

# Systematic Review of Methodologies for the Development of Embedded Systems

Kristina Blašković<sup>1</sup>, Sanja Čandrić<sup>2</sup>  
Department of Informatics  
University of Rijeka, Rijeka, Croatia

Alen Jakupović<sup>3</sup>  
Business Department  
Polytechnic of Rijeka, Rijeka, Croatia

**Abstract**—Embedded systems encompass software and hardware components developed in parallel. These systems have been the focus of interest for many scholars who emphasized development issues related to embedded systems. Moreover, they proposed different approaches for facilitating the development process. The aim of this work is to identify desirable characteristics of existing development methodologies, which provide a good foundation for development of new methodologies. For that purpose, systematic mapping methodology was applied to the area of embedded systems, resulting in a classification scheme, graphically represented by a multilayer conceptual network. Afterwards, the most significant clusters were identified, using the k-means algorithm and the squared Euclidean distance formula. Overall, the results provide guidelines for further research aiming to propose a holistic approach for the development of special case of embedded systems.

**Keywords**—Embedded systems; development; methodology; multilayer conceptual network; cluster analysis; k-means algorithm

## I. INTRODUCTION

Embedded systems are systems that encompass software and hardware components. These systems are important due to their wide use in industry, transportation, household, and various specific applications [1]. For development of these heterogeneous systems, standard existing methodologies are not fully applicable, because their phases and activities are oriented only to software or hardware components, or general-purpose hardware usage is implied. The classic approach for development of these systems is very similar to the waterfall process model where software and hardware components are developed separately [2]. Because of this separation, verification of the system occurs later, which leads to the introduction of late changes, as well as time and financial expenses.

Because of that many scientists begun to address this issue and started proposing different approaches and methodologies for development of these systems, with the aim of unifying software and hardware development processes. In this paper, our aim is to identify the characteristics of the proposed methodologies that are suitable for development of these complex systems and adaptable to the challenges of modern trends. Thus, this paper answers following research questions:

Q1: Are there any proposals for improving the development process of embedded systems?

Q2: Given the heterogeneity of the system, what part of the system covers the proposal?

Q3: Is the proposal an entirely new approach or an extension of an existing known approach?

Q4: What problems do the proposed approaches focus on and try to solve?

Q5: What implications are achieved?

To answer these questions, set of relevant articles in the field of embedded systems is reviewed, following systematic mapping methodology steps [3]. Then, the articles are classified based on the following criteria: coverage, type of contribution, thematic focus and implications. Finally, to detect desirable characteristics k-means cluster analysis is conducted, resulting in guidelines for future research.

The focus of authors research are real-time closed-loop control systems, which are considered a special case of embedded systems, comprising real-time close-loop control software, with very specific requirements, firmly tied to specially designed and developed application-specific hardware [1]. RCS Methodology is a development methodology for real-time closed loop control systems that considers software part of the system and implies general-purpose hardware [4], [5].

Hardware and software co-design (HW/SW Co-design) [6] is the most common methodology or set of techniques that define guidelines for the comparative design of software and hardware components. In these systems, most functionalities can be implemented as software or as hardware. If they are implemented as hardware, the system has a better performance and better efficiency, but if they are implemented as software, the system gains flexibility and supports late changes. It is up to the designer to identify the optimal design, from the whole set of possibilities, for the defined application. This set of all possible designs, until implementation, is known as the design space [7].

As these methodologies don't fully cover the specific requirements of the special case of embedded systems in research focus, analysis of the characteristics of existing approaches and methodologies for the development of embedded systems is conducted with the aim to define backbone of the framework which will sustain the development of other new methodologies.

The paper is organized as follows. After background in Section II, in Section III research methodology is presented. Section IV presents research results and the next chapter discussion of obtained results. In the final chapter the paper gives conclusion and announces future work.

## II. BACKGROUND

Embedded systems are in the research focus of many scientists. Authors of this paper identified several systematic literature reviews in this field.

Research [8] and [9], considered articles from 2008 to 2014. These papers focused on highlighting appropriate tools for system modeling based on requirements. The same authors [10] performed additional systematic literature review where they analyze the use of UML in modeling of embedded systems. The problem of requirements engineering in embedded systems was studied in [11], where articles from 1970 to 2016 were reviewed. The use of special case of these systems, as well as their requirements, were reviewed in [12] and [13].

Authors also found one systematic review of development methodologies, which considers agile concepts and focuses on the time period from 1990 to 2015 [14].

Considering research questions of this paper and experience of the above-mentioned papers which focus on larger time span, this paper reviews the entire set of papers available, taking into account several classification criteria, as described in the following chapter.

## III. RESEARCH METHODOLOGY

### A. Systematic Review

First a set of relevant scientific articles needs to be created, focusing on the issues in development process of embedded systems and proposing new or extended approaches to overcome identified problems. For that purpose, Scopus database and Google Scholar service were identified as relevant sources. Preliminary results focused only on the last five years, but this time span was too narrow to get a sufficient number of articles. Therefore, time limit was not applied, i.e., the entire set of articles available in Scopus and Google scholar was considered.

The complexity and importance of the systematic development process of embedded systems, where software modules are directly tied and implemented on hardware modules, have been emphasized by scientists all around the world for quite some time. To create a systematic overview of relevant articles and define the classification scheme, systematic mapping methodology is applied to the field of development of embedded systems. For that purpose, following queries were executed on identified relevant sources:

- 1) Hardware software digital systems simulation.
- 2) "Hardware software" specification and design methodology.
- 3) HW/SW co-design methodology model embedded systems.

- 4) HW/SW "reconfigurable" architectures.
- 5) Integrating hardware (and/) software.
- 6) Integrated hardware software approach framework.
- 7) "HW/SW" synthesis model design partitioning VHDL.
- 8) Hardware software co-design "Very Large Scale Integration".
- 9) ASIC/SoC design.
- 10) Hardware|HW software|SW codesign methodology dynamic reconfiguration +architecture.
- 11) HW/SW codesign rapid prototyping extended finite state machine model.
- 12) VLSI design methodology PLA.
- 13) Hardware software partitioning integrated design automation.
- 14) Platform-based design.
- 15) Mixed "hardware-software" systems co-simulation co-synthesis model.
- 16) Hardware software systems agile.

On the obtained results for each query exclusion criteria evaluation is performed, excluding patents, citations, books and articles not related to the topic. Assuming that sorting algorithm for identified relevant sources sorts results based on importance, for each defined query only first three results were considered. Additionally, multiple appearances of the same article were counted only once. This resulted in the list of 68 articles. Finally, for each article in the resulting list the abstract, conclusion, title, and (if needed and accessible) the entire content were reviewed. Key words along with important concepts were extracted for identification of coverage, type of contribution, base method, thematic focus and achievement.

### B. Classification

Once the set of articles was prepared for analysis, it was classified according to the several criteria. The following criteria (as presented in Table I) was used: coverage criterion, type of contribution criterion, thematic focus criterion and implications criterion. As a result, classification scheme is obtained and graphically represented through multilayer conceptual network.

### C. Analysis

Finally, set cluster analysis was performed using the k-means algorithm. Each article was assigned to the cluster with the nearest centroid, and the distance was determined with the squared Euclidean distance formula. The algorithm follows these steps:

- 1) Choose a random number of clusters k.
- 2) Choose random k cluster representatives from the analysis set.
- 3) For each article, the distance from every cluster representative is calculated.
- 4) The article is assigned to the cluster with the minimal distance.
- 5) Cluster representatives are updated so that they coincide with the corresponding centroid.
- 6) Steps 1–5 are repeated until there are no updates.

TABLE I. OVERVIEW OF EXISTING APPROACHES TO DEVELOPMENT OF EMBEDDED SYSTEMS

Ref.Num	Article	Coverage	Type of contribution	Base method	Thematic focus	Achievement
1	[15]	E_SY	extension	HW/SW Co-design	TOOLS	SL_SPEC
2	[16]	E_SY	extension	HW/SW Co-design	TOOLS	SL_SPEC
3	[17]	E_SY	extension	HW/SW Co-design	PARTITIONING	PPA
4	[18]	E_SY	new approach	platform-based design	DEV_P	LCS
5	[19]	E_SY	new approach	HW/SW Co-design	TOOLS	FDP
6	[20]	E_SY	new approach	HW/SW Co-design	SYS_MODEL	E_ver
7	[21]	SC	new methodology	new methodology	SPEC_REQ	APPSS
8	[22]	E_SY	new approach	HW/SW Co-design	SYS_MODEL	SL_SPEC
9	[23]	E_SY	extension	HW/SW Co-design	SYS_MODEL	PPA
10	[24]	E_SY	extension	HW/SW Co-design	PARTITIONING	DRA
11	[25]	E_SY	extension	HW/SW Co-design	PARTITIONING	DRA
12	[26]	E_SY	extension	HW/SW Co-design	PARTITIONING	DRA
13	[27]	E_SY	new methodology	new methodology	SYS_MODEL	SL_SPEC
14	[28]	E_SY	new approach	prototyping	TOOLS	E_int
15	[29]	SC	new methodology	new methodology	SPEC_REQ	APPSS
16	[30]	SC	new methodology	new methodology	SPEC_REQ	APPSS
17	[31]	E_SY	new approach	HW/SW Co-design	TOOLS	FWE
18	[32]	E_SY	extension	HW/SW Co-design	TOOLS	SL_SPEC
19	[33]	E_SY	extension	HW/SW Co-design	SYS_MODEL	SL_SPEC
20	[34]	E_SY	extension	HW/SW Co-design	SYS_MODEL	SL_SPEC
21	[35]	E_SY	extension	HW/SW Co-design	PARTITIONING	E_ver
22	[36]	E_SY	extension	HW/SW Co-design	SYS_SIMUL	E_ver
23	[37]	E_SY	new approach	HW/SW Co-design	SYNTHESIS	SL_SPEC
24	[38]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
25	[39]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
26	[40]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
27	[41]	E_SY	extension	HW/SW Co-design	PARTITIONING	DRA
28	[42]	E_SY	new approach	HW/SW Co-design	TOOLS	DRA
29	[43]	E_SY	extension	HW/SW Co-design	SYS_MODEL	FDP
30	[44]	E_HW	new methodology	new methodology	SYS_MODEL	SL_SPEC
31	[45]	E_SY	new approach	platform-based design	SYS_MODEL	LCS
32	[46]	E_SY	new approach	platform-based design	SYS_MODEL	LCS
33	[47]	E_SY	extension	HW/SW Co-design	TOOLS	SL_SPEC
34	[48]	E_SY	extension	HW/SW Co-design	TOOLS	SL_SPEC
35	[49]	E_SY	extension	HW/SW Co-design	TOOLS	FWE
36	[50]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
37	[51]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
38	[52]	E_HW	new methodology	new methodology	SYS_MODEL	FDP
39	[53]	E_SY	extension	HW/SW Co-design	PARTITIONING	FDP
40	[54]	E_SY	extension	HW/SW Co-design	TOOLS	PPA
41	[55]	E_SY	extension	HW/SW Co-design	PARTITIONING	PPA
42	[56]	E_SW	new approach	agile	DEV_P	E_int
43	[57]	E_HW	new approach	agile	DEV_P	E_int
44	[58]	E_HW	new approach	agile	DEV_P	E_int
45	[59]	E_SY	new methodology	new methodology	PARTITIONING	PPA
46	[60]	E_SY	new approach	HW/SW Co-design	SYS_SIMUL	E_ver
47	[61]	E_SY	extension	HW/SW Co-design	PARTITIONING	SL_SPEC
48	[62]	E_HW	extension	HW/SW Co-design	SYS_MODEL	DRA
49	[63]	E_SY	new approach	HW/SW Co-design	SYS_MODEL	DRA

Ref.Num	Article	Coverage	Type of contribution	Base method	Thematic focus	Achievement
50	[64]	SC	extension	HW/SW Co-design	SPEC_REQ	APPSS
51	[65]	SC	extension	HW/SW Co-design	SPEC_REQ	APPSS
52	[66]	E_SY	new methodology	new methodology	DEV_P	FWE
53	[67]	E_HW	new methodology	new methodology	DEV_P	FWE
54	[68]	E_SW	new approach	HW/SW Co-design	SYS_SIMUL	E_ver
55	[69]	E_SY	new approach	HW/SW Co-design	SYS_MODEL	E_ver
56	[70]	SC	new approach	HW/SW Co-design	SPEC_REQ	APPSS
57	[71]	E_HW	new approach	prototyping	PARTITIONING	LCS
58	[72]	SC	new methodology	new methodology	SPEC_REQ	APPSS
59	[73]	SC	new methodology	new methodology	SPEC_REQ	APPSS
60	[74]	E_HW	extension	HW/SW Co-design	PARTITIONING	PPA
61	[75]	E_HW	new approach	HW/SW Co-design	PARTITIONING	FWE
62	[76]	E_HW	extension	HW/SW Co-design	PARTITIONING	DRA
63	[77]	E_SY	new approach	platform-based design	SYS_MODEL	SL_SPEC
64	[78]	E_SY	new approach	platform-based design	SYS_MODEL	SL_SPEC
65	[79]	E_SY	new approach	platform-based design	SYNTHESIS	PPA
66	[80]	E_SY	extension	HW/SW Co-design	SYS_SIMUL	E_ver
67	[81]	E_SY	extension	HW/SW Co-design	SYS_MODEL	PPA
68	[82]	E_SY	new approach	agile	SYS_MODEL	E_ver

In the process of implementation of the k-means algorithm, Excel STANDARDIZE, SUMXMY2 and MIN functions were used.

#### IV. RESULTS

Results gathered in the process described in the Methodology section are shown in Table I. Obtained results give answer to the research question Q1 as they present proposals for improving the development process of embedded systems. These results are further analyzed in accordance with the selected criteria and research questions set in this paper.

1) *Coverage criterion*: Classification based on the Coverage criterion answers research question Q2.

Articles that consider the system as a whole, were categorized as “embedded system” (or E\_SY, as noted in the column Coverage in Table I). If the article covered only part of the system, i.e. only software or only hardware, it was categorized either as “embedded software” (E\_SW) or “embedded hardware” (E\_HW). The category “special case” (SC) includes articles which cover special cases of hardware-software systems.

The results in Fig. 1 show that most articles focused on the whole system, while eight articles (7,15,16,50,51,56,58,59) focused on solving specific requests of special cases of these systems. Two articles (42,54) focused on the development of embedded software, and 15 articles (24,25,26,30,36,37,38, 43,44,48,53,57,60,61,62) prescribed steps in the development process of embedded hardware.

2) *Type of contribution criterion*: According to this criterion, methodologies presented in the observed articles were further classified. In Table I in the column Type of contribution this classification is shown.

Articles that include existing methodologies (HW/SW Co-design, agile, platform-based design (PLB), and prototyping) were subcategorized in these subcategories: extension (of an existing methodology), a new approach (based on an existing methodology). Understandably, if a methodology was recognized as completely new, it does not have subcategories and therefore it is noted as new methodology.

The results in Fig. 2 show that the HW/SW Co-design methodology is the most common approach for developing these complex systems. Much effort has been invested in proposing extensions of this methodology (1,2,3,9,10,11,12, 18,19,20,21,22,27,29,33,34,35,39,40,41,47,48,50,51,60,62,66, 69).

However, in 23 articles, scientists stated that these heterogeneous complex systems demand novel development methods. Therefore, 12 articles (5,6,8,17,23,28,46,49,54,55, 56,61) proposed new approaches based on HW/SW Co-design methodology, 12 articles (4,14,31,32,42,43,44,57,63,64,65,68) introduced modern concepts in the development process, and 16 articles proposed new development methodologies (7,13, 15,16,24,25,26,30,36,37,38,45,52,53,58,59).

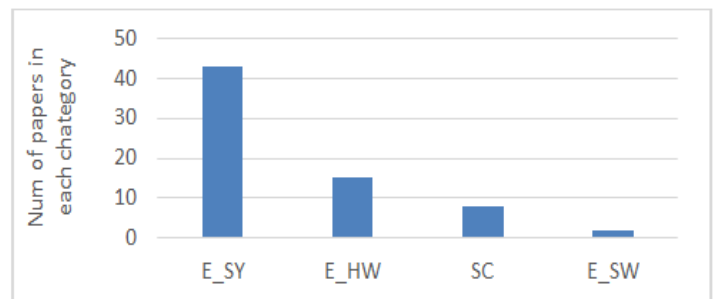


Fig. 1. Graphical Representation of Article Classification based on Coverage.

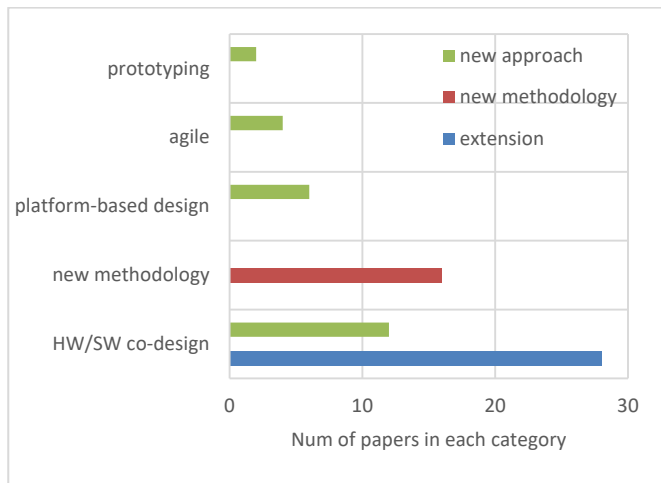


Fig. 2. Graphical Representation of Classification of Articles based on the Type of Contribution.

These results answer the research question Q3.

3) *Thematic focus criterion*: To answer research question Q4 thematic focus criterion was defined. In this criterion articles were first categorized by coverage criterion and then according to their focus on:

- support tools (TOOLS) - programming tools that support the development process,
- partitioning (PARTITIONING) - improvement and automation of the process for choosing appropriate SW/HW components,
- system modeling (SYS\_MODEL) - definition of good practices when choosing between software or hardware implementation,
- synthesis of system components (SYNTESIS) - integration of developed SW/HW components, including verification and validation,
- system simulation (SYS\_SIMUL) - simulation of the system operation in the absence of sufficiently developed hardware components,
- the entire development process (DEV\_P),
- solving specific requirements (SPEC\_REQ).

This data is presented in the column Thematic focus of Table I. Graphically, the results are presented in Fig. 3.

Fig. 3 shows that of a total of 43 articles that considered the system as a whole, 11 articles (1,2,5,14, 17,18,28,33,34,35,40) focused on defining the appropriate set of tools that would support the development process. Four articles (22,46,54,66) dealt with system simulation, and two articles (23,65) focused on the integration of developed software and hardware components. Furthermore, 23 articles dealt with the problem of defining the unique system model, of which 15 articles (6,8,9,13,19,20,29,31,32,49,55,63,64,67, 68) considered the whole system, and eight articles (24,25,26,30,36,37,38,48) considered only the hardware part of the system. Additionally, six articles tried to improve the

entire development process. Two of them (4,52) followed the development of the whole system, one article (42) proposed improvements by introducing agile methods in the software development process, and three articles (43,44,53) dealt with the hardware development process. Out of 14 articles that tried to optimize the partitioning process for hardware and software components, ten articles (3,10,11,12,21,27,39,41,45,47) considered the whole system, and four articles (57,60,61,62) the hardware part of the system. Finally, eight articles (7,15,16,50,51,56,58,59) tried to solve problems for special cases of these systems.

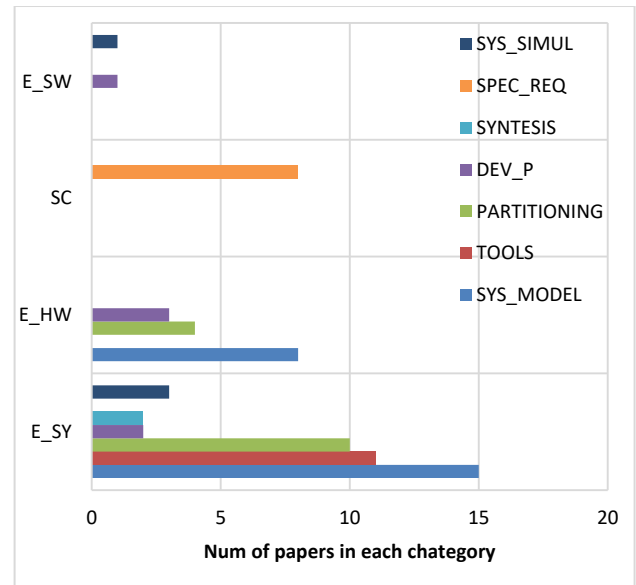


Fig. 3. Graphical Representation of Classification of Articles based on the Thematic Focus.

4) *Implications criterion*: According to this criterion articles were categorized by achieved implications and then by thematic focus criterion, to show how these two criteria are correlated. The following implications were identified:

- system-level specification (SL\_SPEC) - software and hardware components are equally specified, and the type of implementation for each component is selected in the implementation phase,
- facilitated development process (FDP) - introduction of improvements in the development process,
- part of process automation (PPA) - introduction of tools or definition of algorithms that automate a certain development phase,
- dynamic reconfiguration architecture (DRA) - flexibility of the hardware part of the system with respect to the implementation of late changes,
- early integration (E\_int) - in the case of disagreement with defined requirements, in verification and validation process of the system design, needed changes must be introduced and implementing changes earlier in the development process is less expensive,

- early verification (E\_ver) - verification of software and hardware components interaction in the early development stage when the hardware structure is not fully functional,
- late changes support (LCS) - due to the heterogeneous structure of the system, changes can be introduced in software or in hardware and since modification of the hardware part of the system is very difficult in the late development stage the goal is to implement the majority of functionalities as software,
- application-specific solution (APPSS)-solving problems and requirements of application-specific systems,
- framework established (FWE) - defining a set of program tools that support every phase in the development process of these complex systems.

This data is shown in the Achievement column of Table I. The results presented in Fig. 4, give answer to the research question Q5.

14 articles dealt with the problem of system heterogeneity. Of those, five articles (1,2,18,33,34) proposed the introduction of software tools, seven articles focused on system model definition (8,13,19,20,30,63,64), one article (23) proposed improvements in the synthesis phase, and one article (47) examined the partitioning phase. To facilitate the overall development process of these complex systems, seven articles (24,25,26,29,36,37,38) focused on system model definition, one article (39) on the partitioning phase, and one article (5) on defining tools that would support the entire process.

In eight articles, scientists introduced automation of part of the process. Out of these eight, in four articles (3,41,45,60) partitioning was improved by introduction of authors own partitioning algorithms, and in two articles (9,69), the MARTE extension and Web Ontology Language were used to create semantic representations of hardware and software components, from which system model and automatic code were generated, in one article (65), the authors focused on the synthesis phase, and in one article (40), software tools were proposed. Furthermore, eight articles (7,15,16,50,51,56, 58,59) were oriented toward solving application-specific problems. With further development of hardware technology, eight articles proposed the introduction of dynamic reconfiguration architecture as the hardware part of the system. Five of these articles (10,11,12,27,62) proposed improvements in the partitioning phase, two articles (48,49) suggested improved system models, and in one article (28), software tools that allowed selection of dynamic reconfiguration architecture, and thus, variability of the hardware part of the system, were used in the design phase. It is very important to test as soon as possible during the development process does the system in development satisfy preset requirements. For that purpose, eight articles proposed approaches that would allow early verification, out of which three articles (6,55,68) focused on improving system model definition, one article (21) on improved system partitioning, and four articles (22,46,54,66) on the introduction of simulators that would allow verification of the system in early stage. Furthermore, five articles (17,35,52,53,61) contributed

to the establishment of a framework for supporting each development process phase.

In the hardware part of the system, it is very difficult to introduce changes late in the development process. However, as requests for changes are unavoidable, four articles contributed to the introduction of late changes. Of these articles, one (4) tried to ease the entire development process by introducing general-purpose hardware as the hardware part of the system, and application specifics were defined in the software part of the system that supports late changes. The other two articles (32,30) defined the system model on the platform level, where instances from a higher level of abstraction are mapped on lower levels, and each level of abstraction hides implementation details allowing the designer to make compromises in selecting the required components. In one article (57), late changes were supported with the introduction of modern concepts in partitioning. To enable early integration of system components, in three articles (42,43,44), scientists proposed the introduction of modern agile concepts in the entire development process, and in one article (14), support tools were presented, with which mockups of actual software and hardware components were developed, verified, and validated, using an augmented reality (AR) system. If the design is valid, development begins.

As the criteria for classifying the articles were defined, the relationship between them can now be analyzed. For that purpose, a multilayer conceptual network (Fig. 5) is generated to give a better overview of how the criteria affect each other. There are four layers where a single layer is occupied by values for one defined criterion. The values are represented by blue circles. Two layers can be connected, through their values, only if the layers are adjacent. If a single value appears in more than one connection, frequency increases resulting in larger radius circle representation.

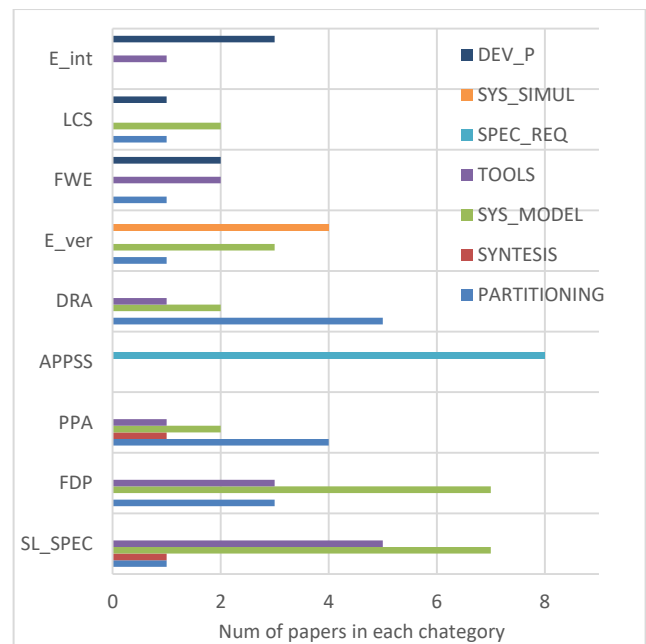


Fig. 4. Graphical Representation of Classification of Articles based on the Implications.



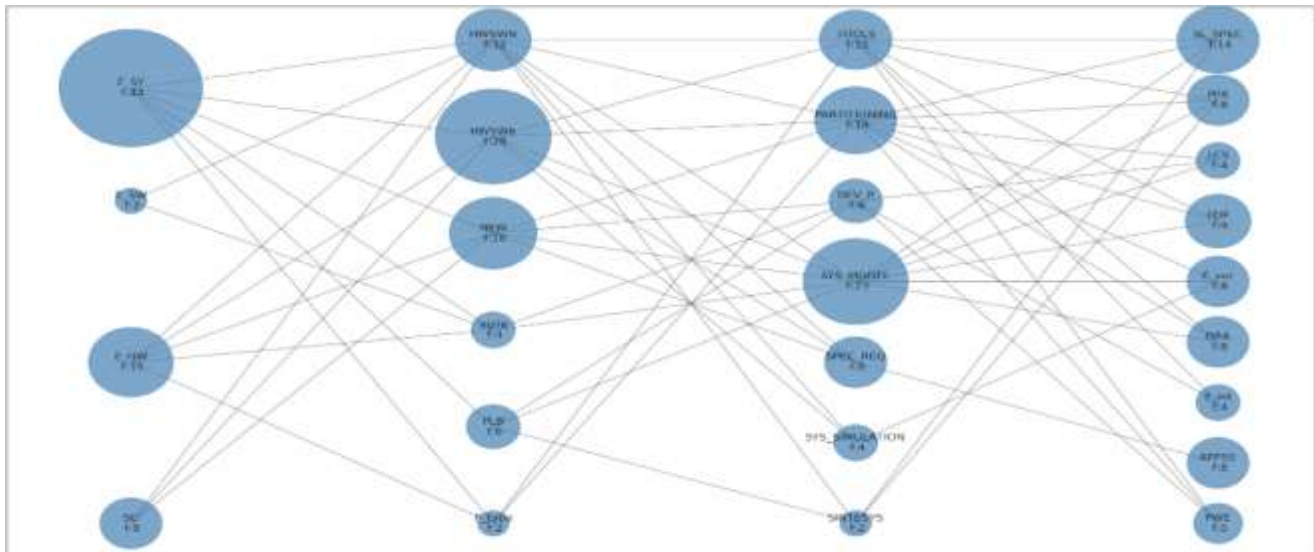


Fig. 5. Multilayer Conceptual Network.

5) *Cluster analysis*: Since k-means algorithm is based on determining the distances from the cluster centroids which implies numerical values, non-numerical values from the analysis set are first assigned to numerical values as shown in Table II.

Next, the mean value and the standard deviation for each criterion were calculated, and data normalized. Afterwards, the desired number of clusters is chosen using the elbow method. In this method first the graphical representation (Fig. 6) is generated where on the y axis the standard deviation of the shortest distances from the cluster centroid is shown, and on the x axis the number of clusters can be found. Finally, the point, where the graph starts flattening, which represents the optimal number of clusters, is chosen (in this case,  $k = 3$ ).

Once the k random cluster representatives were chosen from the analysis set of data, the distance for each element from each cluster representative was calculated and minimal distance corresponding to assigned cluster identified.

When each element of the analysis set is assigned to a certain cluster, cluster representatives are updated so that their position is in the center of each cluster. For that Excel Solver add-on is used which minimizes the sum of the shortest distances by changing the cluster representatives. As a result, clusters presented in Table III are created.

In Fig. 7 the graphical representation is shown. Here on the x axis the number of clusters is shown and on the y axis distance from cluster centroid for each element of the cluster can be found. If the element coincides with cluster centroid the distance equals 0 and the values increase with distance. If more than one element has the same distance value, the radius of the circle marker increases proportionally.

TABLE II. ASSOCIATING NUMERICAL VALUES TO CLASSIFICATION CRITERIA VALUES

Values for coverage	
Abbreviation	Value
E_SY	1
E_SW	2
E_HW	3
SC	4
Values for type of contribution	
Abbreviation	Value
HWSWE	1
HWSWN	2
NEW	3
agile	4
PLB	5
P_type	6
Values for thematic focus	
Abbreviation	Value
TOOLS	1
PARTITIONING	2
DEV_P	3
SYS_MODEL	4
SPEC_REQ	5
SYS_SIMULATION	6
SINTESYS	7
Values for implications	
Abbreviation	Value
SL_SPEC	1
PPA	2
LCS	3
FDP	4
E_ver	5
DRA	6
E_int	7
APPSS	8
FWE	9



Fig. 6. Determining the Optimal Number of Clusters with the Elbow Method.

TABLE III. CLUSTER ANALYSIS RESULTS

Cluster	Articles	Nr of el.
1	4,13,14,31,32,45,57,63,64,65 and 68	11
2	7,15,16,23,24,25,26,30,36,37,38,42,43,44,46,48,50,51,53,54,56,58,59 and 61	24
3	1,2,3,5,6,8,9,10,11,12,17,18,19,20,21,22,27,28,29,33,34,35,39,40,41,47,49,52,55,60,62,66 and 69	33
<b>Total</b>		<b>68</b>

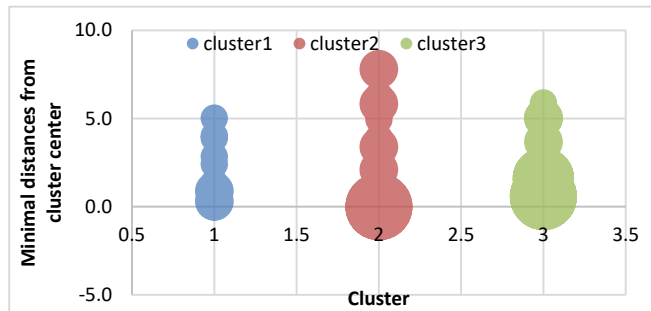


Fig. 7. Graphical Representation of the Cluster Distribution.

## V. DISCUSSION

Relevant articles on methodologies for the development of embedded systems were analyzed by applying the systematic mapping methodology. This resulted in classification schema based on several criteria. In terms of coverage, most articles considered the system as a whole. Recently, the number of articles that considered special cases of these systems has increased. In terms of contribution, most articles proposed improvements based on the HW/SW Co-design methodology. Numerous articles also proposed entirely new approaches for facilitating the development process. Furthermore, articles proposed the introduction of modern concepts in the development of software, as well as the hardware part of the system. In terms of thematic focus, most articles focused on defining a system model. Such model unifies hardware and software development processes. Many articles also dealt with the partitioning phase and the definition of adequate software tools that support the development process of embedded systems. In addition, several scholars dealt with specific requirements of special cases of these systems. In terms of implications, the most frequent implication is the definition of system-level specification. This specification provides an overview of the system at a higher level of abstraction. Other most common implications included facilitated development

process, part of the process automation, application-specific solutions, introduction of dynamic reconfiguration architecture, and the possibility of early verification.

Based on this classification schema cluster analysis is performed using k-means algorithm. Analysis resulted in three clusters as shown in Fig. 7 and Table III.

A review of the obtained clusters shows that cluster 3 is the cluster that includes the most elements. The total number (33 elements) accounts for 48.53% of the initial data set. Articles with the following characteristics were found: coverage (the whole system and the hardware part of the system), contribution (extension or a new approach based on hardware/software co-design methodology and new methodology), thematic focus (software tools that support development process, improvements in the partitioning phase of the development process, definition of the optimal system model, system simulation, and the entire development process), and implications (definition of system-level specification, automation of part of the process, facilitated development process, support for early verification, introduction of dynamic reconfiguration architecture, and framework establishment).

The second largest cluster is cluster 2 with 24 elements which is 35.29% of the initial data set. Articles in this cluster have the following characteristics: coverage (special cases of these systems, the whole system, the hardware part of the system, and the software part of the system), contribution (new methodology, new approaches and extensions based on hardware/software co-design methodology, and new approaches based on agile concepts), thematic focus (definition of the optimal system model, solving specific requirements of special cases of these systems, the synthesis phase of the development process, the entire development process, system simulation, and the partitioning phase of the development process), and implications (automation of part of the process, application specific solutions, definition of system-level specification, facilitated development process, support for early integration, support for early verification, introduction of dynamic reconfiguration architecture, framework establishment).

The smallest cluster is cluster 1 with 11 elements which is 16.18% of the initial data set. Articles in this cluster have the following characteristics: coverage (the whole system and the hardware part of the system), contribution (proposals of new approaches based on platform-based design, agile concepts, prototyping and new methodologies), thematic focus (the entire development process, definition of the optimal system model, software tools that support the development process, the partitioning phase of the development process), and implications (support for introduction of changes in a late stage of the development process, definition of system-level specification, support for early integration, part of process automation, and support for early verification).

The results show that for the first criterion (coverage) there are no major deviations between clusters 1 and 3, and special cases of these complex systems appear only in cluster 2. For the second criterion (contribution), new approaches based on modern concepts appear only in clusters 2 and 3. The third



criterion (thematic focus) is similar in all three clusters, but a focus on solving specific requirements appears only in cluster 2. Equally, all clusters are similar in fourth criterion (implications), but application-specific implications can be found only in cluster 2 and support for late changes implications only in cluster 1.

## VI. CONCLUSION

Embedded systems are systems that encompass software and hardware components. For the last 50 years, great effort has been invested in proposing improvements and new approaches to the development process. Guided by many researchers who recognized the significance of these complex systems integrated in our everyday lives, in this paper the desirable characteristics of existing approaches to the development of embedded system were detected, which can be used as the foundation for the development of new methodologies.

After extracting relevant scientific articles through systematic review process, based on defined criteria cluster analysis was conducted, implementing k-means algorithm and the squared Euclidean distance. It resulted in three clusters, where cluster 3 and cluster 2 were identified as significant, considering the number of associated elements.

In our previous research we proposed Software development methodology for special case of real-time closed loop control systems. This methodology supports development of application-specific hardware and solves specific requirements.

In our future work a new methodology for development of special case of embedded systems will be designed and proposed. This methodology will unify development process of software and hardware components and will be based on newly developed graphical methods. Furthermore, evaluation procedures will be developed, intended for graphical methods and methodology assessment.

## ACKNOWLEDGMENT

This work has been fully supported by the University of Rijeka under the project uniri-drustv-18-140.

## REFERENCES

- [1] K. Blašković and S. Čandrić, "DEM4RTS: Software development methodology for special case of real-time closed-loop control systems", *Annals of DAAAM & Proceedings*, vol. 29, 2018.
- [2] M. Chiodo, "Automotive Electronics", In *Hardware/software co-design*, Springer, Dordrecht, pp. 295-310, 1996.
- [3] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, "Systematic mapping studies in software engineering", In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*, pp. 1-10, 2008.
- [4] T. Barbera, M. L. Fitzgerald and J. S. Albus, "Concepts for a real-time sensory-interactive control system architecture", In *Fourteenth Southeastern Symposium on System Theory*, 1982.
- [5] R. Quintero and T. Barbera, "A Real-Time Control System Methodology for Developing Intelligent Control Systems", 1992.
- [6] J. Staunstrup and W. Wolf, "Hardware/software co-design: principles and practice", Springer Science & Business Media, 2013.
- [7] E. Kang, E. Jackson and W. Schulte, "An approach for effective design space exploration", In *Monterey Workshop*, Springer, Berlin, Heidelberg, pp. 33-54, 2010.

- [8] M. Rashid, M. W. Anwar and A. M. Khan, "Toward the tools selection in model based system engineering for embedded systems—A systematic literature review", *Journal of Systems and Software*, vol. 106, pp. 150-163, 2015.
- [9] M. Rashid and M. W. Anwar, "A systematic investigation of tools in model based system engineering for embedded systems", In *2016 11th System of Systems Engineering Conference (SoSE)*, IEEE, pp. 1-6, 2016.
- [10] M. Rashid, M. W. Anwar and A. M. Khan, "Identification of trends for model based development of embedded systems", In *2015 12th International Symposium on Programming and Systems (ISPS)*, IEEE, pp. 1-8, 2015.
- [11] T. Pereira, D. Albuquerque, A. Sousa, F. M. Alencar and J. Castro, "Retrospective and Trends in Requirements Engineering for Embedded Systems: A Systematic Literature Review", In *CIBSE*, pp. 427-440, 2017.
- [12] M. Romero, W. Guédria, H. Panetto and B. Barafort, "Towards a characterisation of smart systems: A systematic literature review", *Computers in industry*, vol. 120, pp. 103224, 2020.
- [13] B. Liao, Y. Ali, S. Nazir, L. He and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review", *IEEE Access*, vol. 8, pp. 120331-120350, 2020.
- [14] R. Hoda, N. Salleh, J. Grundy and H. M. Tee, H. M. (2017). "Systematic literature reviews in agile software development: A tertiary study", *Information and Software Technology*, vol. 85, pp. 60-70, 2017.
- [15] R. K. Gupta, C. N. Coelho and G. De Micheli, "Synthesis and simulation of digital systems containing interacting hardware and software components", In *DAC*, vol. 92, pp. 225-230, 1992.
- [16] G. De Micheli, "Computer-aided hardware-software codesign", *IEEE Micro*, vol. 14(4), pp. 10-16, 1994.
- [17] K. A. Olukotun, R. Helaihel, J. Levitt and R. Ramirez, "A software-hardware cosynthesis approach to digital system simulation. *IEEE Micro*", vol. 14(4), pp. 48-58, 1994.
- [18] A. Sangiovanni-Vincentelli and G. Martin, "Platform-based design and software design methodology for embedded systems", *IEEE Design & Test of Computers*, vol. 18(6), pp. 23-33, 2001.
- [19] D. D. Gajski and F. Vahid, "Specification and design of embedded hardware-software systems", *IEEE Design & Test of Computers*, vol. 12(1), pp.53-67, 1995.
- [20] M. Chiodo, P. Giusto, A. Jurecska, H. C. Hsieh, A. Sangiovanni-Vincentelli and L. Lavagno, "Hardware-software codesign of embedded systems", *IEEE micro*, vol. 14(4), pp. 26-36, 1994.
- [21] J. Vidal, F. De Lamotte, G. Gogniat, P. Soulard and J. P. Diguët, "A co-design approach for embedded system modeling and code generation with UML and MARTE", In *2009 Design, Automation & Test in Europe Conference & Exhibition*, IEEE, pp. 226-231, 2009.
- [22] A. Shaout, A. H. El-Mousa and K. Mattar, "Specification and modeling of hw/sw co-design for heterogeneous embedded systems", In *Proceedings of the World Congress on Engineering*, vol. 1, 2009.
- [23] L. G. Murillo, M. Mura and M. Prevostini, "Semi-automated Hw/Sw Co-design for embedded systems: from MARTE models to SystemC simulators", In *2009 Forum on Specification & Design Languages (FDL)*, IEEE; pp. 1-6, 2009.
- [24] S. Banerjee, E. Bozorgzadeh and N. Dutt, "Physically-aware HW-SW partitioning for reconfigurable architectures with partial dynamic reconfiguration", In *Proceedings of the 42nd annual Design Automation Conference*, pp. 335-340, 2005.
- [25] J. Noguera and R. M. Badia, "HW/SW codesign techniques for dynamically reconfigurable architectures", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10(4), pp. 399-415, 2002.
- [26] M. Baleani, F. Gennari, Y. Jiang, Y. Patel, R. K. Brayton and A. Sangiovanni-Vincentelli, "HW/SW partitioning and code generation of embedded control applications on a reconfigurable architecture platform", In *Proceedings of the tenth international symposium on Hardware/software codesign*, pp. 151-156, 2002.
- [27] S. Donatelli and G. Franceschinis, "The PSR methodology: integrating hardware and software models", In *International Conference on*

- Application and Theory of Petri Nets, Springer, Berlin, Heidelberg, pp. 133-152, 1996.
- [28] T. J. Nam and W. Lee, "Integrating hardware and software: augmented reality based prototyping method for digital products", In CHI'03 extended abstracts on Human factors in computing systems, pp. 956-957, 2003.
- [29] P. Francesco, P. Marchal, D. Atienza, L. Benini, F. Catthoor and J. M. Mendias, "An integrated hardware/software approach for run-time scratchpad management", In Proceedings of the 41st annual Design Automation Conference, pp. 238-243, 2004.
- [30] A. Shriraman, M. F. Spear, H. Hossain, V. J. Marathe, S. Dwarkadas and M. L. Scott, "An integrated hardware-software approach to flexible transactional memory", In Proceedings of the 34th annual international symposium on Computer architecture, pp. 104-115, 2007.
- [31] S. Kumar, J. H. Aylor, B. W. Johnson and W. A. Wulf, "A framework for hardware/software codesign", Computer, vol. 26(12), pp. 39-45, 1993.
- [32] T. B. Ismail and A. A. Jerraya, "Synthesis steps and design models for codesign", Computer, vol. 28(2), pp.44-53, 1995.
- [33] W. Ecker, "Using VHDL for HW/SW co-specification", In Proceedings of EURO-DAC 93 and EURO-VHDL 93-European Design Automation Conference, IEEE, pp. 500-505, 1993.
- [34] C. A. Valderrama, A. Changuel, P. V. Raghavan, M. Abid, T. B. Ismail and A. A. Jerraya, "A unified model for co-simulation and co-synthesis of mixed hardware/software systems", In Proceedings the European Design and Test Conference. ED&TC 1995, IEEE, pp. 180-184, 1995.
- [35] F. Vahid and D. D. Gajski, "Incremental hardware estimation during hardware/software functional partitioning", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 3(3), pp. 459-464, 1995.
- [36] G. De Micheli, "Hardware/software co-design: Application domains and design technologies", In Hardware/software co-design, Springer, Dordrecht, pp. 1-28, 1996.
- [37] S. Parameswaran, "Code placement in Hardware Software Co synthesis to improve performance and reduce cost", In Proceedings Design, Automation and Test in Europe. Conference and Exhibition 2001, IEEE, pp. 626-632, 2001.
- [38] J. Henkel, "Closing the SoC design gap", Computer, vol. 36(9), pp. 119-121, 2003.
- [39] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D., Sylvester, D., ... and S. Kulkarni, "Pushing ASIC performance in a power envelope", In Proceedings of the 40th annual Design Automation Conference, pp. 788-793, 2003.
- [40] G. W. Doerre and D. E. Lackey, "The IBM ASIC/SoC methodology—A recipe for first-time success", IBM Journal of Research and Development, vol. 46(6), pp. 649-660, 2002.
- [41] J. Noguera and R. M. Badia, "A HW/SW partitioning algorithm for dynamically reconfigurable architectures", In Proceedings Design, Automation and Test in Europe, Conference and Exhibition 2001, IEEE, pp. 729-734, 2001.
- [42] A. Antola, M. D. Santambrogio, M. Fracassi, P. Gotti and C. Sandionigi, "A novel hardware/software codesign methodology based on dynamic reconfiguration with Impulse C and CoDeveloper", In 2007 3rd Southern Conference on Programmable Logic, IEEE, pp. 221-224, 2007.
- [43] D. E. Thomas, J. K. Adams and H. Schmit, "A model and methodology for hardware-software codesign", IEEE Design & test of computers, vol. 10(3), pp. 6-15, 1993.
- [44] S. Liao, S. Tjiang and R. Gupta, "An efficient implementation of reactivity for modeling hardware in the Scenic design environment", In Proceedings of the 34th annual Design Automation Conference, pp. 70-75, 1997.
- [45] K. Keutzer, A. R. Newton, J. M. Rabaey and A. Sangiovanni-Vincentelli, "System-level design: orthogonalization of concerns and platform-based design", IEEE transactions on computer-aided design of integrated circuits and systems, vol. 19(12), pp. 1523-1543, 2000.
- [46] A. Sangiovanni-Vincentelli, L. Carloni, F. De Bernardinis and M. Sgroi, "Benefits and challenges for platform-based design", In Proceedings of the 41st annual Design Automation Conference, pp. 409-414, 2004.
- [47] K. Buchenrieder, A. Sedlmeier and C. Veith, "HW/SW co-design with PRAMs using CODES", In Computer Hardware Description Languages and their Applications, North-Holland, pp. 65-78, 1993.
- [48] T. Ismail, M. Abid, K. O'Brien and A. Jerraya, "An approach for hardware-software codesign", In Proceedings of IEEE 5th International Workshop on Rapid System Prototyping, IEEE, pp. 73-80, 1994.
- [49] S. Antoniazzi, A. Balboni, W. Fornaciari and D. Sciuto, "A methodology for control-dominated systems codesign", In Third International Workshop on Hardware/Software Codesign, IEEE, pp. 2-9, 1994.
- [50] N. Jayakumar and S.P. Khatri, "A metal and via maskset programmable VLSI design methodology using PLAs", In IEEE/ACM International Conference on Computer Aided Design, 2004, ICCAD-2004, IEEE, pp. 590-594, 2004.
- [51] N. Jayakuma, R. Garg, B. Gamache and S. P. Khatri, "A PLA based asynchronous micropipelining approach for subthreshold circuit design", In 2006 43rd ACM/IEEE Design Automation Conference, IEEE, pp. 419-424, 2006.
- [52] S. P. Khatri, R. K. Brayton and A. Sangiovanni-Vincentelli, "Cross-talk immune VLSI design using a network of PLAs embedded in a regular layout fabric", In IEEE/ACM International Conference on Computer Aided Design, ICCAD-2000, IEEE/ACM Digest of Technical Papers (Cat. No. 00CH37140), IEEE, pp. 412-418, 2000.
- [53] V. Srinivasan, S. Radhakrishnan R. Vemuri, "Hardware software partitioning with integrated hardware design space exploration", In Proceedings Design, Automation and Test in Europe, IEEE, pp. 28-35, 1998.
- [54] R. Ernst, J. Henkel and T. Benner, "Hardware-software cosynthesis for microcontrollers", IEEE Design & Test of computers, vol. 10(4), pp. 64-75, 1993.
- [55] P. Eles, Z. Peng, K. Kuchcinski and A. Doboli, "System level hardware/software partitioning based on simulated annealing and tabu search", Design automation for embedded systems, vol. 2(1), pp.5-32., 1997.
- [56] J. Ronkainen and P. Abrahamsson, "Software development under stringent hardware constraints: Do agile methods have a chance?", In International Conference on Extreme Programming and Agile Processes in Software Engineering, Springer, Berlin, Heidelberg, pp. 73-79, 2003.
- [57] J. Shalf, D. Quinlan and C. Janssen, "Rethinking hardware-software codesign for exascale systems", Computer, vol. 44(11), pp. 22-30, 2011.
- [58] P. M. Huang, A. G. Darrin and A. A. Knuth, "Agile hardware and software system engineering for innovation", In 2012 IEEE Aerospace Conference, IEEE, pp. 1-10, 2012.
- [59] A. Iguider, K. Bouselam, O. Elissati, M. Chami and A. En-Nouaary, "Embedded systems hardware software partitioning using minimax algorithm", In Proceedings of the 4th International Conference on Smart City Applications, pp. 1-6, 2019.
- [60] S. Karthik, K. Priyadarsini and V. Jeanshilpa, "VLSI Systems for Simultaneous in Logic Simulation", In 2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC), IEEE, pp. 23-27, 2018.
- [61] M. King and N. Dave, "Automatic generation of hardware/software interfaces", ACM SIGARCH Computer Architecture News, vol. 40(1), pp.325-336, 2012.
- [62] P. Zhang, J. Zambreno and P. H. Jones, "An embedded scalable linear model predictive hardware-based controller using ADMM", In 2017 IEEE 28th International Conference on Application-specific Systems, Architectures and Processors (ASAP), IEEE, pp. 176-183, 2017.
- [63] M. Cauwels, J. Zambreno and P. H. Jones, "HW/SW Configurable LQG Controller using a Sequential Discrete Kalman Filter", In 2018 International Conference on ReConfigurable Computing and FPGAs (ReConFig), IEEE, pp. 1-8, 2018.
- [64] D. Sabella, P. Serrano, G. Stea, A. Virdis, I. Tinnirello, F. Giuliano, ... and N. Bartzoudis, "Designing the 5G network infrastructure: a flexible and reconfigurable architecture based on context and content information", EURASIP Journal on Wireless Communications and Networking, vol. 2018(1), pp. 1-16, 2018.
- [65] D. Sabella, P. Serrano, G. Stea, A. Virdis, I. Tinnirello, F. Giuliano,... and N. Bartzoudis, "A flexible and reconfigurable 5G networking

- architecture based on context and content information", In 2017 European Conference on Networks and Communications (EuCNC), IEEE, pp. 1-6, 2017.
- [66] J. R. Stevens, Y. Du, V. Kozhikkott and A. Raghunathan, "ACCLIB: Accelerators as libraries", In 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, pp. 245-248, 2018.
- [67] C. W. Lin and C. Chen, "A processor and cache online self-testing methodology for os-managed platform", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25(8), pp. 2346-2359, 2017.
- [68] P. M. Ortega-Cabezas, A. Colmenar-Santos, D. Borge-Diez and J.J. Blanes-Peiró, "Application of rule-based expert systems in hardware-in-the-loop simulation case study: Software and performance validation of an engine electronic control unit", Journal of Software: Evolution and Process, vol. 32(1), 2020.
- [69] C. C. Huang, K. C. Lin, C. H. Huang, H. P. Lin and C. W. Liu, "SoC-based Software-Hardware codesign of intelligent diagnosis", In 2010 International Symposium on Computer, Communication, Control and Automation (3CA), IEEE, vol. 2, pp. 337-340, 2010.
- [70] A. Zemva and M. Verderber, "FPGA-oriented HW/SW implementation of the MPEG-4 video decoder", Microprocessors and Microsystems, vol. 31(5), pp. 313-325, 2007.
- [71] A. Chimienti, L. Fanucci, R. Locatelli and S. Saponara, "VLSI architecture for a low-power video codec system", Microelectronics Journal, vol. 33(5-6), pp. 417-427, 2002.
- [72] C. P. Iatrou, H. Bauer, M. Graube, S. Höppner, J. Rahm and L. Urbas "Hard Real-Time Capable OPC UA Server as Hardware Peripheral for Single Chip IoT Systems", In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, pp. 1631-1634, 2019.
- [73] A. Coelho, N. E. Zergainoh and R. Velazco, "Nocfi: A hybrid fault injection method for networks-on-chip", In 2019 IEEE Latin American Test Symposium (LATS), IEEE, pp. 1-6, 2019.
- [74] J. B. Lim and D. Chen, "Automated Communication and Floorplan-Aware Hardware/Software Co-Design for SoC", In 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, pp. 128-133, 2019.
- [75] W. Zuo, L. N. Pouchet, A. Ayupov, T. Kim, C. W. Lin, A. Shiraishi and D. Chen, "Accurate high-level modeling and automated hardware/software co-design for effective SoC design space exploration", In Proceedings of the 54th Annual Design Automation Conference 2017, pp. 1-6, 2017.
- [76] S. Yousuf and A. Gordon-Ross, "An automated hardware/software co-design flow for partially reconfigurable FPGAs", In 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, pp. 30-35, 2016.
- [77] M. Lora, S. Vinco and F. Fummi, "Translation, abstraction and integration for effective smart system design", IEEE Transactions on Computers, vol. 68(10), pp. 1525-1538, 2019.
- [78] L. Gauss, D. P. Lacerda and M. A. Sellitto, "Module-based machinery design: a method to support the design of modular machine families for reconfigurable manufacturing systems", The International Journal of Advanced Manufacturing Technology, vol.102(9-12), pp.3911-3936, 2019.
- [79] A. Iannopolo, S. Tripakis and A. Sangiovanni-Vincentelli, "Constrained synthesis from component libraries", Science of Computer Programming, vol. 171, pp.21-41, 2019.
- [80] K. Etzold, T. Fahrbach, S. Klein, R. Scheer, D. Guse, M. Klawitter, M., ... and J. Andert, "Function development with an electric-machine-in-the-loop setup: A case study", IEEE Transactions on Transportation Electrification, vol. 5(4), pp. 1419-1429, 2019.
- [81] A. Perzylo, S. Profanter, M. Rickert and A. Knoll, "OPC UA NodeSet Ontologies as a Pillar of Representing Semantic Digital Twins of Manufacturing Resources", In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, pp. 1085-1092, 2019.
- [82] M. Alhaj, G. Arbez and L. Peyton, "Approach of integrating behaviour-driven development with hardware/software codesign", International Journal of Innovative Computing, Information and Control, vol. 15(3), pp.1177-1191, 2019.