Self-Adaptation for Cyber-Physical Systems: A Systematic Literature Review

3 authors:

Henry Muccini
Università degli Studi dell'Aquila
198 PUBLICATIONS **2,618** CITATIONS

Mohammad Sharaf
Università degli Studi dell'Aquila
27 PUBLICATIONS **296** CITATIONS

Danny Weyns
KU Leuven
287 PUBLICATIONS **7,457** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Brakes: Portable Support for Transparant Thread Migration for Java View project

Project   Autonomic Software Product Lines View project

Self-Adaptation for Cyber-Physical Systems:
A Systematic Literature Review

Henry Muccini
DISIM Department
University of L'Aquila
L'Aquila, Italy
henry.muccini@univaq.it

Mohammad Sharaf
DISIM Department
University of L'Aquila
L'Aquila, Italy
massharaf@yahoo.com

Danny Weyns
Dept. of Computer Science
Katholieke Universiteit Leuven
Linnaeus University Sweden
danny.weyns@kuleuven.be

ABSTRACT

Context: Cyber-physical systems (CPS) seamlessly integrate computational and physical components. Adaptability, realized through feedback loops, is a key requirement to deal with uncertain operating conditions in CPS.

Objective: We aim at assessing state-of-art approaches to handle self-adaptation in CPS at the architectural level.

Method: We conducted a systematic literature review by searching four major scientific data bases, resulting in 1103 candidate studies and eventually retaining 42 primary studies included for data collection after applying inclusion and exclusion criteria.

Results: The primary concerns of adaptation in CPS are performance, flexibility, and reliability. 64% of the studies apply adaptation at the application layer and 24% at the middleware layer. MAPE (Monitor-Analyze-Plan-Execute) is the dominant adaptation mechanism (60%), followed by agents and self-organization (both 29%). Remarkably, 36% of the studies combine different mechanisms to realize adaptation; 17% combine MAPE with agents. The dominating application domain is energy (24%).

Conclusions: Our findings show that adaptation in CPS is a cross-layer concern, where solutions combine different adaptation mechanisms within and across layers. This raises challenges for future research both in the field of CPS and self-adaptation, including: how to map concerns to layers and adaptation mechanisms, how to coordinate adaptation mechanisms within and across layers, and how to ensure system-wide consistency of adaptation.

Keywords

Cyber Physical Systems, Self-Adaptiveness, Systematic Study

1. INTRODUCTION

The Horizon 2020 program refers to CPS as "the next generation *embedded ICT systems* that are interconnected and collaborating, providing citizens and businesses with a wide range of innovative applications and services" [1]. The rapid expansion of mobile devices equipped with smart features combined with the progressing integration of networked computing systems have moved CPS from the traditional area of embedded systems towards the area of large-scale distributed systems [2, 3, 4]. Therefore, CPS inherits all the complexities of modern large-scale distributed systems: they have to handle uncertainty and change during operation, control their emergent behavior, and be scalable and tolerant to threats [5].

To cope with uncertain and emerging situations, software should not only control the operation of the CPS, but also be self-aware, context-aware, and goal-aware (i.e., be self-adaptive). These responsibilities of software may span the full technology stack, from the physical level to computing hardware and the network, up to middleware and the application level. Dealing with these responsibilities poses significant challenges to software engineers of CPS. In this paper, we focus at adaptation mechanisms applied to CPS at the architectural level, incl. approaches based on MAPE-feedback loops, controllers, reflection, agents, and self-organisation.

To address the challenges of CPS, innovative approaches have been proposed. However, there is currently no clear view on how self-adaptation has been applied to CPS. Such knowledge is important to identify recurrent problems and develop effective adaptation solutions. While secondary studies exist on architecture-based self-adaptation [6], engineering of control-based adaptation [7], and architecting CPS [8], to the best of our knowledge, no paper has invested the role of self-adaptation in architecting CPS.

Therefore, this paper aims *to analyse existing approaches to self-adaptation in CPS to better understand the state-of-art*. To that end, we performed a systematic literature review that provides a solid method for such a study. Our focus was on the adaptation mechanisms applied at the architecture level, which is widely acknowledged as a suitable level of abstraction and generality to apply self-adaptation [9, 10, 11, 12]. The outcome of the survey will provide knowledge about how adaptation is currently handled in CPS, what problems have been tackled, which methods have been used to solve them, and how solutions have been evaluated. These insights will help identify areas for further investigation and outline concrete challenges for future research.

This paper is organized as follows. Section 2 summarizes the research method used to identify the main research questions, the search string, and inclusion and exclusion criteria to select primary studies. Section 3 presents the results, answering four research questions. In Section 4 we further elaborate on our results. Conclusions and future work directions are outlined in Section 5.

2. RESEARCH METHOD

The research method applied in this survey follows the classical three-stages process for running systematic studies [13, 14, 15]: Planning, Conducting, and Reporting. We briefly report the main activities carried out for running this study. A replication package

(with the detailed protocol, the list of selected primary studies, and the collected and analyzed data) is available at the survey website. [1]

2.1 Planning

Research Questions: the study goal has been achieved by investigating four research questions, reported in Table 1:

RQ ID	Reserch Question
RQ1	How is self-adaptation applied in cyber physical systems?
RQ2	How do existing approaches for self-adaptation in cyber physical systems handle self-adaptation concerns?
RQ3	What type of evidence is provided by existing approaches for self-adaptation in cyber physical systems?
RQ4	What are the strengths and limitations of proposed approaches?

Table 1: Research Questions

We also derived sub-questions as reported in Section 3.

Search Strategy: The search strategy combines a manual search with an automatic search. The search strategy has been designed as a multi-stage process to carefully select all the primary studies that are relevant for our study . In the following we give a brief description of each stage of our search and selection process.

Stage 1: Manual search. We manually searched Google Scholar with the search string *architect* self-adaptive cyber-physical systems*. Then, we manually browsed relevant journals and conference proceedings (available in the replication package). The output of this stage consists in seven pilot studies, 64 primary studies, the definition of an initial search string, and the definition of inclusion and exclusion criteria.

Stage 2: Automatic search. We performed automatic searches on four of the largest and most complete scientific databases: IEEE Xplore Digital Library, ACM Digital Library, SpringerLink, and ScienceDirect. We used the following search string:

```
( s o f t w a r e  OR  s y s t e m )  AND  ( a r c h i t e c t ∗  OR
" h i g h − l e v e l ␣ d e s i g n "  OR  " c o n c e p t u a l ␣ d e s i g n "
OR  " a b s t r a c t ␣ d e s i g n " )  AND  ( " c y b e r − p h y s i c a l "
OR  " d i s t r i b u t e d ␣ c o n t r o l ␣ s y s t e m "  OR  " n e t w o r k e d
c o n t r o l ␣ s y s t e m "  OR  " s e n s o r ␣ a c t u a t o r ␣ n e t w o r k "
OR  " d i s t r i b u t e d ␣ s c a d a "  OR  " f e d e r a t e d ␣ e m b e d d e d
s y s t e m " )
```

Stage 3: Combination and duplicates removal. In this stage all the results from previous stages are combined together into a single spreadsheet, and duplicates are removed.

Stage 4: Selection of studies. The main goal of this stage is to filter all the selected studies according to a set of well-defined inclusion and exclusion criteria. The criteria are described below.

Stage 5: Exclusion of studies during data extraction. This stage was performed in parallel with data extraction. When reading a study in detail (to extract the data), and based on inclusion and exclusion criteria, studies where definitively selected or rejected.

Inclusion and Exclusion criteria: A study was selected if it satisfies all inclusion criteria, and discarded if it meets any of the exclusion criteria.

[1] www.henrymuccini.com/contents/SEAMS2016.html

Inclusion criteria: i) studies proposing, leveraging, or analyzing an architectural solution, architectural method or technique specific for CPSs, ii) studies in which self-adaptation is explicitly used as an instrument to engineer CPS, iii) studies subject to peer review [14] (e.g., journal papers, papers published as part of conference proceedings), iv) studies published after or in 2006.
Exclusion criteria: i) studies that are written in a language other than English, or that are not available in full-text, ii) secondary studies (e.g., systematic literature reviews, surveys, etc.).

Data Items: Table 2 gives an overview of the data items that were extracted from the primary studies to answer the research questions. Except for the first item, we derived these data items directly from the research questions as indicated in the left column.

Data Item	Data Field	Research question
F1	Study *Title*	Doc
F2	Publication *Year*	Doc
F3	*Venue*	Doc
F4	*Context* study	Doc
F5	*Concerns handled* through adaptation	RQ1
F6	*Concerns affected* by adaptation	RQ1
F7	*Location of the adaptation mechanisms* in the technology stack	RQ1
F8	*Application domain*	RQ1
F9	*Feedback loop* mechanisms	RQ2
F10	*Concerns models* used for or impacted by adaptation	RQ2
F11	*Evaluation method* applied	RQ3
F12	Type of *assurances*	RQ3
F13	*Limitations* of proposed approaches	RQ4
F14	*Strengths* of proposed approaches	RQ4

Table 2: Data collection form

2.2 Conducting

In this phase we set the previously defined protocol in practice. More specifically, we performed the following activities: (i) studies search, (ii) studies selection, (iii) data extraction.

Figure 1 shows our search strategy and the number of the studies being selected or excluded during the subsequent stages.

Based on the approach suggested by [14], the *studies selection* phase has been implemented by two co-located reviewers who assessed a random sample of 50 studies taken from the 783 papers obtained at stage 3 of our search and selection process (see Figure 1). They independently extracted primary studies separately. Reviewer agreement was measured using the Cohen Kappa statistic, and values above or equal to 0.80 have been used as indicator of agreement. Then, the same procedure has been followed again between one of these reviewer and the third reviewer (distributed). The rest of the studies extractions are accomplished by one reviewer and cross checked by the other reviewers in case of doubts. The list of primary studies is available on the survey website.
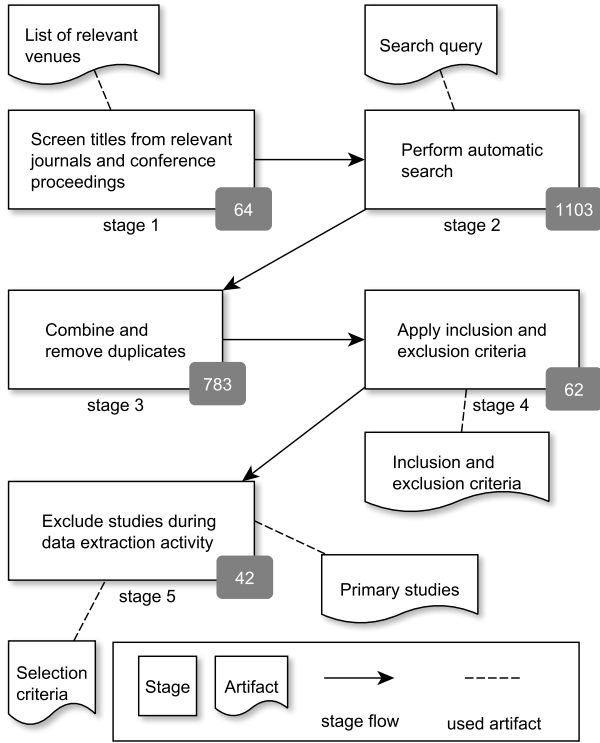
Figure 1: Multi-staged search and selection process

2.3 Reporting

Data extracted from the selected primary studies was collected in a spreadsheet including, for each paper, the following fields: paper ID, paper title (F1), authors, country, abstract, and F2-F14 data items. Sheets have been created for each data item, and for combinations of data items. The spreadsheet is available in the replication package at the survey website.

3. RESULTS

This section provides the results of our survey organized by the four research questions.

3.1 RQ1: How is self-adaptation applied in cyber physical systems?

The answer to RQ1 is derived from quality *concerns handled* (the concerns for which adaptation is applied) (F5), *concerns affected* (the concerns that are affected by adaptation as a side effect) (F6), *feedback loop mechanisms* (such as MAPE, control theory, self-organizing) (F7), and *location technology stack* (for each feedback loop mechanism, its location in the technology stack) (F8).

RQ1.1: Which concerns are addresses by existing approaches?

The *concerns related to self-adaption in CPS* (F5) we found in the studies are quite diverse. We clustered them according to the CPS-specific quality requirements groups discussed in [15]. The most frequent clusters are reported in Figure 2 and discussed below.

The top three concerns related to self-adaption in CPS (F5) are efficiency/performance (66% of the studies), flexibility (48%), and reliability (24%). Configurability/re-configurability, functionality, and interoperability account each for 14% (or less) of the studies. Dependability, profitability, agility, scalability, composability,
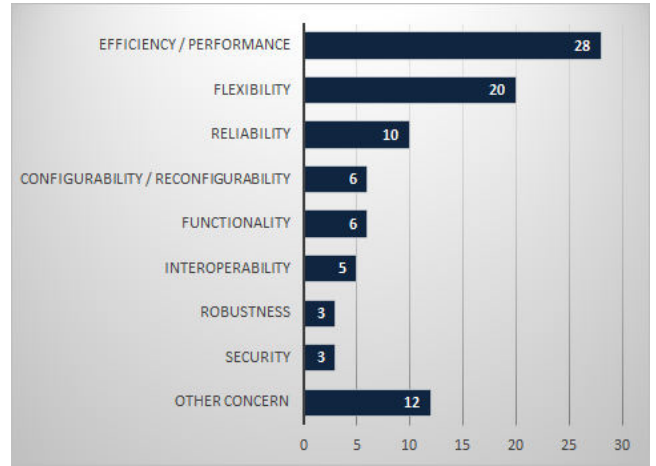

Figure 2: Top concerns related to self-adapting CPS (F5)

maintainability, evolvability, usability, security, and robustness account each for 7% (or less) of the studies. Other reported concerns are reusability, portability, mobility, and accuracy. These concerns are considered in only 9% of the studies in total.

The efficiency/performance category can be further refined in several subcategories: effective operation (54% of the studies which handle efficiency/performance category), energy consumption and knowledge exchange (both 18%), resource utilization (7%), and one study addresses time behavior.

Little attention is given to *concerns that are affected by adaptation as a side effect*, i.e. concerns that with a negative effect on self-adaptation (F6). In particular, only two studies state that self-adaptation in CPS has a negative effect on some concern, and specifically on efficiency and cost (studies [16] and [17]).

RQ1.2: At which level of the technology stack is self-adaptation applied?

Figure 3 shows the frequency of the locations in the technology stack where self-adaptation is applied (F7). The dominant layer for applying adaptation is the application layer with 64% of the studies, and middleware layer (24%). Adaptation is equally applied at the communication and service layer (both 14%). Other minor appearances are reported in the figure.
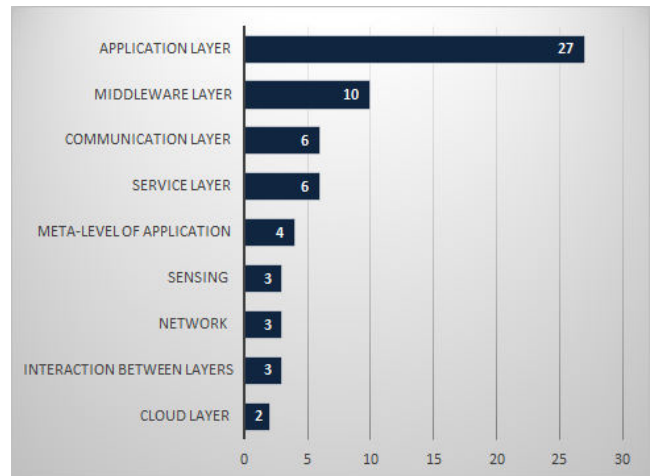

Figure 3: Technology Stack (F7)

RQ1.3: What are the application domains of proposed approaches?

Figure 4 shows the frequencies of application domains (F8). To identify the application domains, we followed the classification provided by the cyberphysicalsystems.org website. The results show that 74% of the studies consider an explicit application domain. In other words, solutions are regularly provided for CPS in general, independent of a specific domain. The dominant application domains are energy (24% of the studies), manufacturing (17%), and transportation (12%).
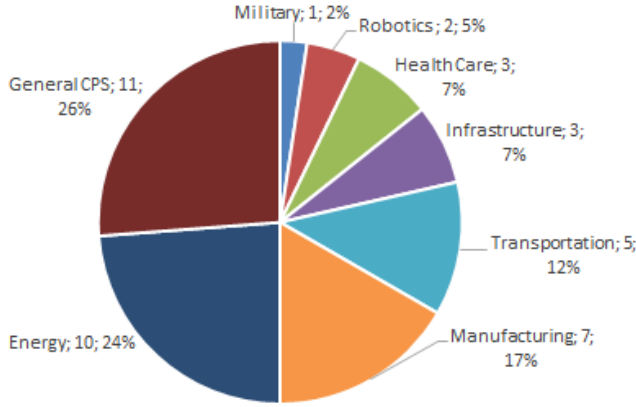

Figure 4: Application Domains (F8)

We further investigated the correlation between the main *quality concerns (F5) and the main types of CPS applications (F8)*. Figure 5 shows the results of this investigation. The table shows that efficiency/performance is relevant to self-adaptation in all primary CPS domains, while flexibility is more related to manufacturing systems (reliability is ignored in any of the related studies).

We looked at the number of concerns considered in individual studies and identified that 19% of the studies consider a single concern, 47% consider two concerns, and the remaining 33% consider more than two concerns. **It is interesting to notice how frequently multiple concerns are handled by the primary studies**. An explanation may be that these domains are characterized by loosely coupled components, in which the applications comprise HW and SW execution units with related but distinct adaptation requirements, handled at different layers of the technology stack.

Application Domains	Efficiency/ Performance	Flexibility	Reliability
Energy	0,70	0,4	0,4
Manufacturing	0,57	1	0
Transportation	0,6	0,6	0,2

Figure 5: Top concerns linked to CPS applications

3.2 RQ2: How do existing approaches for self-adaptation in cyber physical systems handle self-adaptation concerns?

The answer to RQ2 is derived from *feedback loop mechanisms applied* (F9) and *concern models used for or impacted by adaptation* (F10).

RQ2.1: What types of feedback loops are applied?

Figure 6 shows that MAPE (Monitor-Analyze-Plan-Execute) is the dominant adaptation mechanism applied in the primary studies

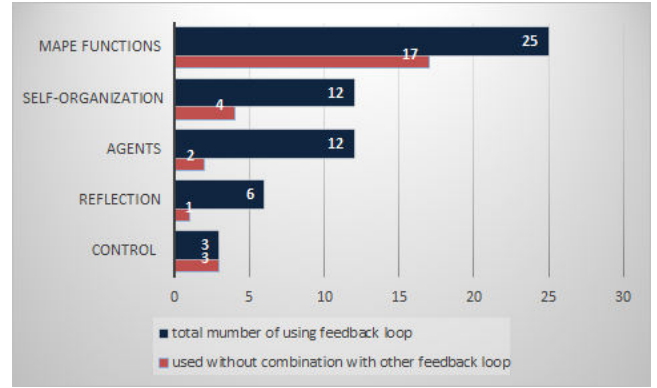(60%), followed by agents and self-organization (both 29%), then reflection 16%, and finally control (7%).

Figure 6: Feedback loops Mechanisms (F9)

Remarkably, 36% of the studies combine different adaptation mechanisms to realize adaptation: 16% combine MAPE with agents, 9% combine reflection with self-organization, while the rest of combined adaptation mechanisms (13%) combine agents with self-organization, agents with reflection and self-organization, and again MAPE function with self-organization (see Figure 7).

RQ2.2: What models are used to represent concerns?

Regarding the models used while architecting self-adaptive CPS, we observe that the most frequent model-type is a multi-agent model (in 10 studies) followed by ontological and goal models (in 6 studies each). Other type of models are used in three or two studies (see Figure 8), while there is a long list of other type of models (in 21 studies) that we could not cluster in any reasonable way. They range from bio-inspired models, mathematical formulations, proxy-based models, Markov decision processes, and many more.

Figure 7: Multi Feedback loops Mechanisms (F9)

Considering the high frequency of multi-agent models, we went through these studies to identify finer-grained concern models. We found an equal distribution of the FIPA (foundation for intelligent physical agents) model, task allocation, negotiation, and planning (each two studies). Self-organized, blackboard-based coordination, and group formation models are used in one study each.

3.3 RQ3: What type of evidence is provided by existing approaches for self-adaptation in cyber physical systems?

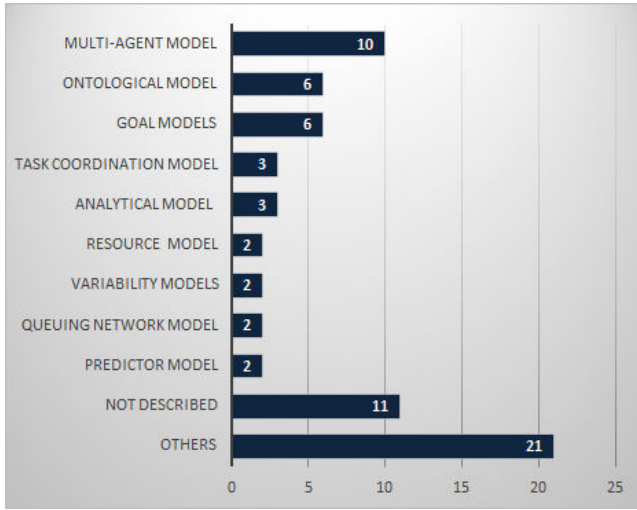The answer to RQ3 is derived from the *evaluation method applied* (F11).

Figure 8: Concern Models (F10)

RQ3.1: What types of empirical methods are applied?

Most of the primary studies use illustrative examples for evaluation (64%). 12% of the studies do not use any type of evaluation to validate their work. Only 3% are using case studies, besides 2% using prototypes and another 2% using experiments.

RQ3.2: What types of assurances are provided?.

Concerning assurances, 21% of the studies provide some level of evidence for claims using simulation. 11% of the studies use some form of consistency checking, formal methods, experimental results, and emulation. In the majority of studies (68%), no assurance is provided at all.

The results for RQ3 confirm earlier results in architecture-based adaptation [6] that the evidence provided by studies is often obtained from applying the research results to toy examples.

3.4 RQ4. What are the strengths and limitations of proposed approaches?

The answer to RQ4 is derived from the *limitations* (F11) and *strengths* (F12) of proposed approaches.

Most of the studies point out the strengths of the proposed approaches. We report a representative set. QoS, optimization (of various factors), performance, and interoperability are the most frequently claimed strengths. QoS improvement is claimed by nine studies. Some of the QoS improvements are related to new sensing services ([18], [19]), while others focus on automated configuration and re-configurability for QoS improvement ([20] and [21]). Other studies focus on flexibility enhancement ([20], [22]), integration, efficiency, agility, and few others. Optimization is claimed to be the main strength in 7 studies. Two of those studies are on energy optimization ([23], [24]), while resources optimization is the main strength of [25] and [26]. Two other studies are about energy saving ([27], [28]) even if their main focus is not on optimization. Performance improvement and optimization (in combination with other QoS) is claimed to be the strength of 4 studies. Interoperabilty as a means to reduce costs is the main strength of [29] and [30].

On the opposite, very limited information is reported about limitations of proposed approaches in the primary studies (F13). Study [31] reports on limitations of the proposed approach related to the formulation of relevant laws and regulations and moves this challenge to future work. Study [32], proposing a calculus of a Context Aware Ambients model, reports that it did not consider security concerns. Study [33] points out the complexity on CPS time management, and argues for improvements. Study [22] reports on the lack of systematic procedure for learning and optimization in shared circuits modeling. The need of better mechanisms for interoperability is advocated in [34], while [35] reports a need for more comprehensive approaches for testing CPS.

4. DISCUSSION

While analyzing the data items F1-F14 individually, our attention was captured by the role of multiple concerns (F5), multiple feedback loops (F9), and the location in the technology stack (F7). They all highlight that different concerns are handled at a time, by using different types of feedback loops, distributed across multiple layers of the protocol stack. These correlations seem to be directly related to the distributed, multi-view nature of CPS. We provide a deeper investigation on the correlation between these data items.

By cross-linking feedback loops type (F9) with the technology stack (F7), we obtained the results presented in Figure 9.

Let us focus first on the left part of the figure. The results tell us that MAPE is used along all the layers of the technology stack we identified, except for the network layer. In most of the cases (56%) MAPE is used at the application layer, while their application to the middleware layer and service layer drop to 12% of the cases. Self-organizing feedback loops are applied in a number of layers as well (with a concentration in the application layer), while agents, reflection, and control are more associated to specific layers. About agents, 67% are applied at the application layer.

The right part of the figure (starting from the self-organization & agents & reflection column onwards) shows how different adaptation mechanisms are distributed to different layers (36% of the studies combine different mechanisms, as already discussed in Section 3.2). The figure shows that mechanisms are distributed over two to four different layers (except for [16] where self-organization & agents & reflection are all applied at the Application Layer).

FEEDBACK LOOP TYPE / TECHNOLOGY STACK	CONTROL	REFLECTION	AGENTS	SELF-ORGANIZATION	MAPE FUNCTIONS	self-organization & agents & reflection	self-organization & MAPE	self-organization & agents	reflection & self-organization	MAPE & agents
CLOUD LAYER					2					
CONNECTION BETWEEN LAYERS				1	3		1			
NETWORK LAYER		1		2					1	
SENSING LAYER			1	1	2			1		
META-LEVEL OF APPLICATION			1		3					
SERVICE LAYER			2	3	5		1			1
COMMUNICATION LAYER		1	2	2	3			1	1	2
MIDDLEWARE LAYER	2		2	2	5			1		1
APPLICATION LAYER	1	4	8	5	14	1		1	3	5

Figure 9: Feedback loops Mechanisms linked to Location of Technology Stack (F9+F7)

We then related feedback loop mechanisms (F9) with the concerns handled (F5). We noticed that 68% of the studies using MAPE tackle efficiency/performance, while 44% tackle flexibility, and 32% reliability. Similarly, for self-organization loops, 66% of them tackle efficiency/ performance, and 58% tackle flexibility. About agents, 50% solved efficiency/performance problem while 41% tackled flexibility. Reflection and control are also used to handle efficiency/performance and flexibility but less frequently.

Although MAPE is dominant in the surveyed studies, it is not used to handle accuracy, reusability, portability, mobility, and robustness concerns. Accuracy and reusability on the other hand are rarely addressed, and only with control feedback loops. Portability and mobility are handled with reflection and self-organization. Only flexibility and efficiency/performance are covered by all feedback loops mechanisms. Another interesting remark is that interoperability, functionality and configurability/re-configurability are all mechanisms except control loops. Referring to the different combinations of feedback loops, we noticed that the majority of concerns were tackled by MAPE.
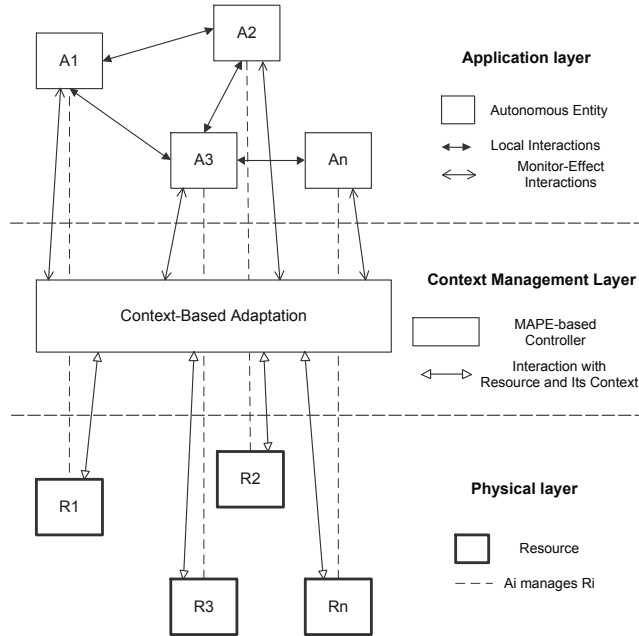

Figure 10: Generalized Three-Layer Adaptation Model

Several of the data items that we studied in this survey were also studied in a recent survey on the application of self-adaptation [36]. However, that survey focussed on the application of MAPE-based approaches to self-adaption for software engineering in general. Comparing the results of both studies, we observe that the concerns related to self-adaptation in CPS match very well with the concerns related to self-adaptation in software engineering in general. An important difference in the survey results is the combined use of multiple feedback loop mechanisms. Whereas [36] does not report any study that combines MAPE-based adaptation with another adaptation mechanism, there are a significant number of studies that combine different adaptation mechanisms to realise self-adaptation in CPS. On the other hand, the results for level of evidence are very similar in both surveys (see answer to RQ3), and the same applies to reporting strengths and limitations; most researchers in the studies of both surveys report the strengths of newly proposed approaches, but neglect limitations.

One of the key insights derived from this survey is that self-adaptation in CPS often combines different adaptation mechanisms that may span multiple layers. Figure 10 provides a generalized three-layer model that we derived from 13 primary studies. At the bottom, we have the physical layer that comprises resources situated in a context. Examples are vehicles ([37], [31]), robots ([38]), and sensor infrastructure ([39]). The context management layer consists of a distributed MAPE-based controller that can access the physical resources and their context. This middle layer offers context related adaptation services. The context management layer can be realized in different ways, e.g. as a middleware platform ([37]), a component framework ([39]), or as a cloud service ([31], [38]). Context management is based on MAPE-based adaption, where distributed feedback loops work together. The context management layer offers adaptation services to the upper layer, such as robust communication ([38]), resource optimization ([31]), effective group formation ([37]), and resource failure handling ([39]). Context management provides context-dependent services to the application layer beyond the local scope of single resources. Finally, the application layer comprises collaborating autonomous entities that manage underlying resources and providing services to users. For example, in [39] agents monitor water resources and manage water distribution infrastructure to avoid that polluted water reaches citizens. In [37], vehicle agents form dynamic ensembles based on context to effectively allocate free parking spaces in a smart city setting; a related approach is applied in [31]. In [38], a set of robots use a gossip approach to share information with each other. Each robot has a corresponding clone in the cloud. A task can be executed by the robot itself or offloaded to its clone. This approach allows for sporadic outage in the physical network.

The main treats to validity of this survey relate to bias in decisions of whether a paper applies self-adaptation to CPS or not, differences in understanding among the reviewers about the search and analysis methods, poor description of some data items which required inferring of certain pieces of information during data collection, and completeness of our search results in terms of the time scope. A discussion of these threats is available on survey website.

5. CONCLUSIONS

Cyber Physical Systems are by nature self-adaptive; they are required to use feedback loop mechanisms to deal with the various sources of uncertainty, control their emergent behavior, and be resilient to changes. As reported in this survey, a number of studies have been using self-adaptation features in the engineering of CPS. What can the self-adaptive community learn from those studies, and how can best practices from the SEAMS community may help to engineering better self-adaptive CPS?

A key insight of this survey is that adaptation in CPS is a cross-layer concern, where solutions combine different adaptation mechanisms within and across layers. The use of multiple feedback loops in the same CPS is growing at a substantial rate, still, feedback loop mechanisms are mostly applied at the application layer. Moreover, while "traditional" (to embedded systems) concerns such as performance and reliability are well covered, CPS most challenging concerns such as interoperability and security are still barely covered by the literature. This raises various challenges for future research, both in the field of CPS and self-adaptation, including: how to map concerns to layers and adaptation mechanisms, how to coordinate adaptation mechanisms within and across layers, and how to ensure system-wide consistency of adaptation.

The self-adaptive community has therefore the unique opportunity to expand its methods and tools towards the new dimensions required for properly engineering self-adaptive CPS.

6. REFERENCES

[1] EU Horizon 2020, "Smart cyber-physical systems," ICT-01-2016. http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/5085-ict-01-2016.html.

[2] E. Lee, "Cyber physical systems: Design challenges," in *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing*, 2008.

[3] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Machine Learning in Cyber Trust*. Springer US, 2009, pp. 3–13.

[4] K.-D. Kim and P. Kumar, "Cyber physical systems: A perspective at the centennial," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

[5] T. Bures *et al.*, "Software engineering for smart cyber-physical systems – towards a research agenda," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 6, 2015.

[6] D. Weyns, M. U. Iftikhar, S. Malek, and J. Andersson, "Claims and supporting evidence for self-adaptive systems: A literature study," in *SEAMS*, 2012.

[7] T. Patikirikorala *et al.*, "A systematic survey on the design of self-adaptive software systems using control engineering approaches," in *SEAMS*, 2012.

[8] I. Malavolta, H. Muccini, and M. Sharaf, "A preliminary study on architecting cyber-physical systems," in *European Conference on Software Architecture Workshops*, 2015.

[9] J. Kramer and J. Magee, "Self-managed systems: An architectural challenge," in *FOSE*, 2007.

[10] B. Cheng *et al.*, "Software engineering for self-adaptive systems: A research road map," ser. LNCS, vol. 5525. Springer, 2009.

[11] R. de Lemos *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," ser. LNCS, vol. 7475. Springer, 2013.

[12] D. Weyns, S. Malek, and J. Andersson, "Forms: Unifying reference model for formal specification of distributed self-adaptive systems," *ACM TAAS*, vol. 7, no. 1, 2012.

[13] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Tech. Rep. EBSE-2007-01, 2007.

[14] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*, ser. Computer Science. Springer, 2012.

[15] G. Sapienza, I. Crnkovic, and P. Potena, "Architectural decisions for HW/SW partitioning based on multiple extra-functional properties," in *WICSA*, 2014.

[16] M. Sadeghi, "Automatic iron cutting device using IEC61499 FBs editor," in *International Conference on Signal processing, robotics and automation*, 2010.

[17] L. Zhang, "Convergence approach to model physical world and cyber world of aviation cyber physical system," in *Dependable, Autonomic and Secure Computing*, 2014.

[18] A. Taherkordi and F. Eliassen, "Towards independent in-cloud evolution of cyber-physical systems," in *Cyber-Physical Systems, Networks, and Applications*, 2014.

[19] W. Kang, K. Kapitanova, and S. H. Son, "Rdds: A real-time data distribution service for cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, 2012.

[20] A. V. Barcnji *et al.*, "An rfid-enabled distributed control and monitoring system for a manufacturing system," in *Innovative Computing Technology*, 2013.

[21] M. U. Tariq, S. Grijalva, and M. Wolf, "Towards a distributed, service-oriented control infrastructure for smart grid," in *International Conference on Cyber-Physical Systems*, 2011.

[22] S. Iarovyi *et al.*, "From artificial cognitive systems and open architectures to cognitive manufacturing systems," in *International Conference on Industrial Informatics*, 2015.

[23] S. R. Valsalam, A. Sathyan, and S. Shankar, "Distributed scada system for optimization of power generation," in *Annual IEEE India Conference*, 2008.

[24] M. Nasri, H. Farhangi, A. Palizban, and M. Moallem, "Multi-agent control system for real-time adaptive vvo/cvr in smart substation," in *Electrical Power and Energy Conference*, 2012.

[25] W. Liu and J. Su, "A solution of dynamic manufacturing resource aggregation in cps," in *Information Technology and Artificial Intelligence Conference*, vol. 2, 2011, pp. 65–71.

[26] K. Wan and V. Alagar, "Achieving dependability of cyber physical systems with autonomic covering," in *Dependable, Autonomic and Secure Computing*, 2014.

[27] M. Engelsberger and T. Greiner, "Software architecture for cyber-physical control systems with flexible application of the software-as-a-service and on-premises model," in *IEEE International Conference on Industrial Technology*, 2015.

[28] V. Rao *et al.*, "On systems generating context triggers through energy harvesting," *Communications Magazine, IEEE*, vol. 52, no. 6, pp. 70–77, 2014.

[29] G. González, C. Angulo, and C. Raya, "A multi-agent-based management approach for self-health awareness in autonomous systems," in *ICAC*, 2007.

[30] M. Sadeghi, "Automatic iron cutting device using IEC61499 FBs editor," in *International Conference on Signal processing, Robotics and Automation*, 2010.

[31] J. Wan *et al.*, "Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions," *Communications Magazine*, vol. 52, no. 8, 2014.

[32] A. Gupta *et al.*, "Towards context-aware smart mechatronics networks: Integrating swarm intelligence and ambient intelligence," in *Issues and Challenges in Intelligent Computing Techniques*. IEEE, 2014, pp. 64–69.

[33] Y. Park and D. Min, "Design and implementation of m2m-hla adaptor for integration of etsi m2m platform and ieee hla-based simulation system," in *Computational Intelligence, Modelling and Simulation*, 2013.

[34] P. Maló *et al.*, "Self-organised middleware architecture for the internet-of-things," in *Green Computing and Communications and Internet of Things*, 2013.

[35] G. Fortino *et al.*, "Integration of agent-based and cloud computing for the smart objects-oriented iot," in *Computer Supported Cooperative Work in Design*, 2014.

[36] D. Weyns and T. Ahmad, "Claims and evidence for architecture-based self-adaptation: A systematic literature review," in *Software Architecture*, ser. Lecture Notes in Computer Science, 2013, vol. 7957.

[37] M. Kit *et al.*, "An architecture framework for experimentations with self-adaptive cyber-physical systems," in *SEAMS*, 2015.

[38] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *Network*, vol. 26, no. 3, 2012.

[39] C. Yu, S. Jing, and X. Li, "An architecture of cyber physical system based on service," in *International Conference on Computer Science & Service System*, 2012.