

Original Article

Open Access



Development complexity of Cyber-Physical Systems: theoretical and practical benefits from Pre-Integrated Architectures

Philippe Gougeon¹, Etienne Hamelin²

¹Innovation & Collaborative Research, Valeo Comfort and Driving Assistance, Creteil 94000, France.

²LIST, CEA, Université Paris-Saclay, Palaiseau 91120, France.

Correspondence to: Dipl. Eng. Philippe Gougeon. Innovation & Collaborative Research, Valeo Comfort and Driving Assistance, 6 rue Daniel Costantini, Creteil 94000, France. E-mail: philippe.gougeon@valeo.com

How to cite this article: Gougeon P, Hamelin E. Development complexity of Cyber-Physical Systems: theoretical and practical benefits from Pre-Integrated Architectures. *J Smart Environ Green Comput* 2023;3:3-17. <https://dx.doi.org/10.20517/jsegc.2022.17>

Received: 8 Aug 2022 **First Decision:** 29 Sep 2022 **Revised:** 24 Nov 2022 **Accepted:** 8 Mar 2023 **Published:** 14 Mar 2023

Academic Editors: Vincenzo Loia, Witold Pedrycz **Copy Editor:** Ke-Cui Yang **Production Editor:** Ke-Cui Yang

Abstract

Aim: The growing complexity of Cyber-Physical Systems (CPS) impacts resources and development time. Hence, numerous component-based design approaches have been developed to mitigate complexity and, consequently, the R&D effort. But is complexity even measurable? The aims of this paper are: to contribute to the CPS and Product Line Engineering (PLE) fields of research; to understand and apply the tools estimating CPS complexity; to evaluate the benefits of Pre-Integrated Architectures (PIARCHs) from the point of view of CPS complexity.

Methods: Based on prior studies, we found that complexity is measurable. We used a structural complexity metric to calculate the impact of PIARCHs by creating variants of a given system architecture. In a practical application project, we used PIARCHs on two types of use cases: generic ones, like localization and perception, and a highly specific one: Urban Automated Driving.

Results: Based on the calculation established by complexity metrics, PIARCHs reduce complexity. This has been revealed in theoretical and practical approaches. Generic use cases like localization and perception of an automated vehicle have more benefits with PIARCHs than the complex Urban Automated Driving use case. This can be explained by the fact that the number of inputs and parameters is smaller, and after the initial investment,



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



the field of applications is wider.

Conclusion: Project complexity is measurable, and the impact of PIARCHs mitigating complexity can be assessed. Their impact varies according to the complexity of the use case and the width of the field of applications. A minimum of complexity is required to justify the initial investment. However, an excessive PIARCH complexity reduces the number of applications and the payback of the initial investment.

Keywords: Cyber-Physical Systems, system architecture, system complexity, CPS4EU project, pre-integrated architectures, automated mobility, product line engineering

INTRODUCTION

Beyond the concept of Cyber-Physical Systems (CPS), their applications are ubiquitous and cover a wide range of domains, such as smart cities, mobility, industry, energy, aerospace, health, consumer electronics, logistics, agrifood, etc. Originally composed of sensors, actuators and computation devices, the deployment of technologies like artificial intelligence, wireless connectivity, edge computing, cloud computing, and robotics significantly improved the performances of these CPS. A wider and heterogeneous ecosystem became necessary to develop and manage the new CPS and their operations.

Recently, societal expectations in the fields of cybersecurity, data privacy, ethics, liability, or CO₂ neutrality, require additional and specific competences, drastically increasing the complexity of the CPS developments.

Facing these new challenges, the companies developing these new CPS often underestimate the time, workforce, and resources needed to reach these increased performances and higher expectations. The discrepancy between the profitability constraints and the R&D efforts required by new CPS raises questions about their sustainability, even for large enterprises. Access to these technologies also becomes more difficult for a broad base of small and medium enterprises. An example of this widening gap in the automotive industry is illustrated in the study by Fletcher *et al.*^[1]. Globally, evaluating the complexity and the risks associated with this new kind of CPS development becomes a crucial question for program directors and R&D experts.

Assessing this complexity allows us to define sustainable business goals, and the associated strategic decisions enable economies of scale and reduction of risks. If the literature review from Section “Prior work related to system complexity” gives useful concepts, CPS developers need a more global view of the questions mentioned here above, more adapted to the context of CPS, with more quantified practical approaches. Therefore, our purpose within this present research is also to connect the topics of system complexity, CPS and Product Line Engineering.

A previous publication introducing the developments performed in the CPS4EU project^[2] formulated several solutions to mitigate the growing complexity of new CPS developments. That paper proposed the Pre-Integrated Architecture (PIARCH) concept as a way to mitigate this complexity increase, with illustrations of this concept based on the work performed in the CPS4EU project, see [Figure 1](#).

As stated in^[2], the PIARCH concept sits between components and application systems. A PIARCH integrates components and allows the exploration of a set of possible instances or products, with components variations and market adaptations. As a reuse-oriented concept, it aims at mutualizing the development work between similar but different products in a networked ecosystem. The following Section

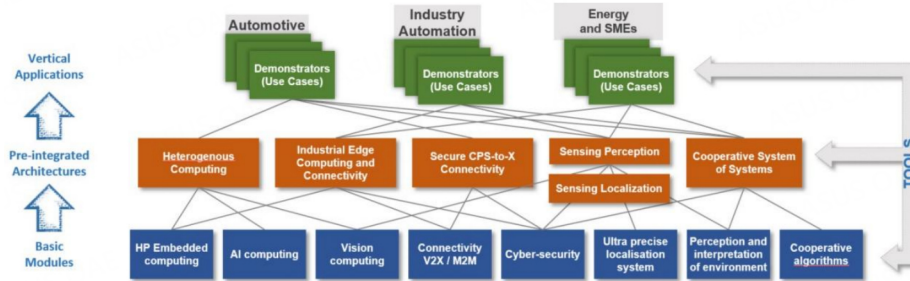


Figure 1. The CPS4EU project overview; Basic Modules, Pre-Integrated Architectures, Vertical Applications, and Tools.

“Proposed theoretical approach” also recalls the main principles behind this Pre-Integrated Architecture concept.

The present paper aims to contribute to the CPS and Product Line Engineering fields of research. Therefore, this publication will provide potential solutions to the R&D experts, the system engineers, the system architects, the project managers and the program directors to more efficiently develop their future products based on complex CPS. The proposed solutions answer two research questions related to the same topic.

- RQ1: Which available tools allow us to estimate the R&D effort required to develop a new and complex CPS?
- RQ2: What benefits can we expect from the Pre-Integrated Architectures developed in the CPS4EU project?

This paper builds together a theoretical formulation and a practical approach to these questions. The theoretical formulation provides an adaptation of system architecture tools to the CPS context. The practical approach is based on several use cases developed in the CPS4EU project.

To provide more details and propose answers to the research questions RQ1 and RQ2, this paper is organized as follows:

- Section “METHODOLOGY”: Methodology, prior work and theoretical approach
- Section “RESULTS”: Results, practical applications
- Section “DISCUSSION”: Discussion
- Section “CONCLUSIONS”: Conclusions

METHODOLOGY

Prior work related to system complexity

The topic of system complexity and CPS has generated a huge amount of research publications. The following non-exhaustive summary gives some highlights about this prior work:

Starting this review with system complexity, Garbie^[3] provides an extensive review of complexity metrics of industrial systems. Two main forms of system complexity are outlined: (a) static or structural complexity designed into the system architecture and (b) operational or dynamic complexity. More than forty different complexity metrics are listed in this literature review. Gomes *et al.*^[4] also present an interesting state of the art about system complexity. Then, they consider a system complexity metric based on its connections, and a system sensitivity metric, with two examples corresponding to a distribution center and a medical center.

In^[5], H.Simon presents the links between the theory of complex systems and the theory of hierarchy. This publication also includes the famous parable of the two watchmakers as an example of the benefits associated with stable sub-system decompositions. Braha *et al.* in^[6] propose two categories of system complexity: (a) a structural complexity and (b) a functional complexity, a function of its probability of successfully achieving the required specifications. The notion of effort complexity measure is also introduced in this document. Reilhac *et al.*^[7] mentioned the concept of modular design applied to automotive CPS. To illustrate the optimization function guiding the design process, an analogy with the Tetris game is suggested. Suh^[8] proposed a classification with four different types of complexity in engineering: time-independent real complexity, time-independent imaginary complexity, time-dependent combinatorial complexity, and time-dependent periodic complexity. Complexity is viewed as a measure of uncertainty in achieving the specified functional requirements. Sinha and de Weck^[9] develop a quantitative measure for structural complexity and apply this complexity measure to real-world engineered systems like gas turbine engines. For a given system, the methodology proposed in this paper distinguishes 3 contributions to complexity: the complexity which arises from each of its components, the complexity related to their interfaces, and eventually the complexity inherent to the system structure. This architecture is described by a graph and its adjacency matrix similar to a Design Structure Matrix. A relationship between the structural complexity metric and the system development cost is also included in this paper.

In^[10], Koto *et al.* present the advantages of using the Dependency Structure Matrix to optimize the design of large mechanical systems. An application to offshore structures supports this objective. This paper also provides interesting properties of the DSM, with a direct positive impact on reducing the complexity of the mechanical systems. Jacobs^[11] proposes another complexity metric with a different perspective corresponding to a product portfolio. Here, a function of product multiplicity, diversity and inter-relatedness generates the Generalized Complexity Index. Again, the product structure diagram is the foundation for the calculation of this complexity metric. Alkan *et al.* transposed in^[12] the principle mentioned in^[9] to the domain of system manufacturing. Two applications illustrate this adaptation of the product assembly complexity metric and a correlation is again proposed with the corresponding assembly time.

Concerning the subject of CPS, the paper from the NIST^[13] provides useful definitions and practical views of generic CPS with three layers corresponding to the physical layer, the cyber layer and the internet of CPS layer. This document proposes a full framework covering the entire life cycle of a CPS from the business case, design, development, manufacturing, operation and decommissioning. Sheikh *et al.*^[14] also formulate a CPS decomposition in three layers (perception, transmission, application) and four types (time-based, event-based, lattice-based event, and hybrid). This CPS structure supports an extensive cybersecurity analysis, covering the different security goals and the different possible types of attacks. The authors suggest that Artificial Intelligence methodologies can provide good solutions to reach the security goals linked to the CPS. Sampayo and Peças^[15] target a reduction of development cost and time for the CPS, taking into account the business context. Using the 8C model for the CPS, and with a focus on the industrial domain, a design and development methodology is proposed in six phases, from design to production ramp-up. The

CPS architecture, the list of CPS features and the CPS implementation plan are the main outputs of this new methodology adapted to CPS. In^[16], Thakur and Sehgal propose a new architecture for industrial CPS. This paper gives a detailed perspective oriented towards the optimization of the CPS computing resource usage, and several aspects related to industrial CPS control and stability. Again, the architecture of the CPS is described by a graph and its adjacency matrix, and a formula for a modularity metric is also provided.

To finish this review of the prior work related to our paper, we would like to mention two documents promoted by INCOSE, the International Council on Systems Engineering, and more specifically, by its working group on Product Line Engineering^[17]. The working group sets clear objectives for reducing systems development cost and time to market. They formulated the concept of Feature-based Product Line Engineering, built upon a feature portfolio, a shared asset superset, a PLE factory configurator, and product asset instances. This work has prepared the process for a new standard ISO 26580^[18], providing ‘industry-validated implementation guidance and patterns for Feature-based Product Line Engineering’. The INCOSE white paper^[19] provides practical steps towards the implementation of the concept described in^[17]. Systems are described as simple, complicated, complex, and complex adaptive. Fourteen characteristics contribute to identifying complex systems and fifteen guiding principles are suggested to implement this feature-based PLE concept. Practical guidelines allow to address system complexity, and modeling methods for complex systems are recommended.

Based on this non-exhaustive list of prior work related to our research questions, we confirm the suitability of the CPS formalism defined by the NIST^[13]. We also propose to extend the structural complexity metric formula explained in^[9] to the field of CPS. Finally, we identify a strong convergence between the approach proposed in this paper and the Feature-based Product Line Engineering concept outlined by INCOSE^[17].

Proposed theoretical approach

As illustrated in [Figure 1](#), the main goal of the CPS4EU project is to implement a new approach to developing complex CPS in a more sustainable way, keeping the R&D efforts under manageable limits.

Recall here the concept of PIARCH, which is presented very briefly in the introduction. A PIARCH is:

- a configurable set of components, integrated and packaged as a coherent sub-system, which might cover from design patterns to software, electronic or mechanical parts,
- which enables together some specific emerging property of interest (i.e. some added value that stems from their interaction, but is not a direct consequence of any of its components),
- with accompanying documentation, examples, configuration guides, safety and security manuals when applicable,
- which targets reuse in multiple CPS products or projects, possibly within a single-company product line or across multiple partners, typically if an open-source or commercial offer is structured.

This concept of PIARCH was implemented in the CSP4EU project, where six PIARCHs were developed, then used and shared to build the sixteen project use cases. Thanks to the modular reusability aspect, PIARCHs enable to share the cost of pre-integration among several CPS products.

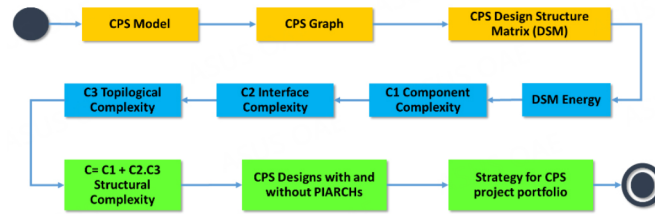


Figure 2. Methodology for the theoretical approach.

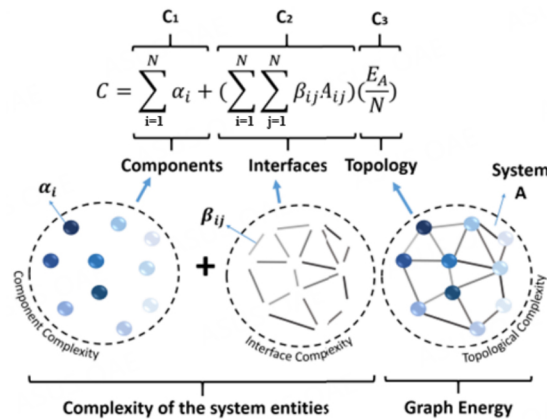


Figure 3. Elements of the structural complexity, from^[12].

In this paper, we evaluate the impact of PIARCHs reusability on reduction of use case complexity, and quantify this impact using the approach based on the Design Structure Matrix^[9].

For the perspective of PIARCH benefits, Figure 2 illustrates in its two final steps the outline of a strategy when a product line of CPS is considered. This strategy is based on an estimation of the initial investment to create the PIARCHs, and the follow-up benefits related to the usage of the PIARCHs in the following CPS project developments.

CPS systems rely on many forms of interactions between their components, or sub-systems: from mechanical or electrical interaction, various physical flows are specifically important when involving sensors and actuators; various forms of electronic or even radio signaling are used to build network interfaces; and software interactions take various forms between software components, tasks, libraries, services, etc.

Figure 3 below shows the formulation of the structural complexity of a system^[9]:

The terms included in this formulation are the following:

C: Structural complexity

C1: Component complexity

C2: Interface complexity

C_3 : Topological complexity

N : Number of components in the system

A_{ij} : Adjacency matrix representing the connectivity structure of the system

α_i : Component complexity

β_{ij} : Interface complexity

E_A : Energy of the graph modeling the architecture of the system

For the calculation of the term E_A , we use the formulation defined by Woods^[20], which states that “the energy of a graph is equal to the sum of the absolute value of the eigenvalues of its adjacency matrix”.

When a PIARCH is used to build p distinct CPS projects, the pre-integration costs can be shared among all p projects. Assuming comparable levels of performance and requirements, in order to define the potential benefits of a strategy using PIARCHs, the following terms are defined:

C_b : CPS complexity baseline without using the PIARCH

C_{pi} : CPS complexity of the initial PIARCH

C_p : CPS complexity using the PIARCH

p : number of projects using the PIARCH in the portfolio

C_{pp} : average CPS complexity for p projects using the PIARCH in the project portfolio

$$C_{pp} = C_{pi}/p + C_p$$

Notice here that the terms C_b, C_{pi}, C_p refer to our reply proposal to RQ1, applied to complex CPS.

Similarly, the term C_{pp} refers to the result of the strategy implementing the PIARCHs in the CPS project portfolio, and therefore formulates a reply proposal to RQ2.

RESULTS

This section illustrates the practical application of the methodology defined here above to the Pre-Integrated Architectures from the CPS4EU project.

In the CPS4EU project, the partners develop sixteen CPS use cases across four domains: automotive, industry, energy distribution and SME applications^[2]. Even if the CPS4EU project provides a large panel of possibilities to test and evaluate the theoretical approach proposed in Section “Proposed theoretical approach”, we selected for this practical example the systems implemented for the automotive use cases.

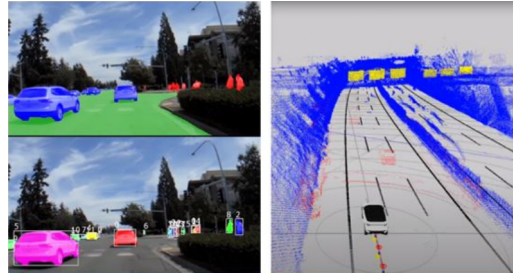


Figure 4. CPS4EU use cases: (A) AI for perception and (B) AI for Localization.



Figure 5. CPS4EU use case Urban Automated Driving.

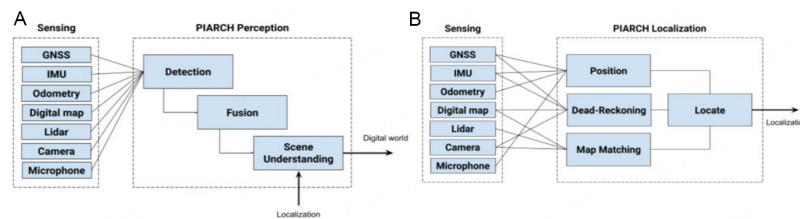


Figure 6. (A) Sensing PIARCH for perception; (B) sensing PIARCH for localization.

We showcase here this methodology on two use cases led by Valeo, illustrative of the challenges of CPS used in driver assistance systems and autonomous driving^[21]. The use cases AI for perception and AI for localization use on-board vehicle sensors, electronic computing units and AI software to provide the necessary object list, vehicle pose and context interpretation so that the equipped vehicle can navigate safely. The use case Urban Automated Driving is built on top of the use cases mentioned above and performs an Automated Driving function. In this use case, the main software modules include perception, localization, trajectory planning, generation of vehicle commands, interactions with the driver and the passengers, diagnostic of the system, supervision of the function execution, and data recordings.

These use cases are illustrated in [Figures 4](#) and [5](#) below.

To simplify the reasoning, we limit here our practical application to interactions in the form of data streams.

CPS complexity baseline without using the PIARCH

For each use cases, the baseline architectures are encoded in their Design Structure Matrix A_{ij} presented in [Tables 1](#) and [2](#).

Table 1. AI based perception & localization use case, baseline DSM

| Components | N = 13 | 1 | 3 | 5 | 6 | 7 | 8 | 9 | A1 | A2 | X1 | X2 | X3 | X4 | α_i |
|-------------------------------------|--------|---|---|---|---|---|---|---|----|----|----|----|----|----|------------|
| Vehicle state | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 |
| Navigation interface | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| GNSS | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| Lidars | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 |
| Dense map | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| Cameras | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| HD Map | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| Detect environment | A1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 5 |
| Locate in environment | A2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 5 |
| Vehicle state with lateral position | X1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| Diag_localization | X2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Environment state | X3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| Diag_perception | X4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

The energies of the Design Structure Matrices A_{ij} are determined from the expression defined in ^[20], and their eigenvalues are calculated with the spreadsheets provided by ^[22].

Abstract parameters α_i rank the component complexity, from $\alpha_i = 1$: complexity, to $\alpha_i = 5$ highest complexity, while interface complexities are evaluated through β_{ij} ($\beta_{ij} < 0 < 1$).

A jury of domain experts defined these parameters. Note that, due to the real-time and safety-relevant aspects of these systems, the interface complexity is estimated mostly from the applicable real-time constraints.

For all the following results $\beta_{ij} = 1$ for time critical data flows, and $\beta_{ij} = 0,5$ for operational or tactical data flows.

CPS complexity of the initial PIARCH

Based on experience and state-of-the-art design patterns, we performed a manual clustering of the DSM starting from the baseline CPS; we introduced two reusable PIARCHs, namely Sensing PIARCH for perception and Sensing PIARCH for localization, see [Figure 6](#).

Table 2. Urban Automated Driving use case, baseline DSM

| Components | N = 17 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A10 | a_i |
|-------------------------------|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|-------|
| Vehicle state | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 2 |
| Human Machine Interface | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| Navigation interface | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| Exterior diagnostic interface | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| GNSS | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Lidars | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| Dense map | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Cameras | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| HD Map | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| Detect environment | A1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 5 |
| Locate in environment | A2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 5 |
| Generate vehicle commands | A3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| Plan vehicle movement | A4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 5 |
| Diagnose system status | A5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 |
| Supervise automation system | A6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3 |
| Interact with the driver | A7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| Record data | A10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

The initial complexities C_{pi} of these two PIARCHs are computed using the data provided in [Tables 3](#) and [4](#).

In the following tables, A3P will refer to the Sensing PIARCH for perception, and A3L will refer to the Sensing PIARCH for localization.

CPS complexity using the PIARCHs

Now that the PIARCHs have been defined, the structural complexity C_p of the two use case CPS using the PIARCHs is computed again, using the data provided in [Tables 5](#) and [6](#).

Table 3. PIARCH perception DSM

| Components | N = 9 | 1 | 3 | 6 | 8 | 9 | A1 | A2 | X3 | X4 | α_i |
|-----------------------|-------|---|---|---|---|---|----|----|----|----|------------|
| Vehicle state | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| Navigation interface | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Lidars | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| Cameras | 8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 |
| HD Map | 9 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| Detect environment | A1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 5 |
| Locate in environment | A2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| Environment state | X3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| Diag_perception | X4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

Table 4. PIARCH localization DSM

| Components | N = 8 | 1 | 3 | 5 | 6 | 7 | A2 | X1 | X2 | α_i |
|-------------------------------------|-------|---|---|---|---|---|----|----|----|------------|
| Vehicle state | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| Navigation interface | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| GNSS | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| Lidars | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| Dense map | 7 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| Locate in environment | A2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 5 |
| Vehicle state with lateral position | X1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Diag_localization | X2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

Table 5. AI-based perception and localization use case, PIARCH-based DSM

| Components | N = 2 | A3P | A3L | α_i |
|---------------------|-------|-----|-----|------------|
| PIARCH Perception | A3P | 1 | 0 | 2 |
| PIARCH Localization | A3L | 1 | 1 | 2 |

PIARCH: Pre-integrated architecture.

Status for p CPS projects using the PIARCH in the portfolio

When a PIARCH is reused across several p projects, the structural complexity C_{pi} specific to the PIARCH itself represents an overhead amortized over these projects. The difference between the baseline structural complexity C_b and the average structural complexity with PIARCHs C_{pp} represents the benefit provided by the implementation of PIARCHs in our CPS. The synthesis of the results covering the overall methodology described in this paper is presented in [Figures 7 and 8](#).

Assuming that structural complexity and R&D effort are correlated^[9,12], it appears clearly that packaging a sub-system of a complex CPS as a PIARCH represents an additional effort: complexity is higher for one product designed using PIARCHs. However, as soon as the PIARCH is reused in two or three products, the additional effort is shared among them, and the amortized effort becomes much lower. If the PIARCH can be reused in many projects, then its marginal effort is essentially null.

Table 6. Urban driving use case, PIARCH-based DSM

| Components | N = 12 | 1 | 2 | 3 | 4 | A3P | A3L | A3 | A4 | A5 | A6 | A7 | A10 | α_i |
|-------------------------------|--------|---|---|---|---|-----|-----|----|----|----|----|----|-----|------------|
| Vehicle state | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| Human Machine Interface | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| Navigation interface | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| Exterior diagnostic interface | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| PIARCH Perception | A3P | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| PIARCH Localization | A3L | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 2 |
| Generate vehicle commands | A3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| Plan vehicle movement | A4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 5 |
| Diagnose system status | A5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 |
| Supervise automation system | A6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3 |
| Interact with the driver | A7 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| Record data | A10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

PIARCH: Pre-integrated architecture.

In our practical examples, the total effort reduction reaches -17% for the Urban Automated Driving use case and -71% for the Perception & Localization use case.

DISCUSSION

The results presented above are in line with lessons learned from experienced project managers and effort measurements from similar projects. In particular, the Urban Automated Driving use case requires many more components to be added together with the PIARCHs, whereas the Perception and Localization use case draws most of its added value from the PIARCHs themselves.

This shows that the PIARCH concept indeed enables economies of scale. This concept gives a measurable effort reduction from reusability, and the DSM-based methodology provides an efficient way to estimate quantitatively that effort reduction before the actual development effort.

As planned, those benefits can change in great proportion, as shown in the two use case CPS used for the practical approach. The main parameter influencing the results is the simplification brought by the PIARCHs relative to the overall baseline CPS complexity. Even if the methodology provides mainly comparison

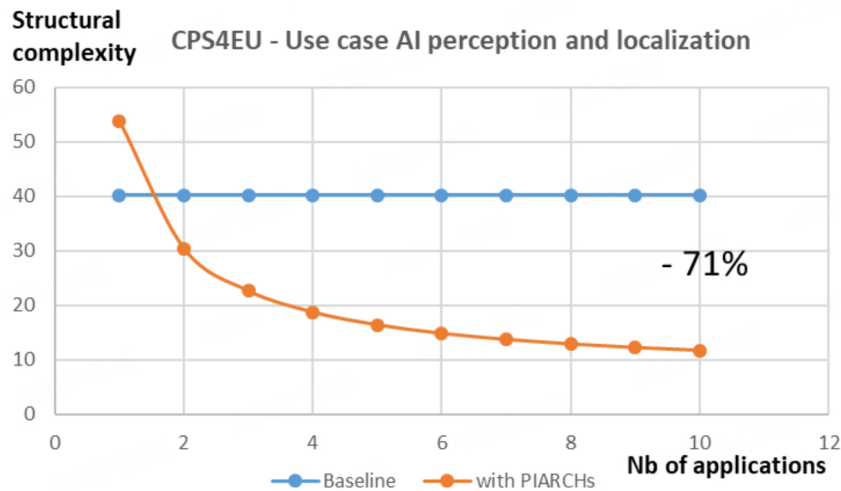


Figure 7. Overall complexity reduction for the Perception & Localization use cases.

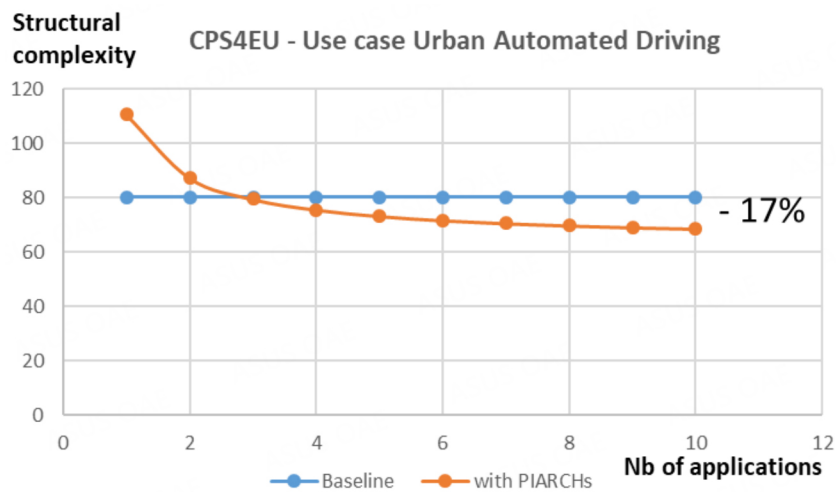


Figure 8. Overall complexity reduction for the Urban Automated Driving use case.

scores, the parameters used to compute the structural complexity are directly accessible to system architects or R&D experts.

Another positive outcome is the fact that this methodology enables the program directors to define an optimum strategy to reach a given R&D effort reduction with mitigated risks. Depending on the number of projects, and their anticipated reuse factor, the program director can determine whether the development effort of the PIARCHs is worth being shared or not within the project portfolio.

CONCLUSIONS

In this paper, we answer the research questions expressed in the introduction. We propose a methodology for the evaluation of project complexity in the field of CPS. Based on two practical examples from the CPS4EU project, we illustrate that the Pre-Integrated Architecture concept deployed in CPS4EU can lead to significant reductions in the structural complexity scores.

The proposed process is generic and applicable to CPS in a wide range of industrial domains. Likewise, large Enterprises, small and medium enterprises, or academics can benefit from Pre-Integrated Architectures involved in their own developments and operations. This methodology can also provide useful indications to the system architects, the R&D experts, and the program directors to maintain a sustainable approach for their future developments of complex CPS.

In a possible further study on this subject, and still, as a contribution to the CPS and Product Line Engineering fields of research, it may be useful to investigate with practical examples the relationship between these structural complexity scores and the R&D efforts needed to develop the new complex CPS.

The authors will also consider proposing a future contribution to the INCOSE working group on Feature-based Product Line Engineering.

DECLARATIONS

Acknowledgments

We warmly thank Ulrike Sinner, Kevin Nguyen and Jean-Denis Piques from Valeo Comfort and Driving Assistance for their advice and improvement proposals during the reviews of this paper.

Authors' contributions

Research questions and review of prior art: Gougeon P

Theoretical approach and practical applications: Gougeon P, Hamelin E

Discussion and conclusions: Hamelin E

Availability of data and materials

The data and the methods supporting this study are available upon request to the authors.

Financial support and sponsorship

This work has been carried out within the CPS4EU project, which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement (No 826276). The JU receives support from the European Union's Horizon 2020 research and innovation program and France, Spain, Hungary, Italy, and Germany. The proposed results reflect only the authors' view. The JU is not responsible for any use that may be made of the information contained in the present work.

Conflicts of interest

Both authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2023.

REFERENCES

1. Fletcher R, Mahindroo A, Santhanam N, Tschiesner A. The case for an end-to-end automotive-software platform. Available from: <https://www.mckinsey.com/~media/McKinsey/Industries/Automotive%20and%20Assembly/Our%20Insights/>

- [The%20case%20for%20an%20end%20to%20end%20automotive%20software%20platform/The-case-for-an-end-to-end-automotive-software-platform.pdf](#) [Last accessed on 10 Mar 2023].
2. Gougeon P, Goubier T, Nguyen K, et al. Pre-integrated architectures for sustainable complex cyber-physical systems. 2021 24th Euromicro Conference on Digital System Design (DSD). *IEEE* 2021:319-24. DOI
 3. Garbie IH. Concepts and measurements of industrial complexity: a state-of-the-art survey. *Int J Ind Syst Eng* 2012;12:42-83. DOI
 4. Gomes VM, Paiva JRB, Reis MRC, et al. Mechanism for measuring system complexity applying sensitivity analysis. *Complexity* 2019. DOI
 5. Simon HA. The architecture of complexity, Proceedings of the American Philosophical Society, Vol. 106, No. 6; 1962. Available from: <https://www.jstor.org/stable/985254> [Last accessed on 10 Mar 2023].
 6. Braha D, Maimon O, Braha D, et al. The measurement of a design structural and functional complexity; 1998. pp. 241-77. DOI
 7. Reilhac P, Eustache J P, Simon L, Deshouillers P. Modular rear closure: an analogy with the 'Tetris'® game for electrical component design and integration. SAE Technical Paper; 2002. DOI
 8. Suh NP. Complexity in engineering. *CIRP Ann* 2005;54:46-63. DOI
 9. Sinha K, de Weck OL. Structural complexity quantification for engineered complex systems and implications on system architecture and design. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. *Am Soc Mech Eng* 2013;55881:V03AT03A044. DOI
 10. Koto J, Efendy MH, Bahadi N. Dependency structure matrix and its application on offshore structures construction. Available from: <https://isomase.org/OCari/E-Learning-5.php> [Last accessed on 10 Mar 2023].
 11. Jacobs MA. Complexity: towards an empirical measure. *Technovation* 2013;33:111-8. DOI
 12. Alkan B, Vera D, Ahmad B, et al. A method to assess assembly complexity of industrial products in early design phase. *IEEE Access* 2017;6:989-99. DOI
 13. NIST. Framework for cyber-physical systems. Release 1.0. Available from: https://s3.amazonaws.com/nist-sgcps/cpspwg/files/pwgglobal/CPS_PWG_Framework_for_Cyber_Physical_Systems_Release_1_0Final.pdf [Last accessed on 10 Mar 2023].
 14. Sheikh ZA, Singh Y, Singh PK, Ghafoor KZ. Intelligent and secure framework for critical infrastructure (CPS): current trends, challenges, and future scope. *Comput Commun* 2022;193:302-31. DOI
 15. Sampayo M, Peças P. CPSD²: a new approach for cyber-physical systems design and development. *J Ind Inf Integr* 2022;28:100348. DOI
 16. Thakur P, Sehgal VK. Emerging architecture for heterogeneous smart cyber-physical systems for industry 5.0. *Comput Ind Eng* 2021;162:107750. DOI
 17. INCOSE. Feature-based systems and software product line engineering: a primer. Available from: <https://conference.confli.com/events/Demo/showcases/incose/INCOSE%20PLE%20Primer.pdf> [Last accessed on 10 Mar 2023].
 18. Online Browsing Platform (OBP). Software and systems engineering - methods and tools for the feature-based approach to software and systems product line engineering. Available from: <https://www.iso.org/obp/ui/#iso:std:iso-iec:26580:ed-1:v1:en> [Last accessed on 10 Mar 2023].
 19. McEver J, Sheard SA, Cook S, et al. A complexity primer for systems engineers. Available from: <https://connect.incose.org/StoreProducts/A%20Complexity%20Primer%20for%20Systems%20Engineers.pdf> [Last accessed on 10 Mar 2023].
 20. Woods C. My favorite application using graph eigenvalues: graph energy. Available from: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=146946acc67db5cd321f5dcc835ad1dff2db9dd9> [Last accessed on 10 Mar 2023].
 21. CPS4EU. Project deliverable D7.3, Automotive use case software integration; Feb 2021, CPS4EU confidential document.
 22. Dcode website. Eigenvalues of a matrix. Available from: <https://www.dcode.fr/matrix-eigenvalues> [Last accessed on 10 Mar 2023].