CPSoSaware: Cross-layer Cognitive Optimization Tools & Methods for the Lifecycle Support of Dependable CPSoS

Georgios Keramidas[1,2], Christos P. Antonopoulos[2], Nikolaos Voros[2], Pekka Jääskeläinen[3], Marisa Catalán Cid[4], Evangelia I. Zacharaki[5], Apostolos P. Fournaris[6], Aris Lalos[6]

[1]School of Informatics, Aristotle University of Thessaloniki, Greece
{gkeramidas@csd.auth.gr}

[2]Electrical and Computer Engineering Department, University of Peloponnese, Greece
{ch.antonop@uop.gr, voros@uop.gr}

[3]Department of Pervasive Computing, Tampere University, Finland
{pekka.jaaskelainen@tuni.fi}

[4]Mobile and Wireless Internet Unit (MWI), i2Cat Foundation, Spain
{marisa.catalan@i2cat.net}

[5]Electrical and Computer Engineering Department, University of Patras, Greece
{ezachar@upatras.gr}

[6]Industrial Systems Institute, Athena Research and Innovation Center in Information Communication and Knowledge Technologies, Greece
{fournaris@isi.gr, lalos@isi.gr}

***Abstract*—Cyber-physical Systems of Systems (CPSoS) are large complex systems where physical elements interact with and are controlled by a large number of distributed and networked computing elements as well as human users. Their increasingly stringent demands on efficient use of resources, high service and product quality levels and, of course low cost and competitiveness on the world market introduce big challenges related to the design operation continuum of dependable connected CPSs.**

The CPSoSaware project aims at developing the models and software tools to allocate computational power/resources to the CPS end devices and autonomously determining what cyber-physical processes will be handled by the devices' heterogeneous components (CPUs, GPUs, FPGA fabric, software stacks). The project relies on Artificial Intelligence (AI) support to strengthen reliability, fault tolerance and security at system level and also to lead to CPS designs that work in a decentralized way, collaboratively, in an equilibrium, by sharing tasks and data with minimal central intervention. The CPSoSaware system will interact with the human users/operators through extended reality visual and touchable interfaces increasing situational awareness. The CPSoSaware system will be evaluated: i) in the automotive sector, in mixed traffic environments with semi-autonomous connected vehicles and ii) in the manufacturing industry where inspection and repair scenarios are employed using collaborative robots.

Keywords—CPS, Embedded Systems, Monitoring and Control Systems, Computer Hardware and Architecture, System of Systems, Automotive, Manufacturing

I. Introduction

In the past few years, the Cyber Physical System domain has been going into a transition phase from individual systems operating isolated to a collection of systems that collaborate in order to achieve a highly complex cause, realizing a system of systems (SoS) approach. There is significant investment in CPSoS both within and outside Europe for domains, like automotive sector, industrial manufacturing, transportation, smart buildings, and industrial processes [2], that have significant impact in European economy and society.

However, there are significant challenges in CPSoS applicability and usability. The fact that even a small CPSoS, (e.g., a connected car) consists of several subsystems and executes thousands of lines of code [3] highlights the complexity of the SoS solution and the need for an approach beyond traditional control and management. In view of this, having a centralized authority that handles all CPSoS processes seems to be very hard to capture and implement. Decentralization of CPSoS processes by appointing tasks to individual CPSs within the SoS can be a reasonable solution, yet still, the collaborative mechanism between CPSs remains a point of research. Therefore, appropriate tools and methodologies are needed in order to assess that functional requirements are retained and the non-functional requirements are matched (i.e., the CPSoS remains resilient, safe, and efficient).

The EU-funded CPSoSaware project [1] envisions the CPSoS as a living organism that behaves autonomously (without human intervention), is aware of its physical and cyber environment and reacts to it accordingly so that it constantly matches its intended purpose. The target of the project is to develop the models and software tools to describe a CPSoS in a holistic and abstract way and to allocate computational power/resources to the CPS devices of the system by determining and generating autonomously how cyber-physical processes will be handled by each heterogenous hardware of the device (general purpose processor cores, GPUs, FPGA fabric) and software components (software and firmware stacks).

The CPSoSaware solution relies on Artificial Intelligence support to strengthen reliability, fault tolerance, and security at system level but also will be able to lead to CPS designs that work in a decentralized way, collaboratively, in an equilibrium, by sharing tasks and data with minimal central intervention. Also, the CPSoSaware system will interact with the CPS/CPSoS human users/operators through extended

2159-3477/20/$31.00 ©2020 IEEE
DOI 10.1109/ISVLSI49217.2020.00-12

reality modules (e.g., through AR glasses and haptics interfaces) to increase human situational awareness, but also to include human behavior in the CPSoS design and operation phase (using human-based reinforced learning of the CPSoSaware Artificial Intelligence (AI)).

II. Challenges in CPSoSs

CPSoSs are heterogenous systems. They are comprised of various and autonomous CPSs, each of them having unique performance capabilities, criticality level, priorities and pursued goals. CPSs in general are self-organized and, in several occasions, they may have conflicting goals thus competing to get access to common resources. However, from a CPSoS perspective, all CPSs must also harmonically pursue system-based achievements and collaborate in order to make SoS based decisions. Considering that a CPSoS consists of many CPSs, finding the methodology to achieve such an equilibrium in a decentralized way is not an easy task.

The collection of data and the data analytics need to be refined in such a way that only the important information is extracted and forwarded to other CPSs and the overall system. Also, mechanisms to handle, in a distributed way, large amount of data are needed in order to extract cognitive patterns and detect abnormalities. Thus, part of data classification, labelling and refinement mechanisms should be put in place locally to offload the complexity and communication overheads [4].

In the above described setup, we cannot overlook the fact that a CPSoS depends on humans since humans are part of the CPSoS services and behavior. Thus, we need to structure a close symbiosis between computer-based systems and human operators/users and constantly enhance human situational awareness as well as devise a collaborative mechanism on handling CPSoS decisions, forcing the CPSoS to comply to human guidelines and reactions. Novel approaches on human machine interfaces, that employ eXtended Reality (XR) principles, need to be devised in order to help humans to grasp insight of the CPSoS processes, but also to enroll them seamlessly to the CPSoS operation.

Viewing the above challenges from an engineering perspective, it can be remarked that modelling and designing a CPSoS can be a very complex task. The CPSoS consists of various, heterogenous components including CPS hardware and software modules, network modules, and communication channel protocols. Current modelling tools have heterogeneity, but they can capture only specific aspects of a CPSoS and cannot represent CPSoS emergent behaviors and all CPSoS functional and non-functional requirements. Thus, a model of models approach must be followed, manifested in a modeling tool integration and enhancement toolbox that will provide all necessary meta-models and modeling formalisms capable of capturing all the CPSoS specificities (beyond a single CPS). This needs to be extended to the simulation level, where the research landscape is fragmented in model-based simulation specialized to individual CPS components (software, hardware, network). Thus, a meta-modeling simulation toolbox to capture abnormal behavior and CPSoS emerging characteristics must also be realized to validate the CPSoS functionality.

Typically, CPSs are designed using a model-based approach which however does not take into account the fact that CPSoS have a continuous evolution: the frequent addition, removal, and modification of hardware and software CPS components over the CPSoS life cycle (often many years) [2]. This poses a considerable CPSoS challenge since the CPSoS design phase and operation phase are not separated but rather coexist through time, thus forming a design operation continuum that must be supported through some innovative engineering methods and tools. This continuum leads to the need for a system-wide dynamic reconfigurability and adaptability of CPS resources and CPS process lifecycles. However, the complexity and autonomy of the CPSoS makes it very hard to identify when a reconfiguration is needed, thus highlighting the need for introducing a CPSoS cognitive mechanism. The cognitive CPSoS must be able to provide situational awareness in a decentralized manner (matching the way CPS operate within the system). This requires research in several different topics related to machine learning, data analysis, and collaborative cognitive mechanisms that must consider human behavior and physical environment updates. Eventually, the CPSoS situational awareness must lead to high resilience, high responsiveness and resistance to faults.

Finally, it cannot be overlooked that security and trust in CPSoS operation must be retained. Security breaches can lead to serious incidents that may affect human lives (in automotive, energy, aerospace, railways, industrial domains etc.). The autonomous nature of the CPSoS, the high heterogeneity and the use of legacy components however, makes traditional security measures hard to apply. To overcome this, security components must be modelled based on the security-by-design-principle considering that they may be placed in CPSs with various and different, security needs and performance capabilities. Such components must be realized during design/redesign of the CPSoS, while in parallel specialized security monitoring mechanisms and tools must be introduced in the autonomous CPSs and the system as a whole, so that they can detect, identify, respond and mitigate to a security attack in the presence of unforeseen conditions.

III. CPSoSaware Objectives

Obj. 1: *Design cognitive, reconfigurable and autonomous CPSoS orchestration, that support CPSoS full lifecycle (requirements, design, test, operate and decommissioning) and design operation continuum.* The target is to create a new holistic model-based design paradigm that is extended during the operational phase of the CPSoS and supports an autonomic, cognitive, self-awareness mechanism for the CPSoS components and system as a whole. The CPSoSaware approach is supported by two concepts, the Model, Optimize, Design, Deploy (MODD) concept and the cognitive and cooperative control concept.

Obj. 2: *Provide a decentralized, cooperative, autonomic control and management that is resilient, fail-safe and adaptable to unforeseen physical and cyber-changes.* Our aim is to design a system wide solution that can perform operations in a decentralized way, relying on the cooperation between CPSs to handle typical collaborating CPS and CPSoS activities thus requiring minimal intervention from system level. Reliability/resilience failures up to a medium level of criticality can be handled through the decentralized, collaboration mechanism (shifting tasks when a CPS defaults etc.) or through the fail-safe design that is followed in the MODD optimization approach.

Obj. 3: *Structure a CPSoS design approach that can be modelled and simulated at system level.* Our aim is to create a new holistic model based design paradigm that is extended during the operational phase of the CPSoS and is flexible enough to support redesign, deployment, commissioning, and decommissioning of CPS components during the CPSoS operation phase.

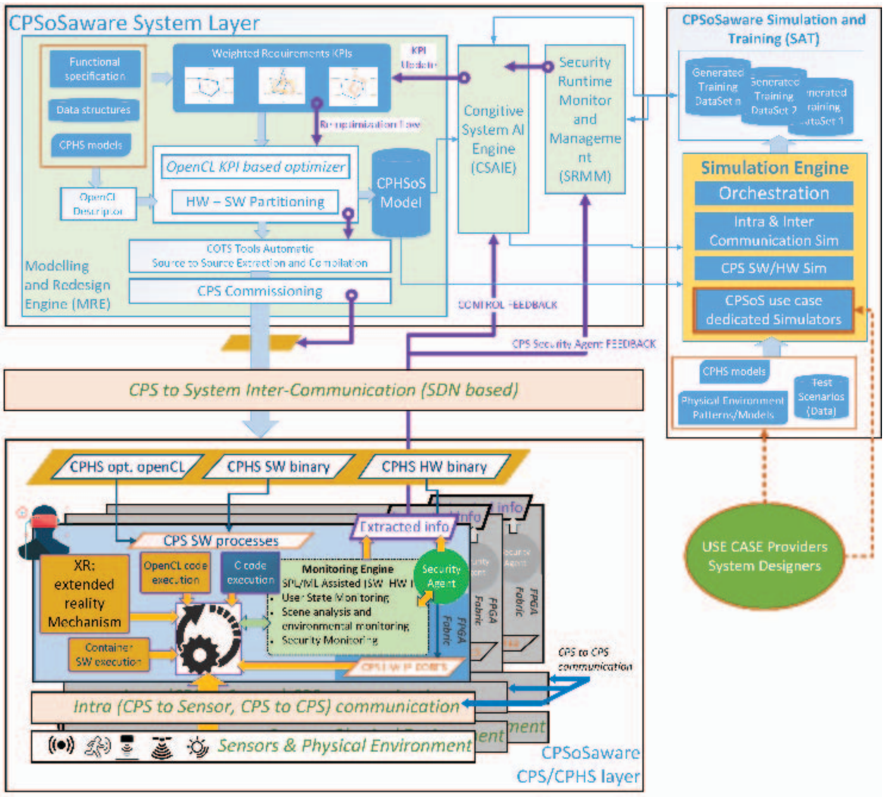
CPSoSaware System Layer
Functional specification
Data structures
CPHS models
Weighted Requirements KPIs
KPI Update
Re-optimization flow
OpenCL KPI based optimizer
HW – SW Partitioning
OpenCl Descriptor
CPHSoS Model
COTS Tools Automatic Source to Source Extraction and Compilation
CPS Commissioning
Modelling and Redesign Engine (MRE)
Congitive System AI Engine (CSAIE)
Security Runtime Monitor and Manageme nt (SRMM)
CONTROL FEEDBACK
CPS Security Agent FEEDBACK
CPSoSaware Simulation and Training (SAT)
Generated Training DataSet n
Generated Training DataSet 2
Generated Training DataSet 1
Simulation Engine
Orchestration
Intra & Inter Communication Sim
CPS SW/HW Sim
CPSoS use case dedicated Simulators
CPHS models
Physical Environment Patterns/Models
Test Scenarios (Data)
USE CASE Providers System Designers
CPS to System Inter-Communication (SDN based)
CPHS opt. openCL
CPHS SW binary
CPHS HW binary
Extracted info
CPS SW processes
XR: extended reality Mechanism
OpenCL code execution
C code execution
Container SW execution
Monitoring Engine
SPL/ML Assisted (SW HW
User State Monitoring
Scene analysis and environmental monitoring
Security Monitoring
Security Agent
FPGA Fabric
CPS to CPS communication
Intra (CPS to Sensor, CPS to CPS) communication
Sensors & Physical Environment
CPSoSaware CPS/CPHS layer

Figure 1: The CPSoSaware conceptual architecture.

Obj. 4: *Provide vertically and horizontally secure and trusted designs (Security-by-Design) and runtime cybersecurity monitoring to protect against cyber-threats and respond to attacks.* In this objective we deal with security and reliability that can be supported through the introduction of security and trust components during design time in CPSs and a security monitoring CPSoS level entity that can provide responses to cybersecurity threats using specialized mechanisms.

Obj. 5: *Consider throughout the CPSoS lifecycle human users and operators and provide Extended Reality solutions that increase their situational awareness (human in the loop).* CPSoSaware will research and develop a suite of tactile and visual experience optimizations for mobile and standalone XR Cyber Physical Human System's (CPHS) interfaces.

Obj. 6: *Integrate the CPSoSaware various tools into a unified solution and test it in two distinct use cases: connected autonomous cars and manufacturing processes with robotics and human interaction.* The CPSoSaware solution will be evaluated/tested in small scale trials in two pilot sites (Germany and Italy) where we will execute specific use case scenarios in order to evaluate the CPSoSaware methods and tools in different requirement settings.

Obj. 7: *To define evidence-based business and financing models along with a business plan for the post-project sustainable exploitation of the CPSoSaware framework.* The project will demonstrate the financial benefits (including cost-effectiveness and efficiency gains) that its novel framework can introduce to connected semi-autonomous cars with quantifiable evidence.

IV. The CPSoSaware Approach

CPSoSaware solution is formulated as a framework that spans horizontally and vertically on the overall CPSoS architecture thus providing a holistic, cognitive and decentralized way of designing, running and decommissioning CPS components of the system or even the CPSoS as a whole. Since such an endeavour is highly complex and hard to manage, in CPSoSaware we use AI and Machine Learning (ML) assistance in order to make the above-mentioned procedures feasible and pragmatic. In the CPSoSaware, we also consider the human users and operators of CPSoS as an integral part of the system since CPSs in many complex systems are meant to assist/collaborate with its human users thus forming, as noted, cyber physical human systems (CPHS).

The CPSoSaware system is divided into three different architectural blocks that are interconnected to handle the CPSoS heterogeneity and complexity. These blocks are i) the **CPSoS system layer block**, ii) the **simulation and training block**, and iii) the **CPS/CPHS layer block**. The conceptual CPSoSaware architecture concept is depicted in Figure 1.

1) System Layer Block

The System Layer of the CPSoSaware architecture is responsible for handling the SoS functionality of a CPSoS and providing system level orchestration, control, and monitoring of the various CPS/CPHS. Its main role is to capture, evaluate/re-evaluate the high level CPSoS functional and non-functional requirements and provide all necessary optimizations in order to reconfigure and redesign the System's CPSs/CPHSs so as to holistically match systemic design and operational goals/parameters achieving reliability, robustness, responsiveness, CPS/CPHS criticality and security/trust.

At system level, a CPSoS designer can determine the required design parameters that act as Key Performance Indicators (KPIs) for the overall CPSoS lifecycle and apply appropriate weights to them in order to determine how the CPSoS components should be optimized in order to always match these weighted KPIs. This includes CPS/CPHS software and hardware structures, networking and communication, control loop processes, individual CPS behaviors as well as the emergent CPSoS behaviour and properties.

The above system level analysis and optimization process requires a CPSoS description that will be modelled using a design toolboxes that relies on appropriate user, physical environment models, datasets, patterns, and principles provided by the use case partners. The modelling approach is a consolidation of various modelling tools that need to be collectively used in order to create a model of models that can capture the CPSoS overall functionality.

The extracted detailed CPS descriptions, which realize the autonomous CPS functionality and constitute part of the CPSoS emerging functionality, will be translated in an *"executable specification."* It will then be fed to an automated code optimization and execution extraction toolbox to create the actual software and hardware runnables (software and hardware components and the connecting "glue" that will be actually deployed at the CPS nodes).

The extracted, optimized "executable specification" focuses on matching functional and non-functional requirements of each CPS/CPHS and their communication. In addition, it will also provide the necessary system level tasks per CPS/CPHS that will enable the appropriate decentralized strategy to be followed in order to minimize the traditional, complex and centralized control and orchestration of the overall CPSoS. This strategy will implement the collaborative, cognitive, CPS/CPHS behaviours during CPSoS operation and includes resource, computation and task re-distribution per CPS and across CPSs (including cross CPS collaborative, distributed multimodal fusioning, computation of measurements, and ML results extraction).

Finally, the decentralization strategy also includes those data structures, ML processes and behaviours that need to be deployed in the CPSs/CPHSs that will act as a physical environment, failure indicator monitoring and reporting mechanism to the system layer. Such feedback from the CPS/CPHSs (i.e., the CPS layer) is evaluated by the system layer as a trigger for CPSoS reconfiguration-redesign thus providing self-awareness.

Hence, the system layer plays a crucial role during the operation phase of the CPSoS by having the responsibility of constantly monitoring the CPSoS in order to assess (with the use of cognitive artificial intelligence and assisted/reinforced learning) the status of the CPSoS in terms of KPI achievement. The system layer cognitive AI mechanism can detect with the help of existing training datasets and generation of new ones (through the CPSoSaware simulator) that cover possible corner cases, changes in the system's various CPS/CPHS environment that may lead to non-compliance with the KPI values or KPI updates, thus triggering a redesign process (i.e., recreation of the "executable specification"). A similar procedure is also followed when new CPS components need to be added or removed from the CPSoS (commissioning/decommissioning). To sum up, this process provides autonomic support of the CPSoS design operation continuum.

Security is considered during the overall CPSoS lifecycle as an integral part of the CPSoS. Therefore, the KPIs definition, CPS processes modelling and optimization are extended appropriately to include security related processes and resources by deploying on the CPSs the security-by-design directive. The security processes and resources follow the decentralization principles that are used in CPSoSaware to create CPS autonomy yet accomplish the additional role of providing feedback to the runtime security monitoring (anomaly detection) component.

Based on the above overall functional description and as depicted in Figure 1, the system layer block consists of three components, a) the CPSoSaware Modelling and Redesign component (MRE), b) the Cognitive System AI Engine (CSAIE), and c) the Security Runtime Monitoring and Management (SRMM) component.

2) Simulation and Training Block (SAT)

The CSPoSaware SAT block constitutes the basic testing and training data extraction environment for the (re)design procedures performed in the MRE system layer component. It provides the integration and orchestration framework for various, different simulation subcomponents that simulate specific areas of the CPSoS architecture like the CPS/CPHS hardware and software processes, the CPS/CPHS to CPS/CPHS and CPS/CPHS to system network environment and the CPS to human interaction. The SAT orchestrator will be able to extract simulation data from all the subcomponents and will provide a data control and collection environment for CPSoS specific simulators (e.g., connected semi-autonomous car simulator and human-robot collaboration in manufacturing environment simulator).

By specifying various test case scenarios that involve input measurement patterns from the physical environment and human behaviors, the simulation block can generate test output data that can primarily be used for the learning/training process of the AI that is utilized in the CSAIE block and more specifically at the CSAIE design and later during CSAIE operation to identify corner cases. Thus, the simulation block acts as an extension/enhancement of CSAIE by generating, through simulation, training datasets that closely depict systemwide abnormalities of the CPSoS which may have considerable impact on the CPSoS requirement KPIs. This approach, over time, will improve the CSAIE learning mechanism and heighten the CSAIE self-awareness level and accurate responsiveness to systemwide failures. However, to effectively train artificial intelligence logic of complex systems through typical training datasets may require a very large amount of data and considerable time. In CPSoSaware we will also work on optimization techniques for labeling and classifying large data sets to extract training features (metadata) that can speed up the training process.

3) CPS Layer Block

The CPS/CPHS Layer Block of the CPSoSaware architecture is focused on assisting the design operation continuum support mechanism (MODD, MRE, CSAIE etc.) at each CPS/CPHS in terms of CPS redesign, operation and commissioning/decommissioning of hardware, software and communication modules. In CPSoSaware, we consider that each CPS node may be based on a legacy device, a low, mid or high-end embedded system or an emerging computing device structured on a System-on-Chip (SoC) that supports multi-core CPUs, GPUs, and/or FPGA logic on the same chip (e.g., Xilinx Zynq series, Intel Stratix/Cyclone/Aria series, Microsemi SmartFusion 2 series etc.).

The SoC based approach is supported by many embedded system key players [5] that on the upcoming years may bring their technologies on CPS devices, in CPSoSaware. Therefore, we acknowledge the high level of heterogeneity that such emerging technologies will bring in the modelling and design of CPSs and CPSoSs. The foreseen heterogeneity is addressed by including an OpenCL-based software stack for the CPS processes that can provide optimal distribution of tasks/processes to the various heterogenous CPS architectural components. Thus, the CPSoSaware System Layer Block code generation outputs (i.e., optimized OpenCL kernels, CPS host software executable, FPGA hardware bitstream), in the CPS Layer and we will include several technologies to support the deployment of these outputs to the CPS devices. In brief, these technologies are: i) **Dynamic Partial FPGA Reconfiguration (DPR), ii) Software Container based Reconfiguration, and iii) the heterogeneous parallel OpenCL API [11] for application structuring and kernel distribution over diverse compute platforms.**

The OpenCL based software stack is based on POCL (POrtable Computing Language [9]) and its *distributed* driver [10] which is being developed in another ongoing EU project FitOptiVis [13]. Higher-level programming layers that support domain specific APIs such as OpenVX [12] are also planned for added engineering productivity, as well as integrating sensors and hardware accelerators seamlessly to a common CPSoSAware OpenCL platform.

Moreover, there are three additional functionalities implemented as specialized, customized, and optimized components and realized at the CPS Layer Block. These are:

i) Distributed, Cognitive and Cooperative Intelligence. A very important feature of the CPSoSaware solution is the decentralized, cognitive, and cooperative multitasking mechanism shared between CPSs without the involvement of the System Layer block. The various CPHS, even if they act independently, they will be able to realign their processes in order to collectively provide fault-tolerance, resilience and reliability in the presence of unforeseen critical events (e.g., abnormalities from the physical or cyber world).

More importantly, critical and useful nodes with respect to a system wide objective (e.g., scene analysis and identification of free space in cars or mobile robots) will be selected in a distributed manner while optimizing the system/network wide objective (e.g., improving safety in connected cars, or safe operation of collaborative mobile robots). To this end, a proper utility measure will be computed in a distributed fashion. This will include the usefulness and quality of the signals that a CPS delivers and at the same time the position of mobile CPS in the network (e.g., distance from the obstacle). CPSs will give scores to each of their neighbours and from these local utility scores a network wide rating system will be computed using approaches that are based on the spectral graph theory concept of eigenvector centrality. Additionally, the CPSs will be capable of executing robust distributed signal processing and learning algorithms including feature extraction, detection and labelling (classification) techniques. For example, if a source is detected in multiple CPSs, all CPSs must label this detection as coming from a single entity, which requires distributed classification techniques. Such techniques are close to optimal under nominal conditions and highly reliable, ensuring a) robustness with respect to uncertainties attributed to sensing and communication failures and/or possible CPS malfunctioning, physical and cyber-attacks, b) adaptive, to cope with environment non-stationarities, and c) transmission power efficient.

ii) Human in the Loop Situational Awareness using Extended Reality tools. CPHs will be equipped with a suite of tactile and visual experience optimization interfaces for mobile and standalone XR systems targeting the evolving needs and specificities of users, in terms of: i) situational awareness, ii) reaction time, iii) quality of experience, and iv) user engagement. By tracking physical responses to different training situations and dynamically adapting the visual, auditory and tactile information rendered in the real world, the system will ensure that its amount will not exceed the user's ability to handle it. More importantly, those XR interfaces are expected to improve situational awareness, reduce reaction time in emergency situations, by utilizing touchable spatial interfaces, ensuring safe, reliable and timely response to systems' request to intervene messages.

iii) Intra-CPS Communication Layer. Considering highly distributed tasks and cooperating nodes/sensors, the impact of intra-CPS communication layer can be quite significant on the overall CPSoSaware platform. The role of this layer is effectively twofold. The first role concerns the reliable, efficient and energy aware communication between a specific CPSoSaware node with its assigned sensors. The second one, is related to the communication between different CPS nodes to collectively execute a distributed task. In order to tackle both aspects CPSoSaware node architecture will explore the support of prominent communication interface(s) from the domain of IoT such as WiFi, BLE, ZigBee, UWB, or LoraWAN and V2X communications. Concerning the first aspect and assuming multiple and diverse sensors are connected to a specific CPSoSaware node, optimum selection of communication technology and its configuration will be driven by application requirements, traffic conditions, link quality indicators etc. possibly activating specific scheduling or redundant transmission algorithms. Focusing on the second aspect, where different and possibly heterogenous CPSoSaware nodes need to collaborate, a second level of communication scheduling will be applied to assure reliability and time constrained performance.

V. CPSoSaware Trials & Use Cases

The CPSoSaware architecture is going to be tested in two different pilot sites (Germany and Italy) by performing trial scenarios for two different use cases. The first use case is focused on **connected semi-autonomous vehicles** where trials focusing on human in the loop scenarios will be performed, such as non-predictable failures that may involve the human driver and how this affects the design operation continuum support of the CPSoSaware solution as well as human situational awareness enhancements when using the CPSoSaware architecture. We will also use this use-case to assess the cybersecurity mitigation strategies using the CPSoSaware architecture and its response to cyberattacks.

The second use case focuses on **human-robot collaboration in the manufacturing environment** and will involve trials that challenge i) the MODD CPSoSaware concept and ii) the collaborative control mechanism through accidents/failures and cybersecurity attacks. The use case will also assess the autonomic decentralized operation of the CPSoSaware solution as well as the design operation continuum support in the presence of cybersecurity attacks.

The two use cases complement each other since they have different requirements and specificities (open spaces and moving CPS, close interconnection with humans versus closed space environment, static CPS, and more relaxed interaction with humans).

VI. Project Breakdown

CPSoSaware is a 36-month project. The overall work plan consists of eight work packages (WPs), each divided into manageable sections of coherent tasks. Figure 2 illustrates the dependencies between the formulated WPs.

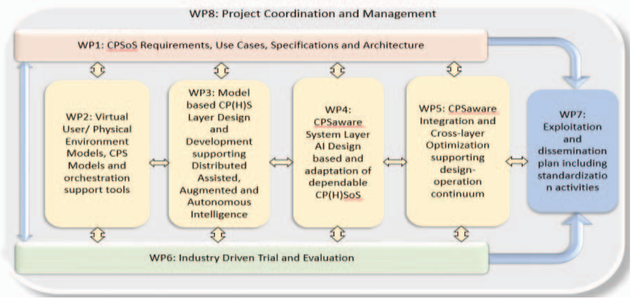
WP8: Project Coordination and Management
WP1: CPSoS Requirements, Use Cases, Specifications and Architecture
WP2: Virtual User/ Physical Environment Models, CPS Models and orchestration support tools
WP3: Model based CP(H)S Layer Design and Development supporting Distributed Assisted, Augmented and Autonomous Intelligence
WP4: CPSaware System Layer AI Design based and adaptation of dependable CP(H)SoS
WP5: CPSaware Integration and Cross-layer Optimization supporting design-operation continuum
WP7: Exploitation and dissemination plan including standardizatio n activities
WP6: Industry Driven Trial and Evaluation

Figure 2: Work packages descriptions and dependencies

VII. Impact

The automotive domain provides jobs for 12M people and accounts for 4% of the EU GDP including sales and maintenance for 4.3M and transport for 4.8M. The automotive industry has been thoroughly investing in CPSs inside cars either to provide control for traditional automotive processes (like brake system) or to introduce new concepts like (semi-) autonomous driving. This domain has been making a transition to the system-of-systems approach aiming to offer a series of emergent functionalities like traffic and collaborative car fleet management or large-scale automotive adaptation to physical environment, thus providing significant environmental (e.g., air pollution reduction) and societal impact.

Similarly, large infrastructure domains, like industrial manufacturing with more than 30M employees, a turnover of 6B, and an added value of 1.6B in 2010 are evolving into global, highly integrated CPSoS that go beyond pure production. CPSoSs cover all parts of the value chain, including research, design, and service provision. This novel approach can enable a high level of flexibility that can be interpreted to fast adaptation to customer requirements, high degree of product customization and better industrial sustainability [6, 7, 8]. The wide adoption and use of CPSoS in the above-mentioned domains, will bring significant advantages in EU economy and help Europe capitalize its CPS excellence. This is easily justified considering that the embedded systems industry alone creates 50K new jobs every year, and that Europe accounts for 30% of the world production of embedded systems, with particular strengths in the automotive sector, aerospace, and health.

VIII. Conclusions and Further Work

Today's and future dynamic Cyber Physical Systems of Systems (CPSoS) require a radical change to the design-operation continuum. Towards this direction, the CPSoSaware project sets forward a novel, holistic approach on supporting the CPSoS design operation that is based on a cognitive, self-aware mechanism to assist modeling and design. The project is formulated around seven well-defined and measurable objectives that correspond to specific research challenges in various technological fields, like CPS model-based design, cognitive artificial intelligence based CPS control, extended reality human-based situational awareness, and security and cyberattack countermeasures. To address these challenges, a multilevel, yet clear, architectural approach is going to be established. This approach spans over the whole CPSoS lifecycle stages; from design, to development, commissioning, maintenance, and decommissioning. Finally, the impact and the potential added value of the project outcomes, are highlighted by the involvement of key players from the automotive and manufacturing domains offering challenging real-life use cases to test and validate the CPSoSaware toolset.

Acknowledgment

This work has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 871738 - CPSoSaware: Cross-layer cognitive optimization tools & methods for the lifecycle support of dependable CPSoS.

References

[1] CPSoSaware web site https:// http://cpsosaware.eu/

[2] S. Engell, R. Paulen, C. Sonntag, H. Thompson, M. Reniers, S. Klessova, and B. Copigneaux. Proposal of a European Research and Innovation Agenda on Cyber-physical Systems of Systems 2016-2025, 2016.

[3] https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code

[4] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang. Big data meet cyber-physical systems: A panoramic survey. IEEE Access, 2018.

[5] https://www.eetimes.com/author.asp?section_id=36&doc_id=133093 6#

[6] M.-M. Meinecke, M. Obojski, D. M. Gavrila, E. Marc, R. Morris, M. Töns, and L. Lettelier. Strategies in terms of vulnerable road user protection. EU Project SAVE-U, 2003.

[7] S. Sedighi, D. Nguyen, and K.D. Kuhnert. Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications. Intl. Conference on Control, Automation and Robotics, 2019.

[8] S. Engell, R. Paulen, M. Reniers, C. Sonntag, and H. Thompson. Core Research and Innovation Areas in Cyber-Physical Systems of Systems, 2015.

[9] P. Jääskeläinen, C.S. de La Lama, E. Schnetter, K. Raiskila, J. Takala, and H. Berg: "pocl: A Performance-Portable OpenCL Implementation" Journal of Parallel Programming, 2015.

[10] J. Solanti, M. Babej, J. Ikkala, and P. Jääskeläinen. POCL-R: Distributed OpenCL runtime for low latency remote offloading. Intl. Workshop on OpenCL, SYCL, Vulkan and SPIR-V, 2020.

[11] The OpenCL specification. Khronos® Group. https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_API.html

[12] The OpenVX specification. OpenVX Working Group version 1.3, Thu, 08 Aug 2019. Available at https://www.khronos.org/registry/OpenVX/specs/1.3/html/OpenVX_Specification_1_3.html

[13] Z. Al-Ars, T. Basten, A. de Beer, M. Geilen, D. Goswami, P. Jääskeläinen, J. Kadlec, M.M. deAlejandro, F. Palumbo, and G. Peeren. The FitOptiVis ECSEL project: Highly efficient distributed em-bedded image/video processing in cyber-physical systems. Intl. Conference on Computing Frontiers, 2019.