Reconfigurable Systems · Self-adaptive · Systematic Literature Review

Gabriel Fornari
Avenida dos Astronautas, 1758
Jardim da Granja - 12227-010
São José dos Campos, SP - Brasil
E-mail: gabriel.fornari@inpe.br

Valdivino Alexandre de Santiago Júnior
Avenida dos Astronautas, 1758
Jardim da Granja - 12227-010
São José dos Campos, SP - Brasil
E-mail: valdivino.santiago@inpe.br

1 Introduction

Reconfiguration is a property growing in importance in today's systems. The change in a system configuration may be related to different areas, such as a change in a product design, upgrade in a software application, fault tolerance, or even replacing obsolete parts. In this sense, reconfiguration can be seen as an important part of a system design [4].

Reconfigurability is strictly related to the ability of changing, i.e., how flexible the configuration of a system can be. A reconfigurable system can be simple as a digital clock or complex as a network composed of hundreds of computers, and may involve different areas, such as reconfigurable matter and digital and analog electronic systems [54].

The concept of reconfigurable systems is extremely broad and may present different meanings [26, 54]. Digital systems, such as computers, are reconfigurable systems as well as a 3D printer, capable of modeling materials in different forms [54]. However, there is a class of such systems called Dynamically Reconfigurable Systems (DRSs), that allows reconfiguration after its manufacture and during its operation.

DRSs are able to perform their reconfiguration at runtime, i.e., to set a new configuration while running. This is the main difference from static systems, such as Application Specific Integrated Circuits (ASICs), which once manufactured can not be changed, or systems that allow a single configuration, such as those based on Programmable Read Only Memory (PROM), or systems whose reconfiguration is a complex operation and takes a large amount of time, such as those based on Erasable Programmable Read Only Memory (EPROM) [54].

Dynamic reconfiguration enables real-time systems adaptation. Unmanned Aerial Vehicles (UAVs), which have become popular in the last decade, can be built with fixed wing or rotors. There are researches that study the possibility of using both systems, the so-called hybrid UAVs, which can take off on a small runway, but keeps the benefits of fixed wings [33, 3]. In this case, the exchange between both systems characterizes a reconfiguration, and the concept of DRSs can be applied. In addition, reconfiguration and adaptation have already been used in several real intelligent control and robotic systems. We give some examples below.

Simulations demonstrate that using intelligent control system based on brain emotional learning may present better results in attitude control of UAVs than traditional methodology [71]. An intelligent control system based on ant colony and fuzzy self-adaptive PID control was proposed for mine hoists automation [53]. An autonomous mobile robot that switches internal systems to adapt to the environment and reach a goal without colliding with walls is studied in [51]. A control algorithm was developed to operate a balloon semi-automatically under a series of conditions and seeking to save energy consumption [85].

In addition, still in the area of UAVs, the autonomous navigation system is another possible example of DRSs. There is a survey that presents different ways of performing autonomous navigation, such as through the use of GPS or imaging sensors [41]. A DRS could manage these systems, using the most appropriate depending on the environment, or using them redundantly to avoid failures.

Another possible example of the importance of DRSs is in avionics. The new generation of Integrated Modular Avionics (IMA) platform allows a reduction in the required number of aircraft control equipment. The purpose of a reconfigurable system in this case is to use the same electronic device for different functions, rather than using a different device for each function [11].

Historically, it can be said that control systems engineering was the first discipline to put efforts on the design of systems reconfiguration mechanisms, allowing its automation. It has played a crucial role in the development of piloting systems for aircrafts, guided missiles, and industrial production [62]. With the development of smaller and more powerful digital computers, its use as control units managing reconfigurations has become an increasingly common practice [62].

Based on the relevance of DRSs for intelligent systems, the goal of this work is to investigate the area of DRSs addressing the following research questions:

1. What is the understanding of the scientific community to define a DRS?
2. What are the hardware and software platforms, methodologies and techniques engaged in DRS in a broader context, and not specifically related to only some characteristics of DRSs?
3. In what application domains DRSs are used?
4. Which countries have published more in this area?

By answering these questions, we aim to perceive how DRSs are understood within the community. Although there are surveys published in this area, to the best of our knowledge, no other article addressed the same questions presented above. For example, Lyke et al. present concepts, a taxonomy and future challenges in the area of reconfigurable systems, but the focus of their work is not DRSs [54]. Jzwiak and Nedjah present and discuss the concept of reconfigurable systems, but the main objective of the work is related to reconfigurable computing and architectures for embedded systems [37]. Krupitzer et al. present a study on self-adaptive systems and a taxonomy to classify them, but the focus of their work is more conceptual and theoretical than practical [49]. Also, as far as the authors know, no other Systematic Literature Review (SLR) was performed in the same context.

The idea of an SLR is to define a research protocol and strictly follow it. In this way, the results can be reproducible [46]. In addition, an SLR should delineate the state of the art of a research area and present its gaps. An SLR is considered a secondary study because it extracts information from other studies.

There are several ways to perform an SLR [46]. In this work, the search methodology involved a search string, i.e., a string of keywords is built and employed in several different databases, such as IEEE and Scopus, to select papers related to the keywords. Inclusion and exclusion criteria are applied to decrease the initial number of papers and improve its selection. From the final set of selected articles, properties are extracted with the aim of answering the research questions.

In this work, 5 databases were used and a total of 616 articles were obtained at first. Based on this first retrieval, 72 were completely read. In addition, 13 papers, which were not obtained through the string, were included in the review. In total, 85 papers published between 1995 and 2017 were read, being 77 selected for data extraction. The difference (85 to 77) is due to similar contents, being kept only the most complete version of the paper, and to quality assessment, being excluded papers with low score.

This paper is structured as follows. Section 2 presents in detail all the steps of this SLR. Section 3 presents the main results and analysis based on such results. Concluding remarks and a research agenda are in Section 4.

2 Systematic Literature Review: Steps

The SLR conducted in this work was based on [46, 47, 56]. Before starting the review itself, we conducted a search including *Google Scholar* and *Compendex* data bases to verify the existence of other secondary works about DRSs. The following search string was used:

Dynamically Reconfigurable System AND (Review OR SLR OR Survey OR Taxonomy)

Through visual inspection, the papers were analyzed and selected for a more detailed inspection. Among all papers returned, none of them presented an SLR related to DRS. Furthermore, with only one exception, all the selected papers presented discussions about reconfigurable computing. They are briefly discussed below.

In [81], a survey about Field Programmable Gate Array (FPGA) coarse grained architectures is presented. Some of these architectures were also identified during the study of other works in this review. In [58], approaches for reconfigurable architectures are presented. However, due to the year of publication, the technologies involved are already outdated, although the definitions are still valid.

In [82] and [77], solutions and applications of reconfigurable computing for digital signal processing are presented. Finally, [87] present an abstract concept of dynamic reconfiguration and 6 different types of reconfiguration for evolving software systems.

The papers presented above can serve as a basis to understand the concepts related to DRS in computing, but none of them discuss dynamic reconfiguration covering questions about definition, classification and application on industrial systems. Furthermore, to the best of our knowledge, no SLR has been published in this area before, justifying the necessity of the present work.

Figure 1 presents all the steps followed in this work. Step 1, need of a systematic literature review, has just been discussed. The next steps are presented in the sequence.

2.1 Review Protocol

In accordance with Figure 1, the next step is to define the review protocol, which encompasses the research questions, the search methodology and the inclusion/exclusion criteria. Furthermore, we also present the verification of the review protocol. In Table 1 the research questions and its goals are presented.

As shown in Figure 2, the search string was built through an iterative methodology, in which experimental searches were used to refine the string. The papers returned after each search were analyzed through a random inspection to verify its coherency with the aim of the research questions. Only metadata, i.e. title, abstract and keywords, were used during the search. Besides that, it was decided that the total number of papers must be between 400 and 600.

At the beginning, 4 databases were selected: Scopus, ACM, Web of Science, and IEEE. As the number of articles was below the defined, the database Science Direct was also included. The searches were made using the search engines available on the web sites of the respective databases. The final version of the search string is the following:

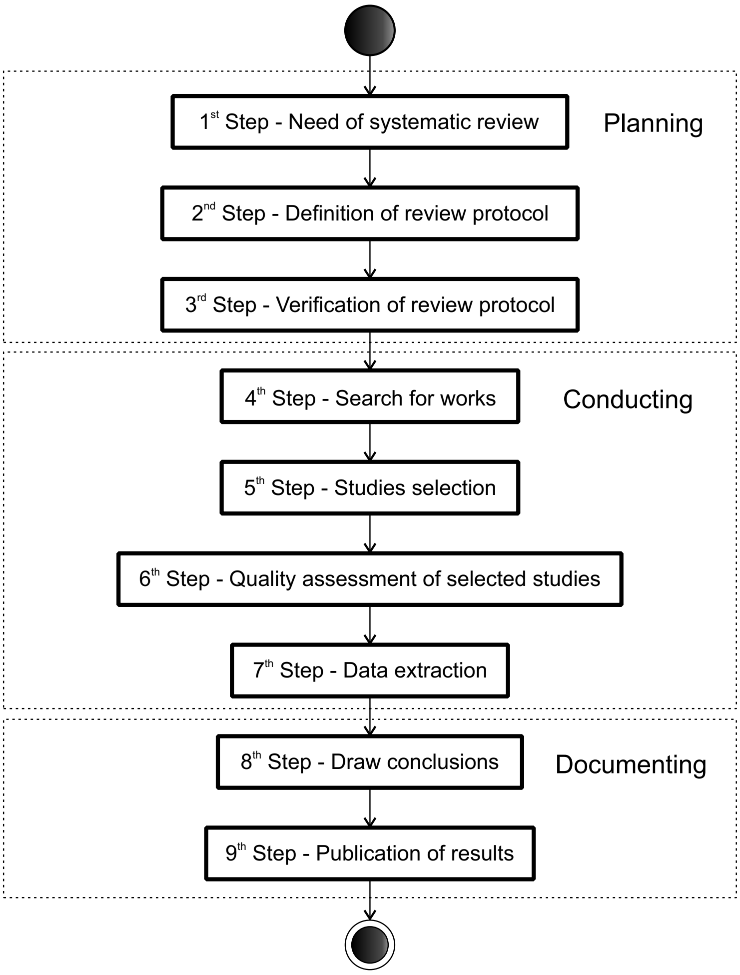

Fig. 1 Steps followed in this systematic review adapted from [56]

((“Dynamic reconfigurable system” OR “Dynamically reconfigurable system” OR “Dynamically self reconfigurable system” OR “Dynamic reconfigurable embedded system” OR “Dynamic reconfigurable real-time system” OR “Reconfigurable embedded system” OR “Dynamic adaptive system” OR “Dynamic adaptable system” OR “Dynamic self adaptable system” OR “Reconfigurable real-time system” OR “Partial reconfigurable system”) AND (“hardware device” OR “system on chip” OR “system-on-a-chip” OR “system-on-chip” OR “embedded software” OR “embedded system” OR “image processing” OR “image processor” OR “image analysis” OR “classification of images” OR “processing of images” OR “image-processing”)).

The above string was applied to four databases: Scopus, ACM, Web of Science, and Science Direct. Since the database IEEE allows a maximum of 15 keywords in the search engine, the original string was modified to the following one:

Table 1 Research questions and its respective aims

Id	Research Question	Aim
RQ1	How the different kinds of DRSs are classified by literature and what are their definitions?	Understand what is the definition of DRSs, what features they can present and verify the existence of some taxonomy for these systems.
RQ2	What are the hardware and software platforms, methodologies and techniques engaged in DRS?	Understand how DRSs can be developed in practice, i.e. what means exist to create a dynamically reconfigurable system.
RQ3	What are the domains of application of DRSs?	Understand in what contexts DRSs are used in practice and for what purpose.
RQ4	Which countries lead the number of publications in DRS?	Outline the number of papers per country by associating the authors with their institutions.

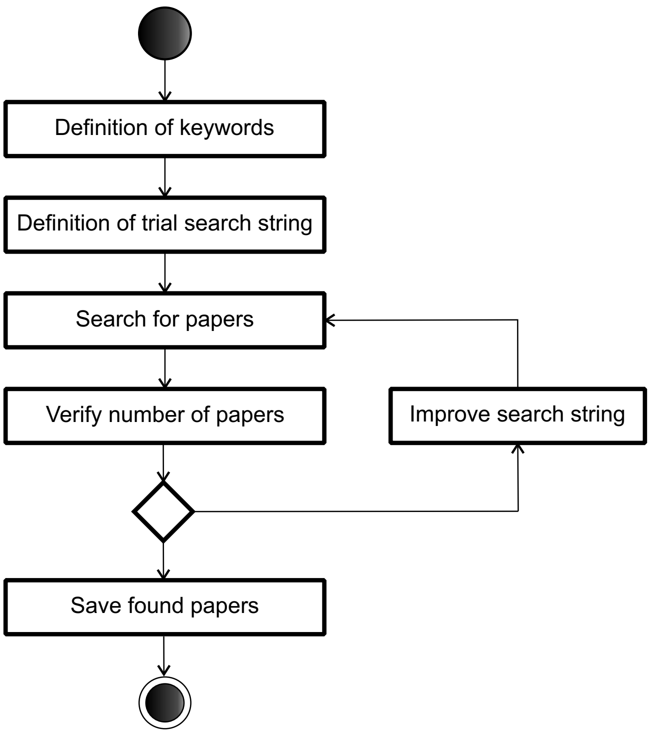

Fig. 2 Steps for creation and improvement of the search string adapted from [56].

((“Dynamic reconfigurable system”) OR (“Dynamically reconfigurable system”) OR (“Dynamically self reconfigurable system”) OR (“Reconfigurable embedded system”) OR (“Partial reconfigurable system”)) AND ((“hardware device”) OR (“system on chip”) OR (“system-on-a-chip”) OR (“system-on-chip”) OR (“embedded software”) OR (“embedded system”) OR (“image processing”)).

The keywords of image processing were added for the secondary purpose of selecting articles that could bring reconfigurable systems as potential solutions for this domain. It is well known that image processing is relevant within intelligent systems. Hence, we added keywords specifically related to this field.

In order to organize this research and manage the classification of the articles, we used Mendeley [1], and StArt [2], which is a tool for development of systematic reviews [96].

After defining the research questions, the search strings and the databases, the inclusion and exclusion criteria were decided. The inclusion criteria are divided in two groups: primary studies and secondary studies. For primary studies, the papers should present a use case in which DRSs was used as a possible solution, or discuss some application in such case. For secondary studies, it was enough if it brings some discussion about DRSs. Any articles published in journals, conferences, books, and workshops were considered while editorial notes and white papers were excluded. The exclusion criteria were defined as papers that do not bring any discussion about DRSs or do not mention a use case for DRSs. Only articles written in English and published between 1995 and 2016 were included. We emphasized more recent and up-to-date publications with this review period.

To verify the inclusion and exclusion criteria, a sample of the results was classified by two different researchers (peer review). An agreement of about 70 % of the sample was reached. The articles in the sample in which no agreement was achieved were not included in the review.

2.2 Studies Selection

The studies were selected by reading the title, abstract, keywords, and applying the inclusion and exclusion criteria presented in the previous section. Table 2 shows the number of papers selected according to the protocol presented in the previous section. In the first step, a total of 616 papers were returned from the 5 databases. From this primary set, the inclusion and exclusion criteria were applied. About 18.3% (113) of the articles were duplicated and 35.7% (220) were editorials or not related to DRSs. From the remaining, a total of 72 articles were chosen because they were directly related to the inclusion criteria.

Table 2 Number of articles selected for each database

Database	1^{st} step	2^{nd} step	3^{rd} step
Scopus	273	24	24
ACM	96	15	11
Web of Science	38	8	6
Science Direct	147	17	17
IEEE	62	8	7
Total	616	72	65

Of the 72 articles selected, 4 presented similar content and similar authors and only the most complete version was considered. Furthermore, 3 presented insufficient score according to our quality assessment (see Section 2.3). Thus, 72 papers were read, but only 65 were selected to extract information.

In addition, 13 other papers that were not returned from the databases were included in the search. Some of these studies were indicated by a specialist while others were requested in order to enrich the search with more examples of direct applications of DRSs in real problems. Only 1 out of these 13 papers was removed because it did not meet the quality assessment (see Section 2.3). In total, 85 articles were completely read, being 72 obtained through the search string and 13 indicated by a specialist. Of these, 77 (65 + 12) were chosen for the last phase. Table 3 presents all the selected papers. The articles that do not have a database specified were not obtained by the search string.

2.3 Quality Assessment

The quality assessment of the papers was constructed based on the work of [35]. As it is shown in Table 4, we defined 8 assessments, the first four about general purposes and the last four about specific issues. With the exception of QA5, the assessments can present 3 different values: 0, 1, or 2 as shown in Table 4.

A paper was evaluated by all the quality assessments presented in Table 4. We then compute a value according to the following equation:

$$\frac{QA1 + QA2 + QA3 + QA4 + QA5 + QA6 + QA7 + QA8}{4} \times 2.5. \quad (1)$$

Thus, it was ensured that the quality assessment produces a value between 0 and 10. Most articles (57) presented scores between 5 and 8, 7 papers presented scores higher than 8 and 13 papers between 4 and 5. As we defined the minimum acceptable score as 4, and 4 articles achieved score lower than the minimum, they were removed. Thus, 77 papers were kept to the next steps of the SLR.

The purpose of conducting a quality assessment is to verify whether the papers are in accordance with the objectives of the research questions, i.e., it is a way to check the consistency of the results. As Table 5 shows, QA4, QA5 and QA8 were the evaluations that obtained the lowest number of papers with maximum score, which reveals where the results differed from the expected ones. This will be better discussed in Section 3.

2.4 Data Extraction

Table 6 shows the 9 properties that have been chosen to be extracted from the selected 69 works (Section 2.2). The first 2 properties, P1 and P2, are not related to any specific research question, but the other properties are. They will be better explained throughout this section.

Table 3 Articles, in chronological order, used to extract properties

N	Year	Database	Reference	N	Year	Database	Reference
1	1995	Science Direct	[40]	40	2009	Web of Science	[63]
2	1998	ACM	[28]	41	2009	IEEE	[90]
3	2000	Science Direct	[98]	42	2009	Science Direct	[70]
4	2000	Scopus	[10]	43	2009	-	[52]
5	2003	Science Direct	[43]	44	2010	Science Direct	[14]
6	2004	Science Direct	[83]	45	2010	Scopus	[88]
7	2006	ACM	[84]	46	2010	Scopus	[72]
8	2006	ACM	[60]	47	2010	Scopus	[48]
9	2006	-	[94]	48	2010	Scopus	[20]
10	2007	Scopus	[93]	49	2010	Science Direct	[44]
11	2007	Scopus	[73]	50	2010	Science Direct	[38]
12	2007	-	[22]	51	2011	Web of Science	[24]
13	2007	Scopus	[74]	52	2011	Scopus	[50]
14	2007	IEEE	[30]	53	2012	-	[66]
15	2007	Science Direct	[9]	54	2012	-	[34]
16	2008	Scopus	[75]	55	2012	-	[33]
17	2008	Science Direct	[6]	56	2013	Scopus	[15]
18	2008	Scopus	[45]	57	2013	Web of Science	[16]
19	2008	Scopus	[59]	58	2013	Scopus	[23]
20	2008	IEEE	[99]	59	2013	ACM	[97]
21	2008	Science Direct	[21]	60	2014	-	[29]
22	2008	Science Direct	[17]	61	2014	Science Direct	[92]
23	2008	Scopus	[68]	62	2014	-	[18]
24	2008	IEEE	[80]	63	2014	ACM	[7]
25	2008	IEEE	[65]	64	2014	ACM	[86]
26	2008	IEEE	[89]	65	2014	Scopus	[64]
27	2008	Scopus	[67]	66	2015	Science Direct	[49]
28	2008	Scopus	[76]	67	2015	Scopus	[69]
29	2009	Scopus	[91]	68	2015	Web of Science	[36]
30	2009	Scopus	[27]	69	2015	ACM	[100]
31	2009	ACM	[32]	70	2015	Web of Science	[5]
32	2009	Science Direct	[39]	71	2015	Scopus	[42]
33	2009	ACM	[78]	72	2015	Web of Science	[79]
34	2009	IEEE	[95]	73	2016	Scopus	[61]
35	2009	Science Direct	[31]	74	2016	Scopus	[57]
36	2009	Science Direct	[8]	75	2016	-	[19]
37	2009	ACM	[13]	76	2017	-	[3]
38	2009	-	[37]	77	2017	-	[55]
39	2009	ACM	[11]				

Scientific domain. This property aims to identify what kind of paper is being analyzed. Different kinds of studies exist, e.g. surveys, short reports and white papers, each of them has a purpose. According to the necessity of the present work, 4 classes were created. They are called secondary study, theoretical, practical or industry related.

Secondary study: In general, a paper can be a primary work, a secondary work, a tertiary work and so on. A paper called primary work is an individual work that contributed to a systematic review; a systematic review is called a secondary work, and is composed by several primary works [46]. A systematic review about systematic reviews, i.e., a systematic review of systematic reviews, is called a tertiary work [47].

Table 4 Quality assessments of the articles adapted from [35]

Id	Evaluation	Answer	Value
QA1	Description of the problem	The authors give an explicit description of the problem	2
		The authors give a general description of the problem	1
		There is no description of the problem	0
QA2	Results of the study	The authors explicitly list the contributions of the study	2
		The authors give a general discussion about the results	1
		There is no discussion about the results	0
QA3	Limitations of the study	The authors explicitly list the limitations of the study	2
		The authors briefly discuss the limitations	1
		There is no discussion about the limitations	0
QA4	Future directions of the study	The authors provide a list of future works	2
		The authors briefly discuss the future works	1
		The is no discussion about the future works	0
QA5	Scope of the study	Studies where DRSs are used as a possible practical application	2
		Architecture or design for DRSs	1.5
		Approaches for simulating and testing DRSs	1
		Studies which DRSs are indirectly cited	0
QA6	About the model presented in the study	High level modeling	2
		Low level modeling	1
		No model is presented	0
QA7	Evaluation/validation presented in the study	The study presents tests using hardware devices	2
		Only simulation is used	1
		No evaluation/validation is done	0
QA8	If the study is linked to a bigger project	Yes, the study is part of a bigger project	2
		No, but presents discussions on other related projects	1
		No, the study is just a research paper	0

Table 5 Number of articles for each of the specified quality assessments

Value	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8
2	68	66	45	10	19	41	39	13
1.5					43			
1	9	10	29	50	12	30	20	14
0	0	1	3	17	3	6	18	50

Here, a paper classified as a secondary work is not only a SLR, but also any review or survey that is composed by several primary studies and aims to summarize and identify current developments of a given research area. Furthermore, if a paper is not considered a secondary study, it is a primary one and should be classified among the classes below.

Industry related: if a work is not a secondary one, it should be checked if it is strictly related to an industry problem. The works of this group focus on presenting a solution for a real world problem independent of its results, if are theoretical or practical ones. If the study is not clearly related to an industry problem, it should be classified in one of the two classes below.

Theoretical: this group encompasses works that do not present anything practical, i.e., pure theoretical works that discuss the subject of interest. It can be a paper presenting discussions about a new concept, a new definition or the future trends. Besides that, this group also encompasses studies that present systems within the scope of simulations.

Table 6 Properties to be extracted from the selected articles

Id	Property	Research question
P1	Scientific domain	Overview
P2	Completeness	Overview
P3	Definition or taxonomy	RQ1
P4	Field of knowledge	RQ1
P5	Control unit	RQ2
P6	Device used	RQ2
P7	Use case	RQ3
P8	Results obtained	RQ3
P9	Country of authors	RQ4

Practical: this group should include works that present a new methodology and, in order to test it, present the results from practical workbench tests. If the work only presents simulations and does not go deep into other tests, it should be classified as a theoretical work.

Completeness. A paper can be incomplete and, in some cases, this incompleteness creates trends in results. Here, an article is classified as incomplete if its authors explicitly say that the work is in progress. If a paper is incomplete, this property must extract the future work defined by the authors.

Definition or taxonomy. This property seeks to extract any definition, concept or taxonomy related to DRSs presented in the paper. The first research question (see Table 1) is directly related to this property. The combination of the concepts present in all papers is the basis to answer this research question.

Field of knowledge. This property should extract the main goal of the study. It could be, for example, presenting a methodology, discussing future trends, presenting a new concept, etc.

Control unit. The control unit of reconfiguration is associated to how reconfiguration occurs. It can be automatically or manually, static or dynamic, local or remote, or some other way that does not fit into any of the previous groups.

Device used. This property aims to know which physical device was used in the paper. For example, if it is a microcontroller, which is its family. If it is an ordinary computer, which is its configuration. If the article is a purely theoretical work or a secondary study, this property does not apply.

Use case. This property is associated to the 3^{rd} research question which seeks identifying the domain of application of DRSs. If the work presents a use case or a solution to a problem, this property should extract in which domain it is. Otherwise, it does not apply.

Results obtained. If the system was tested by a simulation or a device, this property should extract the results found. Although the results may be positive and negative, only the positive ones should be extracted. Example of positive results are power saving, better performance, resource saving, etc. It is important to remember that a work can present positive and negative results. The idea of this property is to understand the purpose of the authors' work, i.e. what is the goal of their methodology.

Authors' country. This property is responsible for extracting the actual country where the author works. If two authors are from the same country we consider the country only once.

2.5 Threats to Validity

One important phases of a systematic literature review is to perform a self-assessment, highlighting its most critical points. In this section we discuss some threats to validity of this work.

Search string. The initial set of studies was obtained through a search string. Although we have tried to minimize any trend that the string may generate and maximize the number of relevant studies of the research topic, it is not possible to completely eliminate these problems. The criterion to evaluate the search string in this work does not follow any strict metric, as presented in [56]. However, we can state that the results returned by the string are significant and they answer, in a very appropriate way, the research questions.

Studies selection. As defined in Section 2.2, the papers were chosen by reading title, abstract, and keywords. If the information was not present there, the work was eliminated. This might have eliminated relevant works for the research. A possible solution to this problem is to include the reading of the introduction and the conclusion if the information is not present in the abstract, as presented in [56].

3 Results and Analysis

In this section, we present the results obtained through the steps presented in the previous sections and we answer the research questions based on such results. The result for P1, "scientific domain", is as follows: 23.4% (18) of the works are theoretical; 46.7% (36) are practical works; 20.8% (16) were classified as industry related; 9.1% (7) are secondary studies. Regarding the results of P2, 3.9% (3) were classified as incomplete. Of these, one of them presents only the full reconfiguration of the system and states that the partial reconfiguration will be done as future work [36]. Another work presents only the static version of the reconfiguration, stating that the next step is the dynamic reconfiguration [32]. And, in [33], a model for a reconfigurable UAV that is able to switch between helicopter and aircraft is presented, but the work is still in progress and more simulations must be done.

In the following subsections 3.1, 3.2, 3.3 and 3.4, the results obtained for each of the properties specifically related to research questions are presented and discussed. At the end of each subsection, a partial conclusion of the results is also presented.

3.1 RQ1

The first research question is "How the different kinds of DRSs are classified by literature and what is their definitions?". The main goal of this question is to understand what are DRSs according to the literature. The properties extracted to answer this question are P3 and P4. First, the results are presented and discussed, and, after, a conclusion is drawn.

In Figure 3 we can see the results from the classification of the papers according to the property "definition", the P3. As it can be seen, more than half of the works (36) presents reconfigurable systems as FPGAs, i.e. the discussion in these articles directly associates the reconfiguration in a system with the reconfiguration of the logical units in an FPGA.

The class "self-adaptation" ranks second with 13 articles and includes theoretical papers, that define concepts about self-adaptive systems, and practical papers, that seek to increase system flexibility. Note that the definition of self-adaptive systems is associated with the behavior of a system, e.g., self-healing, self-optimizing, or self-configuring [49]. Despite the concept of self-adaptation is similar to the concept of reconfigurable system, it describes a behavior. It means that a reconfigurable system not necessarily is self-adaptive, but can be. For example, several works present the concept of reconfigurable UAV, but only a part of them can be considered self-adaptive [3,33,34]. The others can not be considered self-adaptive because they do not perform the reconfiguration authonomously [55, 18]. The difference between reconfigurable systems and auto-adaptive systems is also discussed in [54].

Examples of studies within the class "self-adaptation" are presented as follows. In [49], the authors present a taxonomy of self-adaptive systems and several concepts about them, which makes it a theoretical paper. The practical works, on the other hand, present methodologies for adaptation in different systems, such as automotive [6,86,97] and cardiac monitoring [100,42]. In [97], the authors present a way to perform system adaptation at runtime in AUTOSAR, which is an architecture for automotive system. In [6], a control system for automotive functions is discussed. The aim of the work is to provide a dynamic middleware capable of changing its policies and the policies of applications that operate over it. In [86], a new adaptive driver assistance system is presented. In this case, the driver assistance should adapt to the connection of a truck or a trailer to the car which causes the insertion of new electronic devices into the system. On health monitoring studies, Zong et al. present a dynamic reconfigurable system that exchange power consumption for performance during the measurement of cardiac monitoring [100]. And Kay and Iaione present a Field Programmable Analog Array (FPAA) that is configured by a smartphone and accomplishes acquisitions of electrocardiograms signals in real time [42].

The third place is occupied by the class "tasks", with 12 articles. Initially, we thought to create an intermediate class between "FPGAs" and "tasks" which

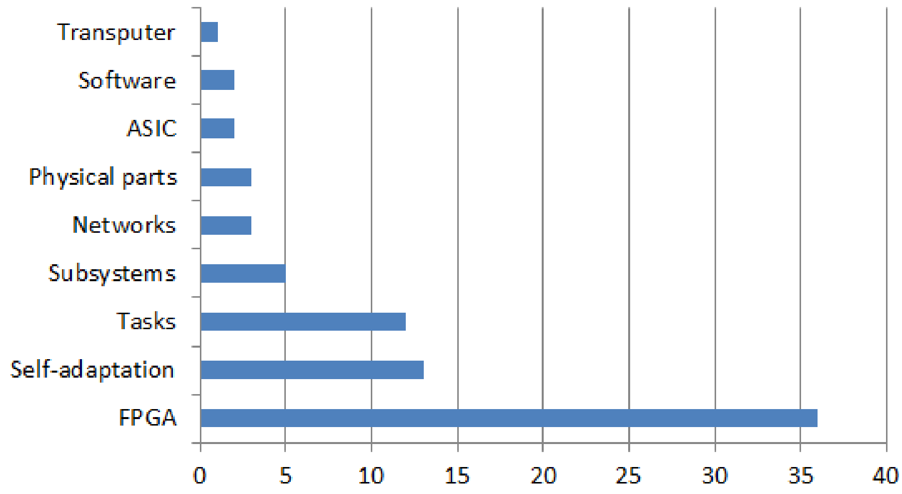

Fig. 3 Results of the classification of the papers according to P3

would represent works that define the reconfiguration as the change of tasks, similar to what an operational system does using RAM, but with FPGAs. An example of this class is the work [67] which presents a framework capable of relocating tasks between a processor (software) and an FPGA (hardware). However, we decided to keep only the class "tasks" and to include all the other works as belonging to it. The papers within this class present reconfiguration as the ability to swap between tasks. A task can be as simple as a function in a computing language, or more complex as an abstract concept that is composed by several subtasks. For example, in [94] a reconfigurable path planning for UAV is presented. The reconfigurability lies on selecting the best strategy during the flight of the UAV according to the circumstances. In this case, each strategy can be seen as a task.

The class "subsystems", ranked in fourth, includes works that sees reconfiguration as the capability of a system to change its subsystems to achieve a specific goal. Then, in 5^{th} place, the class "networks" encompasses works related to networks and communication.

The remaining classes are divided as follows: three works present reconfigurable UAVs whose parts can be arranged around a single platform to configure different types of UAVs [18, 19, 55]; two works are strictly related to reconfiguration with ASICs and two works discuss software reconfiguration; and, finally, one paper presents the reconfiguration using an already obsolete processor called transputer [40].

Based on Figure 3, we can see that DRSs are directly linked to FPGAs. This fact can be explained by the recent enhancement of such devices allowing dynamic and partial reconfiguration, e.g. Xilinx Spartan 3 was released in 2003, Virtex-4 in 2005, and Virtex-5 in 2008 [54]. Thus, the dynamic and partial reconfiguration allows changing only a part of the device while the other parts can keep its normal process. According to [54], FPGA is what we have today as the best example of a reconfigurable system but, as the author also explains, it is not the only way to reconfigure a system. This point is also shared in our analysis. As the results indicate, the use of ASICs, removable UAV parts or even a transputer can also compose the set of tools/devices able to make systems reconfigurable.

Figure 4 shows the results of the classification according to P4, which aims to define the main goal of each study. The classes defined for this property are very specific and consequently numerous. We have done it with the purpose of highlighting the characteristics of each work which would not be possible using generic classes.

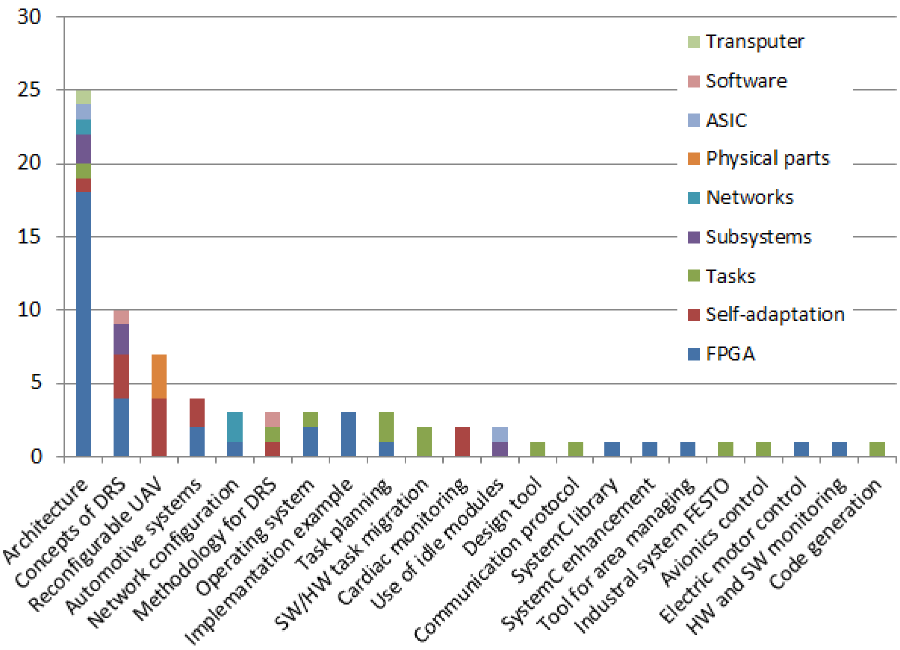

Fig. 4 Results of the classification of the papers according to P4

Despite the huge number of classes, the studies share a common feature which is the discussion about ways, approaches or attempts to make a system reconfigurable. For example, in [11] the authors present a system control for reconfiguration in the context of Integrated Modular Avionics (IMA) platform that aims to improve the reliability of the aircraft. In a different context, a multi-agent architecture is applied to an industrial system to control the reconfiguration among possible scenarios [44]. In another context, [52] presents a low cost autopilot for UAVs that can be easily reconfigured and adapted to other contexts without the need of reprogramming the entire system.

The class "architecture", with 24 studies, presents reconfiguration control architectures for specific platforms, such as FPGAs. Most of the works in this class (18 out of 24) are related to FPGAs, i.e. they present architectures for reconfiguration using FPGAs. One interesting thing to note is that, if all papers related to FPGA were removed, the graph would be close to a straight line, revealing that most part of the research on DRSs is associated to the use of FPGAs. Otherwise, there is not a main driver of research but different approaches in different contexts.

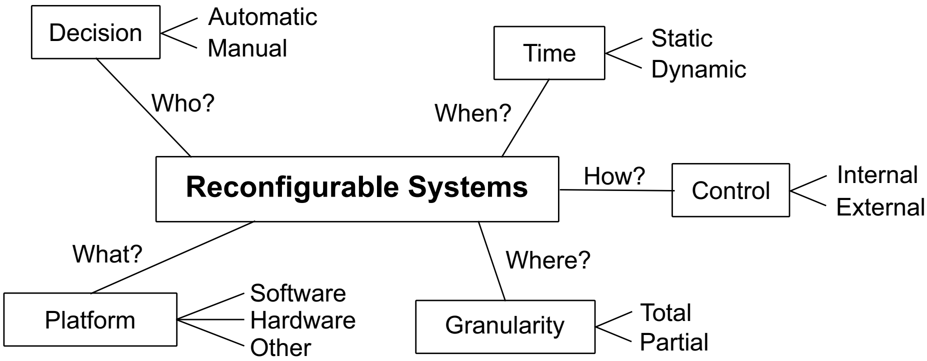

Fig. 5 Proposed classification of reconfigurable systems

We noted different kinds of DRSs when reading the papers. From the differences of these systems, we propose a new classification for DRSs as presented in Figure 5. Our new classification relies on several studies. For example, [40] defines three possible types of reconfigurations: static, dynamic, and semi-dynamic. Since static reconfiguration is performed prior to the execution of a program and dynamic reconfiguration occurs at run time whenever there is a need, semi-dynamic reconfiguration was defined by a reconfiguration that can only occur in rigid intervals of time during the execution of a program. This is the only paper that talks about "semi-dynamic" reconfiguration. However, static and dynamic reconfigurations are common expressions, especially when the subject is reconfiguration of FPGAs [43,44]. Instead of static and dynamic, the terms "offline" and "online" are also used, respectively, to describe the same types of reconfiguration [18]. In addition, "partial" reconfiguration is also common when discussing reconfiguration of FPGAs. Partial reconfiguration has emerged as an alternative to full reconfiguration, and its main idea is to reconfigure only a few internal FPGA modules, while the others remain unchanged, saving reconfiguration time.

Another concept discussed in the articles is about the platform used for reconfiguration. The majority of papers discusses hardware reconfiguration, but there are some papers that present software reconfiguration. For example, [13] defines dynamic reconfiguration as "adaptation mechanisms are actually handled by modifying the software applications configuration" and [22] presents an architecture for software reconfiguration. In addition, two papers present the task migration between hardware and software [31,67] which involves both hardware and software platforms.

The last two features are related to the works [18,55], which present a demountable UAV that can be adapted to configure fixed wings or rotors, and to [19], which presents an strategy for designing a family of UAVs with shared modules. In these papers, the reconfiguration requires a person to assemble or disassemble their parts, i.e., the decision of when it should be done and the reconfiguration itself are necessarily manual. Although these are the only works that present this features, it is important to show that the other works present automatic systems, which means that the system is responsible not only for the reconfiguration, but also for the decision making. In addition, the control system responsible for the reconfiguration can be located anywhere, like a ground station that controls a satellite.

We believe that our taxonomy is more complete than others in the sense that we present the question "who?", which leads to the decision process of reconfigurability and to the two possible answers: manual or automatic. Unlike in the context of reconfigurable systems, an adaptive system cannot be manual, and the question "who?" does not make sense, as presented in [18].

Conclusion. Based on all the papers and the remarks we have just made, we propose a new definition for a DRS as shown below.

Definition 1. A DSR is a system whose subsystems can be modified or have their settings modified during operation to achieve a specific goal.

We believe that this definition is general enough and it does not specify from what a system must be made up of, or how its reconfiguration must happen, or how many times it can be done. In [54], for example, the authors present a definition and say that reconfiguration is not infinitely repetitive neither instantaneous. Furthermore, we totally agree with [54] when it presents configurability as a design choice, and we use this concept to help defining reconfigurable systems, saying that a reconfigurable system is a design choice.

3.2 RQ2

The second research question is "What are the hardware and software platforms, methodologies and techniques engaged in DRS?". The focus of this question is to understand which devices are used in DRSs and how their reconfiguration is managed. The results of properties P5 and P6 are presented and discussed in this subsection and, in the end, a conclusion is drawn.

Property P5, control unit, is strictly linked to the taxonomy and classification of DRSs. As shown in Figure 5, 5 different types of control units were identified during the reading of the studies. With the exception of the results about granularity, the others are presented below.

The results related to time and decision making are strictly related. Of the 77 studies, in 8 of them no such characteristics were identified. In addition, 64 papers present automatic approaches and 62 dynamic. The only two works that present automatic and static systems are FPGA systems whose reconfiguration occurs only once. One of them was classified as incomplete and presents as future work the dynamic version [32]. The 5 works which reconfiguration is manual and static are related do reconfigurable UAVs [34, 66, 18, 19, 55].

The results about subsystems were classified according to the type of device used. Of the 77 studies, 8 had its characteristics unclear and were not classified, 39 present systems developed to hardware devices, 11 to both hardware and software, and 11 to software. Of the remaining studies, 5 of them introduce the idea of UAV with interchangeable modules, 2 of them present hybrid UAVs capable of morphing from helicopter to aircraft, and, finally, one last work present reconfigurable network protocols.

It is important to emphasize the difference between software reconfiguration and hardware reconfiguration. The first one defines a system designed in a high level language that changes its own routines to achieve a goal and runs on general purpose processors. For example, [94] introduces the idea of a reconfigurable path planning for a UAV, that chooses a strategy of flying mode among several strategies given the current conditions of the UAV. The system runs on three PC104 embedded devices. On the other hand, hardware reconfiguration involves low level programming and reconfigurable hardware, such as an FPGA. The problems solved in this kind of systems tend to be simple due to its high complexity during development. Besides that, there is a third approach when software and hardware reconfiguration are mixed, e.g., [31, 67] present the idea of task migration between general purpose processor and reconfigurable hardware.

Figure 6 shows the results about the control unit. In this work, we call control unit the subsystem responsible for reconfiguring the system. Without the 8 works whose characteristics could not be extracted, the vast majority of the works presents reconfiguration internal to the device, i.e., presents a system that does not depend on an external system.

A second part of the studies present systems composed by more than one device, that can be FPGAs, sensors, microprocessors or even a mix of them, whose representation in Figure 4 occurs in 3 different groups: devices, FPGAs, and sensors. The idea of these works is to present DRSs capable of managing the resources of different devices. With one exception, we considered the reconfiguration of these groups as external to the system. In some cases, it is easy to define if the control unit is part of the system, as when a ground station controls a satellite, but in some cases it is not. We wonder that the devices plugged to the system are used as resources, but the control unit runs in another device external to them.

The class "physical pieces", with 5 papers, represents the reconfiguration that occurs in physical pieces, such as flexible wings and replaceable rotors. In this class, the reconfiguration is controlled by another device, and is clearly external. Another reconfiguration strategy is to use a processor attached to some device, such as an FPGA. In this case, the reconfiguration is also external. Finally, 3 works present reconfiguration in network communication, which is controlled by the same system responsible for the communication and the reconfiguration. The other works present reconfiguration using multiple devices.

As this review is related to systems with dynamic reconfiguration, this explains the small number of static reconfiguration approaches in the results. In addition, we can also say that dynamic systems are also automatic, which does not mean that there are no manual dynamic systems, but they have not been detected in this work. The large number of systems whose reconfiguration is hardware-related can be explained by the large number of articles that present strategies for reconfiguration in FPGAs. Finally, we can state that, in DRSs, the reconfiguration methodologies are essentially based on a control unit. This unit can be part of the device where the reconfiguration itself is made or be in an external device.

The results of P6 can be seen in Figure 7. If we count the number of devices in each group we found 84, exceeding the number of selected papers. It happens because, in some studies, more than one device is used. Without the theoretical works, whose properties do not apply, most of the devices used are FPGAs. In addition, they are Xilinx FPGAs, such as the Virtex family. In less quantity, we can found different types of processors, e.g. PowerPC and ARM processors. There is also a paper that presents a system based on Rasberry Pi.

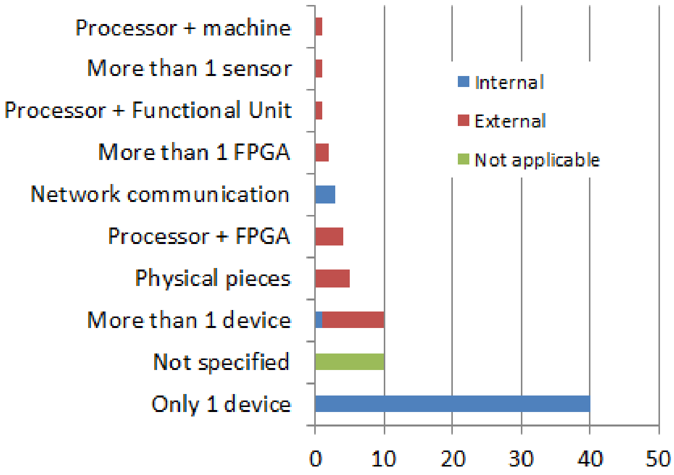

Fig. 6 Results of the classification of the papers according to P5

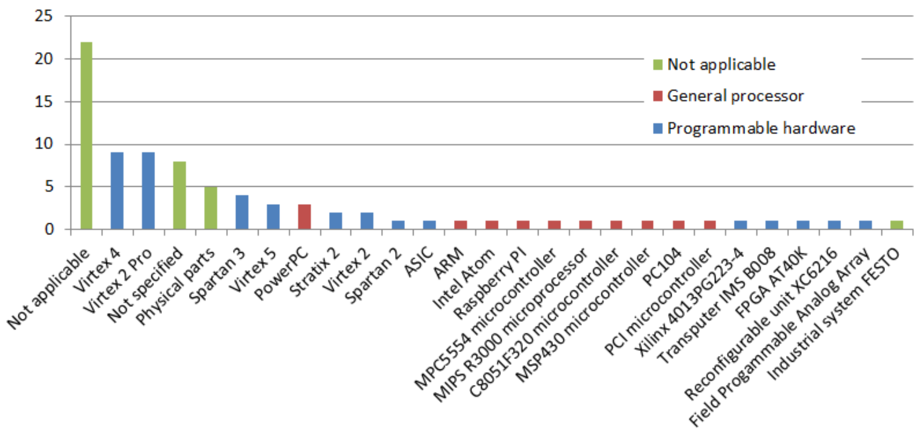

Fig. 7 Results of the classification of the papers according to P6

As discussed in Section 3.1, the majority of studies uses FPGAs as reconfigurable systems. On the other hand, the number of devices that are not FPGAs is proportional to their diversity, making the graph similar to a straight line. It means that there is not a specific device used in the researches.

Conclusion. In this section, we presented and discussed methodologies and devices used for reconfigurable systems. Specific questions about reconfiguration methodologies have not been made, e.g., what kind of algorithms are used in task planning, what techniques are available to reconfigure an FPGA, or how reconfiguration in software can be performed. Although they are important questions, they would lead to very specific responses and more general questions were answered. With the exception of the works about reconfigurable UAVs, we can state that DRSs are automatic and based on either hardware or software. Furthermore, every DRS is based on a control unit, which is responsible for the decision making

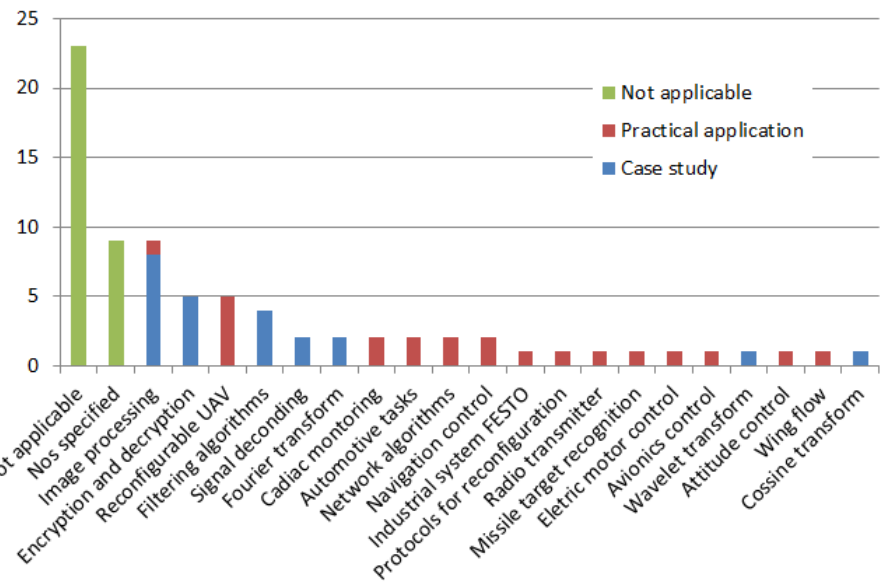

Fig. 8 Results of the classification of the papers according to P7

and to perform the reconfiguration of the system. This unit can be seen as a system that manages another system, and can be external or internal to it. In addition, it can involve several devices, homogeneous or not. In short, the concept of reconfiguration is valid for a large number of different systems, but its implementation is unique and must adapt to the features of the platform.

3.3 RQ3

The third research question is "What are the domains of application of DRSs?". In order to answer it, properties P7 and P8 are discussed in this section.

Figure 8 shows the use cases and application examples extracted from the studies. Although some papers present more than one use case, only one was extracted per work. In total, 9 papers claim to have conducted tests, but they do not specify how they were done or what they involved. In addition, in 23 works the property was not extracted because the articles were conceptual and did not bring any use case. The difference of 1 paper when compared to P6 (see Figure 7) is explained by the study whose verification was done through simulations in an FPGA Virtex 4 [72].

As explained before, there is a good number of studies related to FPGA and, consequently, a good amount of use cases related to it. In general, FPGAs are used for simple, repetitive, and easily parallelizable tasks, such as signal and image processing. In these cases, the application domain includes space applications [63], automotive tasks [20], network communication [39] and others [74,98]. Besides the studies involving FPGAs, the fields of application include avionics [11], industrial pneumatic system [44], reconfigurable UAVs [18,19,34,66,55], navigation control for UAVs [52,94], cardiac monitoring [42,100] and automotive tasks [6,

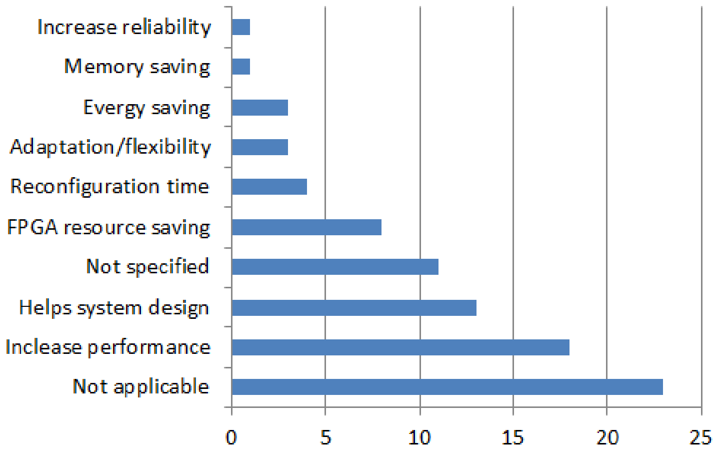

Fig. 9 Results of the classification of the papers according to P8

20,86,97]. Below we discuss some works in more detail. In [8], the authors present a control system embedded in an FPGA that is able to, during runtime, change its configurations. Claus and Stechele present an architecture for video processing during different drive conditions. The main purpose of the work is to reconfigure an embedded hardware with different algorithms depending on the context that the automobile is [20]. Kachris et al. present a system able to dynamically change its modules to better adapt to different requests received by a network [39].

Figure 9 presents the positive results achieved by the studies. The negative results were not considered because they are presented only in a few works, a characteristic also verified in [56]. In addition, sometimes a paper has more than one positive result, so, adding all numbers in the graph, the result is 85, instead of 77 (number of papers). Without considering the first group, which property is not applicable due to theoretical and conceptual papers, the largest group, with 18 papers, shows gains in performance, which includes less resource consumption. Thereafter, 13 works present use cases whose result helps designing systems. The need for area saving is typically a feature of FPGAs and 8 papers claim to decrease the use of FPGA resources through reconfiguration. In addition, decreasing reconfiguration time and saving energy also appear as relevant results.

The class "adaptation" is ranked among the last four positions. Actually, only three article claimed an improvement in adaptiveness. Of these, 2 works present UAVs that could be mounted with fixed wings or rotors, using the best option depending on the context [18,55] and one presents a family of reconfigurable UAVs that can be optimized by customizing the components for different missions [66].

As can be seen in Figure 9, the main focus of the results is increase performance and decrease resources needed. In addition, the results also presents tools and architectures to assist in designing DRSs, which is a fact discussed in this area, since designing such systems is a complex activity. On the other hand, there is a lack of papers that discuss gains such as flexibility, fault tolerance, and reliability.

Conclusion. The large number of FPGA-related studies may have created a trend in the results. Consequently, the results did not present application areas

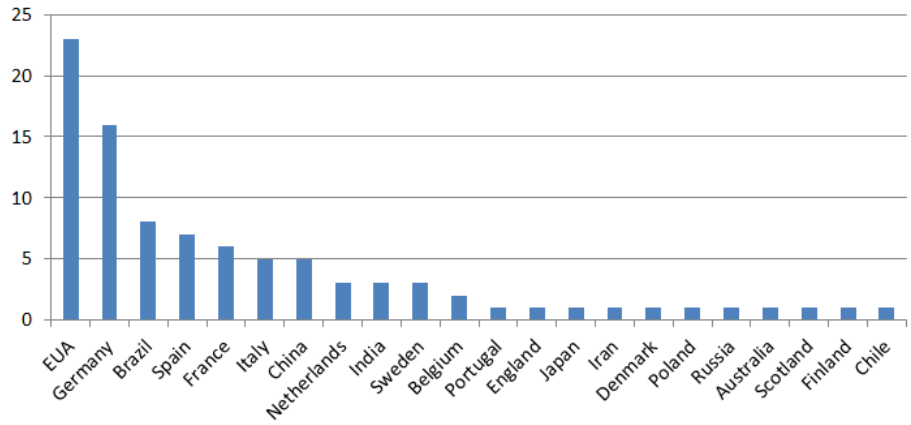

Fig. 10 Results of the classification of the papers according to P9

linked to specific problem solving, but rather to generic areas such as image and signal processing. This gap has already been reported in the quality assessment (see Table 5) and can be compared with the results of Figure 9 which show little concern for adaptability or reliability, but rather for resource management.

3.4 RQ4

The fourth and last research question is "Which countries lead the number of publications in DRS?". To answer this question, the property P9 was extracted from the studies.

Figure 10 shows the results of the property P9 by country and Figure 11 by continent. As we can see, the two countries leading the number of publications are USA and Germany. In a second group we can see Brazil, Spain, France, Italy, and China. The others countries have, approximately, the same small number of publications. Besides that, Europe lead the number of publications with 46 different articles, which is explained by its high number of countries and not only by the number of publications. For example, while the European continent has 48 papers distributed over 13 countries, America has 32 works distributed over 3 countries. The Asian continent is ranked in third with 11 articles, and Oceania is in fourth with only 1 paper. The African continent has zero publications.

Although it is difficult to explain the distribution presented in the Figures 10 and 11, we discuss below some trends involving the two countries that leads the publications, EUA and Germany. The most cited devices in this review are related to two companies of logic programmable devices, Altera and Xilinx (see Fig. 7), both founded in Sillicon Valley, EUA. Maybe improving these devices is a strong research point in EUA. On the other hand, all papers found in this systematic review related to automobile issues have, at least, one German author, which suggests an effort in applying reconfiguration in this area [6, 20, 86, 97].

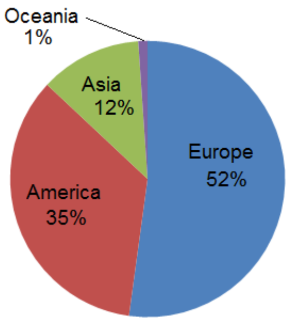

Fig. 11 Results of the classification of the papers according to P9 for each continent

4 Conclusions

This work presented a systematic literature review of DRSs which involved studying their concepts and classifications, domains of application and methodologies engaged in their construction. In total, 85 articles published between 1995 and 2016 were read and 77 were selected to extract properties. We believe that, at the end of this SLR and through the answers obtained for the proposed research questions, a clear and broad view on DRSs was conceived.

The main findings are presented below.

Concept of DRS. Based on the concepts obtained during the read of the studies, we proposed the definition that a DRS is a system whose subsystems can be modified or have their configurations changed during operation to reach a certain goal. This definition is similar to the definition proposed by others papers [54,37] and also encompasses the concept of reconfigurable computing, a term proposed in 1960 to describe electronic systems able to modify their internal structure in order to increase performance when processing specific tasks [25]. Also, as a result of this work, a new taxonomy for DRSs was proposed.

DRSs as digital systems. Although [54] proposes a taxonomy for reconfigurable systems that encompasses analogue and digital architectures, in this review, only digital systems were addressed without considering the work of the reconfigurable UAV that does not has dynamic reconfiguration. It means that the strategies in the domain of DRS are essentially based on software or hardware, being most part of the works based on hardware.

Research involving FPGA. More than half of the papers (52.17%) presented studies related to FPGA. We believe that, due to its recent evolution and commercialization, the research involving such devices has grown. The main issues in this area are not binded to a bigger project to solve industry-related problems, but to the exploitation of the resources of the devices themselves. For example, some works look for strategies to reduce the reconfiguration time of the FPGA, proposing different architectures to do it [90,60,43]. Although several works have demonstrated the power that an FPGA can achieve [29], designing systems for these devices is very arduous and there is still the necessity of skilled workers engaged in these projects.

Small number of studies related to industry. What we here called industrial use cases are works that present solutions to real problems, such as the reconfigurable UAV [18, 34, 55, 66] or the steering assistant for cars [86]. As system adaptation is a growing need, we expected to find more works related to it.

Based on the results obtained in this SLR, we propose the **research agenda** below for the development of DRSs. This agenda includes some unanswered questions in this review that can guide future efforts in this area. The most important points are listed below:

1. To treat the research area of reconfigurable computing, which includes mainly studies with FPGAs, as an independent domain from DRSs. Reconfigurable computing is an area that proposes data processing architectures capable of modifying themselves in order to create specific processing units [25]. Compared to other architectures, reconfigurable computing is located between Von Neumann architecture and ASICs, the former having high flexibility, but low performance, and the second low flexibility, but high performance [12]. Unlike the general purpose processors, which are static, the goal of reconfigurable computing is to be dynamic adapting to the applications needs. In addition, reconfigurable computing can be seen as part of DRSs domain, which leaves us to propose reconfigurable computing as an independent subarea of DRSs. Thus, we believe that treating them individually helps to better define their concepts and goals. While in reconfigurable computing the architectures are discussed at bit level and the aim is to save memory or to improve performance, in adaptive systems the level of the abstraction is higher and the goal is to provide to an entire complex system a certain kind of intelligence. Finally, understanding these two domains as separate areas is important to define new research questions of future literature reviews;
2. To explore methodologies used in DRSs and FPGAs, i.e., define the metrics used for decision making, how decision making occurs and on what it is based, for each of the areas proposed in the previous item;
3. To study the difficulties founded during the implementation of DRSs, the existing solutions and the challenges that still exist;
4. To stimulate the development of methodologies and techniques for DRSs, and their application to a significantly greater number of industrial or research case studies. Specifically, research and development of DRSs into aerospace context, focusing on aspects such as self-adaptation, is of great value, since aerospace systems are subject to operating environments whose characteristics can often change in an unpredictable way;
5. To investigate the use of other hardware platforms besides FPGA. As mentioned in this SLR, we can state that FPGA dominates the context of DRSs. It would be interesting to investigate other hardware platforms, such as microcontrollers and general purpose processors, for the development of DRSs.

References

1. Mendeley. Available at (https://www.mendeley.com (2017/08/15)
2. Start. Available at (http://lapes.dc.ufscar.br/tools/start_tool (2017/08/15)
3. Abnous, R., Zeng, C., Chowdhury, S.: Dynamics and control design of a blended wing-body transitioning UAV. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2017 (June), 5–9 (2017). DOI doi:10.2514/6.2017-4150

4. Aksit, M., Choukair, Z.: Dynamic, adaptive and reconfigurable systems overview and prospective vision. In: 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., pp. 84–89. IEEE (2003). DOI 10.1109/ICDCSW.2003.1203537
5. Ansh, Yadav, R., Deb, D., Ray, A.: Multi-purpose auto-programmable reconfigurable embedded system architecture. In: 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), pp. 48–51. IEEE (2015). DOI 10.1109/EDCAV.2015.7060537
6. Anthony, R., Rettberg, A., Chen, D., Jahnich, I., de Boer, G., Ekelin, C.: A Dynamically Reconfigurable Automotive Control System Architecture. IFAC Proceedings Volumes **41**(2), 9308–9313 (2008). DOI 10.3182/20080706-5-KR-1001.01573
7. Ashjaei, M., Pedreiras, P., Behnam, M., Almeida, L., Nolte, T.: Dynamic reconfiguration in multi-hop switched ethernet networks. ACM SIGBED Review **11**(3), 62–65 (2014). DOI 10.1145/2692385.2692397
8. Astarloa, A., Lázaro, J., Bidarte, U., Jiménez, J., Zuloaga, A.: FPGA technology for multi-axis control systems. Mechatronics **19**(2), 258–268 (2009). DOI 10.1016/j.mechatronics.2008.09.001
9. Astarloa, A., Zuloaga, A., Bidarte, U., Martín, J.L., Lázaro, J., Jiménez, J.: Tornado: A self-reconfiguration control system for core-based multiprocessor CSoPCs. Journal of Systems Architecture **53**(9), 629–643 (2007). DOI 10.1016/j.sysarc.2007.01.011
10. Bapty, T., Neema, S., Scott, J., Sztipanovits, J., Asaad, S.: Model-integrated Tools for the Design of Dynamically Reconfigurable Systems. VLSI Design **10**(3), 281–306 (2000). DOI 10.1155/2000/74708
11. Bieber, P., Noulard, E., Pagetti, C., Planche, T., Vialard, F.: Preliminary Design of Future Reconfigurable IMA Platforms. SIGBED Rev. **6**(3), 7:1—-7:5 (2009). DOI 10.1145/1851340.1851349
12. Bobda, C.: Introduction to Reconfigurable Computing. Springer Netherlands, Dordrecht (2007). DOI 10.1007/978-1-4020-6100-4
13. Borde, E., Feiler, P.H., Haïk, G., Pautet, L.: Model driven code generation for critical and adaptative embedded systems. ACM SIGBED Review **6**(3), 1 (2009). DOI 10.1145/1851340.1851352
14. Brunelli, C., Garzia, F., Rossi, D., Nurmi, J.: A coarse-grain reconfigurable architecture for multimedia applications supporting subword and floating-point calculations. Journal of Systems Architecture **56**(1), 38–47 (2010). DOI 10.1016/j.sysarc.2009.11.003
15. Cancare, F., Pilato, C., Cazzaniga, A., Sciuto, D., Santambrogio, M.D.: D-RECS: A complete methodology to implement Self Dynamic Reconfigurable FPGA-based systems. In: 2013 8th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), pp. 1–6. IEEE (2013). DOI 10.1109/ReCoSoC.2013.6581550
16. Cantó, E., Fons, M., Fons, F., López, M., Ramos, R.: Fast self-reconfigurable embedded system on Spartan-3. Journal of Universal Computer Science **19**(3), 301–324 (2013). DOI 10.3217/jucs-019-03-0301
17. Cho, Y.H.: Automatic Target Recognition Systems on Reconfigurable Devices. In: Reconfigurable Computing, pp. 591–612. Elsevier (2008). DOI 10.1016/B978-012370522-8.50037-6
18. Chowdhury, S., Maldonado, V., Patel, R.: Conceptual design of a multi-ability reconfigurable unmanned aerial vehicle (UAV) through a synergy of 3D CAD and modular platform planning. In: AIAA AVIATION 2014 - 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, June, pp. 16–20 (2014). DOI 10.2514/6.2014-2178
19. Chowdhury, S., Maldonado, V., Tong, W., Messac, A.: New Modular Product-Platform-Planning Approach to Design Macroscale Reconfigurable Unmanned Aerial Vehicles. Journal of Aircraft **53**(2), 309–322 (2016). DOI 10.2514/1.C033262
20. Claus, C., Stechele, W.: AutoVisionReconfigurable Hardware Acceleration for Video-Based Driver Assistance. In: M. Platzner, J. Teich, N. Wehn (eds.) Dynamically Reconfigurable Systems, pp. 375–394. Springer Netherlands, Dordrecht (2010). DOI 10.1007/978-90-481-3485-4_18
21. Compton, K., DeHon, A.: Operating System Support for Reconfigurable Computing. Reconfigurable Computing pp. 231–254 (2008). DOI 10.1016/B978-012370522-8.50016-9
22. Costa, C., Ali, N., Perez, J., Carsi, J.A., Ramos, I.: Dynamic reconfiguration of software architectures through aspects. Lecture Notes in Computer Science **4758**, 279–283 (2007). DOI 10.1007/978-3-540-75132-8_24

23. Diessel, O.: Opportunities and challenges for dynamic FPGA reconfiguration in electronic measurement and instrumentation. Proceedings of 2013 IEEE 11th International Conference on Electronic Measurement and Instruments, ICEMI 2013 **1**, 258–263 (2013). DOI 10.1109/ICEMI.2013.6743028
24. Diguet, J.P., Eustache, Y., Gogniat, G.: Closed-loop–based Self-adaptive Hardware/Software-Embedded Systems: Design Methodology and Smart Cam Case Study. ACM Trans. Embed. Comput. Syst. **10**(3), 38:1—-38:28 (2011). DOI 10.1145/1952522.1952531
25. Estrin, G.: Reconfigurable computer origins: the UCLA fixed-plus-variable (F+V) structure computer. IEEE Annals of the History of Computing **24**(4), 3–9 (2002). DOI 10.1109/MAHC.2002.1114865
26. Ferguson, S., Siddiqi, A., Lewis, K., de Weck, O.L.: Flexible and reconfigurable systems: Nomenclature and review. In: International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (2007)
27. Ferreira, R., Laure, M., Beck, A.C., Lo, T., Rutzig, M., Carro, L.: A low cost and adaptable routing network for reconfigurable systems. IPDPS 2009 - Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium (2009). DOI 10.1109/IPDPS.2009.5161217
28. Fleischmann, J., Buchenrieder, K., Kress, R.: A hardware/software prototyping environment for dynamically reconfigurable embedded systems. In: Proceedings of the Sixth International Workshop on Hardware/Software Codesign. (CODES/CASHE'98), CODES/CASHE '98, pp. 105–109. IEEE Comput. Soc, Washington, DC, USA (1998). DOI 10.1109/HSC.1998.666246
29. Gadfort, P., Dasu, A., Akoglu, A., Leow, Y.K., Fritze, M.: A power efficient reconfigurable system-in-stack: 3D integration of accelerators, FPGAs, and DRAM. International System on Chip Conference pp. 11–16 (2014). DOI 10.1109/SOCC.2014.6948892
30. Gonzalez, I., Aguayo, E., Lopez-Buedo, S.: Self-Reconfigurable Embedded Systems on Low-Cost FPGAs. IEEE Micro **27**(4), 49–57 (2007). DOI 10.1109/MM.2007.72
31. Götz, M., Dittmann, F., Xie, T.: Dynamic relocation of hybrid tasks: Strategies and methodologies. Microprocessors and Microsystems **33**(1), 81–90 (2009). DOI 10.1016/j.micpro.2008.08.001
32. Harb, N., Niar, S., Khan, J., Saghir, M.A.R.: A reconfigurable platform architecture for an automotive multiple-target tracking system. ACM SIGBED Review **6**(3), 1 (2009). DOI 10.1145/1851340.1851355
33. Heredia, G., Duran, A., Ollero, A.: Modeling and simulation of the HADA reconfigurable UAV. Journal of Intelligent and Robotic Systems: Theory and Applications **65**(1-4), 115–122 (2012). DOI 10.1007/s10846-011-9561-9
34. Holland, M., Moore, B., Ferguson, S.: 'Transforming' for Stability and Aerodynamic Performance in a Reconfigurable UAV. In: 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, September, pp. 1–20. American Institute of Aeronautics and Astronautics, Reston, Virigina (2012). DOI 10.2514/6.2012-5530
35. Jamshidi, P., Ghafari, M., Ahmad, A., Pahl, C.: A Protocol for Systematic Literature Review on Architecture - Centric Software Evolution Research. Tech. rep., The Irish Software Engineering Research Centre (2012)
36. Janakiraman, N., Nirmalkumar, P., Akram, S.M.: Coarse Grained ADRES Based MIMO-OFDM Transceiver with New Radix-2^5 Pipeline FFT/IFFT Processor. Circuits, Systems, and Signal Processing **34**(3), 851–873 (2015). DOI 10.1007/s00034-014-9880-8
37. JÓZWIAK, L., NEDJAH, N.: Modern architectures for embedded reconfigurable systems a survey. Journal of Circuits, Systems and Computers **18**(02), 209–254 (2009). DOI 10.1142/S0218126609005034
38. Jóźwiak, L., Nedjah, N., Figueroa, M.: Modern development methods and tools for embedded reconfigurable systems: A survey. Integration, the VLSI Journal **43**(1), 1–33 (2010). DOI 10.1016/j.vlsi.2009.06.002
39. Kachris, C., Wong, S., Vassiliadis, S.: Design and performance evaluation of an adaptive FPGA for network applications. Microelectronics Journal **40**(7), 1103–1110 (2009). DOI 10.1016/j.mejo.2008.05.011
40. Kalinowski, T., Thor, M., Tudruj, M.: Multi-transputer systems with dynamic reconfiguration control based on the serial bus. Computing Systems in Engineering **6**(4-5), 391–400 (1995). DOI 10.1016/0956-0521(95)00031-3

41. Kanellakis, C., Nikolakopoulos, G.: Survey on Computer Vision for UAVs: Current Developments and Trends. Journal of Intelligent & Robotic Systems pp. 1–28 (2017). DOI 10.1007/s10846-017-0483-z
42. Kay, M.S., Iaione, F.: Reconfigurable Embedded System for ECG Signal Acquisition. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, V, pp. 25–26. IEEE (2015). DOI 10.1109/CBMS.2015.58
43. Kessal, L., Abel, N., Demigny, D.: Real-time image processing with dynamically reconfigurable architecture. Real-Time Imaging **9**(5), 297–313 (2003). DOI 10.1016/j.rti.2003.07.001
44. Khalgui, M., Mosbahi, O.: Intelligent distributed control systems. Information and Software Technology **52**(12), 1259–1271 (2010). DOI 10.1016/j.infsof.2010.06.001
45. Kitamichi, J., Ueda, K., Kurod, K.: A Modeling of a Dynamically Reconfigurable Processor Using SystemC. In: 21st International Conference on VLSI Design (VLSID 2008), pp. 91–96. IEEE (2008). DOI 10.1109/VLSI.2008.13
46. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University (2004)
47. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - A systematic literature review. Information and Software Technology **51**(1), 7–15 (2009). DOI 10.1016/j.infsof.2008.09.009
48. Kriesten, D., Pankalla, V., Heinkel, U.: An Application Example of a Run-Time Reconfigurable Embedded System. In: 2010 International Conference on Reconfigurable Computing and FPGAs, pp. 97–102. IEEE (2010). DOI 10.1109/ReConFig.2010.53
49. Krupitzer, C., Roth, F.M., Vansyckel, S., Schiele, G., Becker, C.: A survey on engineering approaches for self-adaptive systems. Pervasive and Mobile Computing **17**(PB), 184–206 (2015). DOI 10.1016/j.pmcj.2014.09.009
50. Kuehnle, M., Britoy, A., Roth, C., Dagas, K., Becker, J.: The study of A Dynamic Reconfiguration Manager for Systems-on-Chip. Proceedings - 2011 IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2011 pp. 13–18 (2011). DOI 10.1109/ISVLSI.2011.35
51. Li, L., Lin, C.J., Huang, M.L., Kuo, S.C., Chen, Y.R.: Mobile robot navigation control using recurrent fuzzy cerebellar model articulation controller based on improved dynamic artificial bee colony. Advances in Mechanical Engineering **8**(11), 1687814016681,234 (2016). DOI 10.1177/1687814016681234
52. Lizarraga, M.I., Elkaim, G.H., Horn, G.M., Curry, R., Dobrokhodov, V., Kaminer, I.: Low Cost Rapidly Reconfigurable UAV Autopilot for Research and Development of Guidance, Navigation and Control Algorithms. In: Volume 3: ASME/IEEE 2009 International Conference on Mechatronic and Embedded Systems and Applications; 20th Reliability, Stress Analysis, and Failure Prevention Conference, July 2015, pp. 585–594. ASME (2009). DOI 10.1115/DETC2009-86547
53. Lou, G.H., Chen, C.: Design of the intelligent control system for mine hoist. In: Mechanical Engineering, Industrial Electronics and Information Technology Applications in Industry, *Applied Mechanics and Materials*, vol. 427, pp. 957–960. Trans Tech Publications (2013). DOI 10.4028/www.scientific.net/AMM.427-429.957
54. Lyke, J.C., Christodoulou, C.G., Vera, G.A., Edwards, A.H.: An introduction to reconfigurable systems. Proceedings of the IEEE **103**(3), 291–317 (2015). DOI 10.1109/JPROC.2015.2397832
55. Maldonado, V., Sarker, P., Chowdhury, S.: A Modular Design Approach to A Reconfigurable Unmanned Aerial Vehicle. In: AIAA Information Systems-AIAA Infotech @ Aerospace, January, pp. 1–12. American Institute of Aeronautics and Astronautics, Reston, Virginia (2017). DOI 10.2514/6.2017-0224
56. Martins, L.E.G., Gorschek, T.: Requirements engineering for safety-critical systems: A systematic literature review. Information and Software Technology **75**, 71–89 (2016). DOI 10.1016/j.infsof.2016.04.002
57. McGee, E.T., McGregor, J.D.: Using dynamic adaptive systems in safety-critical domains. In: Proceedings of the 11th International Workshop on Software Engineering for Adaptive and Self-Managing Systems - SEAMS '16, pp. 115–121. ACM Press, New York, New York, USA (2016). DOI 10.1145/2897053.2897062
58. Mehra, S., Majithia, J.: Reconfigurable computer architectures. IEE Proceedings E - Computers and Digital Techniques **4**(129), 156 – 164 (1982). DOI 10.1049/IP-E:19820032
59. Mei, B., Berekovic, M., Mignolet, J.Y.: ADRES & DRESC: Architecture and Compiler for Coarse-Grain Reconfigurable Processors. In: Fine- and Coarse-Grain Reconfigurable Computing, pp. 255–297. Springer Netherlands, Dordrecht (2008). DOI 10.1007/978-1-4020-6505-7_6

60. Möller, L., Soares, R., Carvalho, E., Grehs, I., Calazans, N., Moraes, F.: Infrastructure for Dynamic Reconfigurable Systems: Choices and Trade-offs. In: Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design, SBCCI '06, pp. 44–49. ACM, New York, NY, USA (2006). DOI 10.1145/1150343.1150360
61. Nam, H., Lysecky, R.: Latency, power, and security optimization in distributed reconfigurable embedded systems. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, pp. 124–131 (2016). DOI 10.1109/IPDPSW.2016.40
62. Ogata, K.: Modern control engineering. Englewood Cliffs, N.J : Prentice-Hall (1970)
63. Osterloh, B., Michalik, H., Fiethe, B.: SoCWire: A robust and fault tolerant network-on-chip approach for a dynamic reconfigurable System-on-Chip in FPGAs. Lecture Notes in Computer Science **5455**(i), 50–59 (2009). DOI 10.1007/978-3-642-00454-4_8
64. Palumbo, F., Sau, C., Raffo, L.: Power-awarness in coarse-grained reconfigurable designs: A dataflow based strategy. In: 2014 IEEE Workshop on Signal Processing Systems (SiPS), pp. 1–6. IEEE (2014). DOI 10.1109/SiPS.2014.6986104
65. Pan, Z., Wells, B.E.: Hardware supported task scheduling on dynamically reconfigurable SoC architectures. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **16**(11), 1465–1474 (2008). DOI 10.1109/TVLSI.2008.2000974
66. Pate, D.J., Patterson, M.D., German, B.J.: Optimizing Families of Reconfigurable Aircraft for Multiple Missions. Journal of Aircraft **49**(6), 1988–2000 (2012). DOI 10.2514/1.C031667
67. Pellizzoni, R., Caccamo, M.: Hybrid hardware-software architecture for reconfigurable real-time systems. Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS pp. 273–284 (2008). DOI 10.1109/RTAS.2008.14
68. Peper, C., Schneider, D.: Component engineering for adaptive ad-hoc systems. In: Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems - SEAMS '08, SEAMS '08, p. 49. ACM Press, New York, New York, USA (2008). DOI 10.1145/1370018.1370028
69. Perera, D.G.: Analysis of FPGA-Based Reconfiguration Methods for Mobile and Embedded Applications. In: Proceedings of the 12th FPGAworld Conference 2015 on - FPGAworld '15, pp. 15–20. ACM Press, New York, New York, USA (2015). DOI 10.1145/2889287.2889297
70. Podolski, I., Rettberg, A.: Towards a Self-Configurable Middleware for Production Control Systems with Wireless Sensor Networks. IFAC Proceedings Volumes **42**(4), 414–419 (2009). DOI 10.3182/20090603-3-RU-2001.0433
71. Pu, H., Zhen, Z., Jiang, J., Wang, D.: Uav flight control system based on an intelligent bel algorithm. International Journal of Advanced Robotic Systems **10**(2), 121 (2013). DOI 10.5772/53746
72. Raffo, M., Gomes Filho, J., Strum, M., Chau, W.J.: A placement tool for a NOC-based dynamically reconfigurable system. In: 2010 VI Southern Programmable Logic Conference (SPL), pp. 47–52. IEEE (2010). DOI 10.1109/SPL.2010.5483005
73. Rana, V., Santambrogio, M.: Dynamic reconfigurability in embedded system design. Circuits and Systems, 2007 pp. 2734–2737 (2007). DOI 10.1109/ISCAS.2007.378618
74. Rodriguez-Andina, J., Moure, M., Valdes, M.: Features, Design Tools, and Application Domains of FPGAs. IEEE Transactions on Industrial Electronics **54**(4), 1810–1823 (2007). DOI 10.1109/TIE.2007.898279
75. Sanchez-Elez, M., Fernandez, M., Bagherzadeh, N., Hermida, R., Kurdahi, F., Maestre, R.: A coarse-grain dynamically reconfigurable system and compilation framework. Fine- and Coarse-Grain Reconfigurable Computing pp. 181–215 (2008). DOI 10.1007/978-1-4020-6505-7_4
76. Santambrogio, M.D., Rana, V., Sciuto, D.: Operating system support for online partial dynamic reconfiguration management. In: 2008 International Conference on Field Programmable Logic and Applications, pp. 455–458. IEEE (2008). DOI 10.1109/FPL.2008.4629982
77. Shoa, A., Shirani, S.: Run-Time Reconfigurable Systems for Digital Signal Processing Applications : A Survey. Journal of VLSI Signal Processing **39**(3), 213–235 (2005). DOI 10.1007/s11265-005-4841-x
78. Sirowy, S., Huang, C., Vahid, F.: Dynamic acceleration management for SystemC emulation. ACM SIGBED Review **6**(3), 1 (2009). DOI 10.1145/1851340.1851345
79. Suvorova, E., Matveeva, N., Rabin, A., Rozanov, V.: System level modeling of dynamic reconfigurable system-on-chip. In: 2015 17th Conference of Open Innovations Association (FRUCT), pp. 222–229. IEEE (2015). DOI 10.1109/FRUCT.2015.7117996

80. Taghipour, H., Frounchi, J., Zarifi, M.H.: Design and implementation of MP3 decoder using partial dynamic reconfiguration on virtex-4 FPGAs. Proceedings of the International Conference on Computer and Communication Engineering 2008, ICCCE08: Global Links for Human Development pp. 683–686 (2008). DOI 10.1109/ICCCE.2008.4580691
81. Tehre, V., Kshirsagar, R.: Survey on Coarse Grained Reconfigurable Architectures. International Journal of Computer Applications **48**(16), 1–7 (2012). DOI 10.5120/7429-0104
82. Tessier, R., Burleson, W.: Reconfigurable computing for digital signal processing: A survey. Journal of VLSI Signal Processing **28**(1), 7–27 (2001). DOI 10.1023/A:1008155020711
83. Tianfield, H., Unland, R.: Towards autonomic computing systems. Engineering Applications of Artificial Intelligence **17**(7), 689–699 (2004). DOI 10.1016/j.engappai.2004.08.029
84. Ullmann, M., Becker, J.: Communication concept for adaptive intelligent run-time systems supporting distributed reconfigurable embedded systems. In: Proceedings 20th IEEE International Parallel & Distributed Processing Symposium, IPDPS'06, p. 8 pp. IEEE (2006). DOI 10.1109/IPDPS.2006.1639442
85. Voss, P.B., Hole, L.R., Helbling, E.F., Roberts, T.J.: Continuous in-situ soundings in the arctic boundary layer: A new atmospheric measurement technique using controlled meteorological balloons. Journal of Intelligent & Robotic Systems **70**(1), 609–617 (2013). DOI 10.1007/s10846-012-9758-6
86. Wagner, M., Zöbel, D., Meroth, A.: Re-configuration in SOA-based Adaptive Driver Assistance Systems. SIGBED Rev. **11**(3), 30–35 (2014). DOI 10.1145/2692385.2692390
87. Walsh, A., Bordeleau, F., Selic, B., by Lionel Briand D Walsh, C.J., Bordeleau, F., Selic, B.: Domain analysis of dynamic system reconfiguration. Softw Syst Model **6**, 355–380 (2007). DOI 10.1007/s10270-006-0038-4
88. Wang, L., Ding, C., Wen, D., Jiang, Y.: A high-performance, low-area reconfiguration controller for network-on-chip-based partial dynamically reconfigurable system-on-chip designs. International Journal of Electronics **97**(10), 1207–1225 (2010). DOI 10.1080/00207217.2010.512019
89. Wang, X.W., Chen, W.N., Peng, C.L., You, H.J.: Hardware-software monitoring techniques for dynamic partial reconfigurable embedded systems. Proceedings - The 2008 International Conference on Embedded Software and Systems Symposia, ICESS Symposia pp. 113–119 (2008). DOI 10.1109/ICESS.Symposia.2008.10
90. Wang, X.W., Chen, W.N., Wang, Y., Peng, C.L.: A Co-design Flow for Reconfigurable Embedded Computing System with RTOS Support. In: 2009 International Conference on Embedded Software and Systems, pp. 467–474. IEEE (2009). DOI 10.1109/ICESS.2009.84
91. Wang, X.W., Chen, W.N., Wang, Y., You, H.J., Peng, C.L.: The design and implementation of hardware task configuration management unit on dynamically reconfigurable SoC. Proceedings - 2009 International Conference on Embedded Software and Systems, ICESS 2009 pp. 179–184 (2009). DOI 10.1109/ICESS.2009.83
92. Wu, H.H., Shen, C.C., Kee, H., Sane, N., Plishker, W., Bhattacharyya, S.S.: Mapping Parameterized Dataflow Graphs onto FPGA Platforms. In: Academic Press Library in Signal Processing: Volume 4 Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing, vol. 4, pp. 643–673. Elsevier Masson SAS (2014). DOI 10.1016/B978-0-12-396501-1.00024-8
93. Wu, K., Madsen, J.: COSMOS: A system-level modelling and simulation framework for coprocessor-coupled reconfigurable systems. Proceedings - 2007 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, IC-SAMOS 2007 pp. 128–136 (2007). DOI 10.1109/ICSAMOS.2007.4285743
94. Wzorek, M., Doherty, P.: Reconfigurable Path Planning for an Autonomous Unmanned Aerial Vehicle. In: 2006 International Conference on Hybrid Information Technology, vol. 2, pp. 242–249. IEEE (2006). DOI 10.1109/ICHIT.2006.253618
95. Zalke, J.B., Pandey, S.K.: Dynamic partial reconfigurable embedded system to achieve hardware flexibility using 8051 based RTOS on Xilinx FPGA. ACT 2009 - International Conference on Advances in Computing, Control and Telecommunication Technologies pp. 684–686 (2009). DOI 10.1109/ACT.2009.174
96. Zamboni, A., Di Thommazo, A., Hernandes, E., Fabbri, S.: Start uma ferramenta computacional de apoio à revisão sistemática. In: Proc.: Congresso Brasileiro de Software (CBSoft10), Salvador, Brazil (2010)
97. Zeller, M., Prehofer, C., Krefft, D., Weiss, G.: Towards Runtime Adaptation in AUTOSAR. SIGBED Rev. **10**(4), 17–20 (2013). DOI 10.1145/2583687.2583691

98. Zhang, X., Ng, K.W.: Review of high-level synthesis for Dynamically Reconfigurable FPGAs. Microprocessors and Microsystems **24**(4), 199–211 (2000). DOI 10.1016/S0141-9331(00)00074-0
99. Zhang, X., Rabah, H., Weber, S.: An Auto-adaptation Method for Dynamically Reconfigurable System-on-Chip. In: 2008 IEEE Computer Society Annual Symposium on VLSI, pp. 499–502. IEEE (2008). DOI 10.1109/ISVLSI.2008.79
100. Zong, C., Mondal, S., Hall, D.A., Jafari, R.: Digitally Assisted Analog Front-end Power Management Strategy via Dynamic Reconfigurability for Robust Heart Rate Monitoring. SIGBED Rev. **12**(3), 36–39 (2015). DOI 10.1145/2815482.2815489