Luigi Pomante, Francesca Palumbo, Claudia Rinaldi, Giacomo Valente, Carlo Sau, Tiziana Fanni, Frank van der Linden, Twan Basten, Marc Geilen, Geran Peeren, Jiří Kadlec, Pekka Jääskeläinen, Marcos Martinez, Jukka Saarinen, Tero Säntti, Maria Katiuscia Zedda, Victor Sanchez, Dip Goswami, Zaid Al-Ars, Ad de Beer

"Design and management of image processing pipelines within CPS:

2 years of experience from the FitOptiVis ECSEL Project"

Euromicro Conference on Digital System Design (DSD) 2020

The publisher's version is available at:
https://ieeexplore.ieee.org/document/9217819

When citing, please refer to the published version, using the following format:
L. Pomante et al., "Design and management of image processing pipelines within CPS: 2 years of experience from the FitOptiVis ECSEL Project," 2020 23rd Euromicro Conference on Digital System Design (DSD), Kranj, Slovenia, 2020, pp. 378-385, doi: 10.1109/DSD51259.2020.00067.

Design and management of image processing pipelines within CPS: 2 years of experience from the FitOptiVis ECSEL Project

Luigi Pomante[1], Francesca Palumbo[2], Claudia Rinaldi[2], Giacomo Valente[2], Carlo Sau[3], Tiziana Fanni[2], Frank van der Linden[4], Twan Basten[5], Marc Geilen[5], Geran Peeren[4], Jiří Kadlec[6], Pekka Jääskeläinen[7], Marcos Martinez[8], Jukka Saarinen[9], Tero Säntti[10], Maria Katiuscia Zedda[11], Victor Sanchez[5], Dip Goswami[5], Zaid Al-Ars[12], Ad de Beer[4]

[1]Università degli Studi dell'Aquila, [2]Università degli Studi di Sassari, [3]Università degli Studi di Cagliari, [4]Philips, [5]Eindhoven University of Technology, [6]Institute of Information Theory and Automation, [7]Tampere University, [8]Thales Alenia Space España, [9]Nokia, [10]University of Turku, [11]Abinsula, [12]Delft University of Technology

***Abstract*— Cyber-Physical Systems (CPS) are dynamic and reactive systems interacting with processes, environment and, sometimes, humans. They are often distributed with sensors and actuators, smart, adaptive, predictive and react in real-time. Indeed, as sight for human beings, image- and video-processing pipelines are a prime source for environmental information for systems allowing them to take better decisions according to what they see. Therefore, in FitOptiVis we are developing novel methods and tools to integrate complex image and video processing pipelines. FitOptiVis aims to deliver a reference architecture for describing and optimizing quality and resource management for imaging and video pipelines in CPS both at design- and run-time. The architecture is concretized in low-power, high-performance, smart components, and in methods and tools for combined design-time and run-time multi-objective optimization and adaptation within system and environment constraints.**

Keywords*— *Image-video processing, distributed systems, heterogeneous system, energy and performance optimization, Cyber-Physical Systems

I. Introduction

Smart systems integration is one of the essential capabilities required to improve the competitiveness of European industry in the ECSEL application domains. This is especially relevant for Cyber-Physical Systems (CPS), i.e., autonomous distributed integrations of electronic systems and software tightly coupled to and interacting with physical systems and their environment. The ECSEL project FitOptiVis (*From the cloud to the edge - smart IntegraTion and OPtimisation Technologies for highly efficient Image and VIdeo processing Systems*) [1] deals with novel design and run-time approaches for imaging and video pipelines in CPS. Images play a central role in human perception and understanding of our environment. Accordingly, CPS need visual context and awareness to make correct decisions and take appropriate actions. Fig. 1 shows a generic distributed image- and video-processing pipeline for CPS. Advanced image and video processing is computing intensive, whereas for adequate behavior results need to be available with low latencies and high throughput. Devices often need to operate with low energy, limited heat dissipation, and optimization for other qualities may be important.

FitOptiVis started on June 1st, 2018 and its duration is 3 years. In the following, we highlight project objectives, explain the selected approach, describe target use cases (UC), and present the main results obtained after 2 year of activities

II. FitOptiVis Objectives

CPS are systems that are in feedback with their environment, possibly with humans in the loop. They are often distributed with sensors and actuators, smart, adaptive and predictive and react in real-time [24].

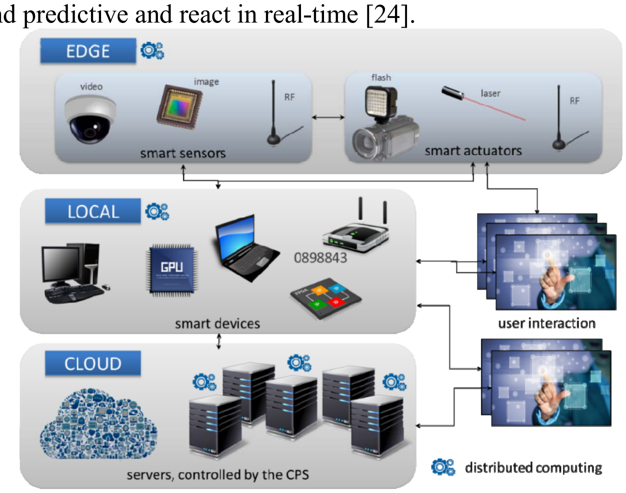

Fig. 1. Generic configuration of imaging and video pipelines in CPS.

Image- and video-processing pipelines are a prime source for environmental information improving the possibilities of active, relevant feedback. They often need to satisfy stringent non-functional constraints, so advanced imaging and video applications become very complex. The main objective of FitOptiVis is to develop an approach for smart integration of image- and video-processing pipelines for distributed CPS applying a combined design- and run-time multi-objective optimization of quality and resource usage within system and environment constraints. This is supported by a reference architecture, low-power high-performance smart devices and proper design-time and run-time methodologies and tools. From the implementation point of view, distributed pipelines (see Fig. 1) consist of a heterogeneous configuration of legacy devices, state-of-the-art multi-vendor devices and components, and newly developed application-specific ones. Smart system integration for imaging and video applications must be built evolutionarily upon earlier developments, and the CPS must be able to cope with individual component (HW and SW) upgrades during its lifetime.

The FitOptiVis objective is being reached by pushing the state of the art in development and run-time support of distributed imaging and video pipelines, targeting primarily real-time performance and energy usage. FitOptiVis is working on providing a reference architecture that enables the integration of state-of-the-art technology and new developments in the mentioned domain, supporting composability built on suitable abstractions of components, embedded sensing, actuation and processing devices. The reference architecture supports design portability, on-line multi-objective quality and resource optimization and run-time adaptation, guaranteeing system constraints and requirements based on platform virtualization. A cloud in the FitOptiVis context is a set of connected servers under the control of the CPS. Non-functional aspects other than performance and energy, such as reliability and security, are taken into account to meet UC specific objectives but are not an explicit target of research and development.

Design- and run-time models provide a suitable set of component abstractions for performance and energy related to the distributed system configurations and the use of processing, communication and storage resources. A Domain Specific Language (DSL), QRML, for Quality and Resource Management Language [21], is defined with a well-defined mathematical semantics [22] that allows to specify the modular structure of a CPS with the essential qualities of interest (performance, energy, visual quality, and so on) and the required resource budgets (processing, memory, bandwidth) of video and imaging tasks, devices and components. Specific design-time methods and tools address performance and energy optimization and seamless, compositional integration of the image and video pipelines. The reference architecture provides templates for a flexible virtual platform built on the component abstractions. The imaging and video pipelines can be designed targeting virtual platforms in various configurations corresponding to different points in the multi-objective performance, energy usage, and resource cost space. These virtual platform configurations can be mapped at run-time onto physical resources, depending on their availability and the needs of other applications. Energy-efficient, high performance, smart devices and components are developed to support and demonstrate the reference architecture. The developed tooling ensures effective resource usage predictions and simulations for design-space exploration at design-time, and multi-objective optimizations during run-time.

FitOptiVis will integrate ultra-low power and high-performance devices and components into the reference architecture. It exploits the advantages of distributed resources, where off-the-shelf processing elements are supported by efficient companion computing elements near the sensors or actuators (edge computing) through, for example, configurable hardware accelerators. By applying novel ultra-low power technology, complex, multi-source functionality, which typically needs to run on multiple heterogeneous components, FitOptiVis-compliant systems are able to meet CPS needs. In particular, to support the adaptation of the complete imaging or video pipeline to ensure dynamic behavior and to guarantee optimal behavior in terms of power and performance, even in the case the execution-condition changes due to internal and external triggers. Finally, FitOptiVis will lead to shorter development times and improved products with richer functionality.

III. FitOptiVis Concept and Approach

FitOptiVis developments are driven by industrial UCs that serve as the basis for requirements, demonstration, and validation. Models and abstractions play a crucial role for integrated reference architecture, design methods and run-time operation of CPS imaging and video pipelines. All UCs will use QRML [22] to describe the resource options and constraints. Devices and components expose their functional and performance set points, with the corresponding resource requirements, through minimal interfaces. Model-driven design methods aim to design set points that trade off functional and performance capabilities against resource usage, for maximal flexibility [3][4].

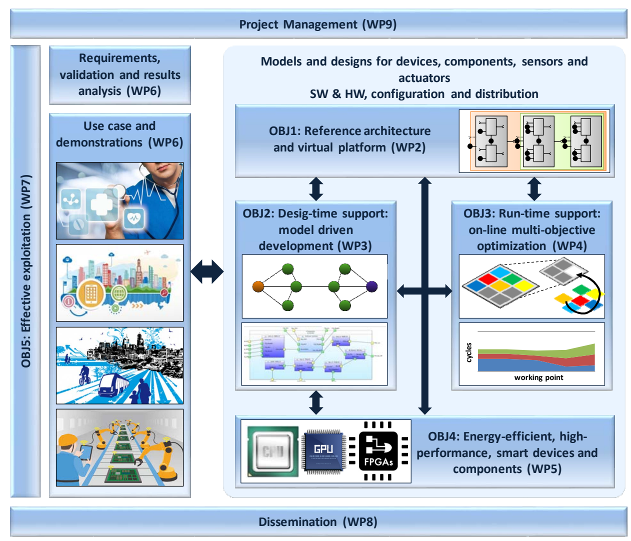

Fig. 2. Organizational approach and dependencies.

Resource virtualization and predictable, composable reconfiguration enable modular, scalable run-time multi-objective adaptation and optimization ensuring quality and real-time performance, even across reconfigurations. On-line monitoring techniques enable the evolution of set points and their resource requirements in changing contexts, both operational environments and use patterns. The common reference architecture captures the essential aspects of the envisioned approach in appropriate component abstractions, virtualization techniques and quality- and resource-management protocols. The approach is grounded in the development of smart, high-performance, energy-efficient devices and components to validate concepts and exploit results of FitOptiVis. Fig. 2 shows organizational approach of FitOptiVis, while Fig. 3 sketches technical approach.

The FitOptiVis operational objectives are motivated by industrial UCs that share demanding image and video processing, while differing in other aspects like available processing power, energy budgets, safety requirements, configuration distribution, configuration dynamics, etc. The UCs dictate the requirements for integral development approach and smart integration of image- and video-processing pipelines for CPS. UCs form also the basis for demonstrators. The demonstrators serve for validation and analysis of the project results, and for their exploitation. The UCs are detailed in the next section. Due to the industry driven approach, technical specifications of FitOptiVis platform, components and tools come from real needs. Both cross-domain and domain-specific requirements are considered to cover all the necessary design set points.

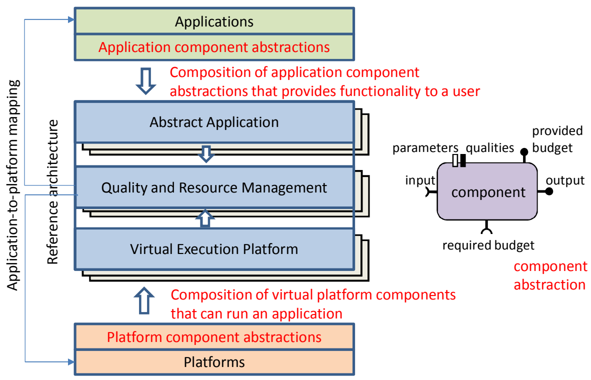

Fig. 3. Technical approach and interfaces – the blue parts form the reference architecture.

UCs and demonstrators have been chosen to have diversity in image and video processing tasks. The different application domains are expected to benefit from the common reference architecture, design methods and run-time approach, and, generally, from re-usable and easily customizable devices and components. Generic and unified platform and design methods enable the quick creation of image and video processing infrastructure in different domains, and simplified connections between systems across domains. This will become increasingly important in our rapidly evolving society where systems are without exception connected in the Internet of Things. At the end of FitOptiVis we intend to deliver generic results applicable in many different domains, even outside the scope of the project itself, and across domains.

IV. FitOptiVis Use Cases

FitOptiVis takes its main requirements, architecture, designs, implementations, demonstration and validation from the UCs listed below:

- *UC1-Water Supply*: the water supply system constitutes a critical distributed infrastructure that requires continuous monitoring, to detect the facility integrity, unauthorized accesses and fast surveillance and maintenance intervention, to restore missing service states.
- *UC2-Virtual Reality*: virtual reality systems are increasingly important in medical, gaming and military applications and cinema industry. Recent advances in digital photography and video led to the development of advanced 3D vision and display systems. Virtual reality applications should be supported by high-quality video capture, efficient coding and processing technologies and an accurate, fast positioning system.
- *UC3-Habit Tracking*: Habit Tracking at home is particularly relevant for the elderly population, to assess their physical habits and identify situations where a healthy lifestyle can be supported and to detect deviations from a standard behavioral pattern, or early discovering of potential deterioration of users' health or wellness.
- *UC4-Industrial Inspection*: Industrial inspection applies machine vision to quality control in production processes. Cameras assess the produced objects and automatically determine their quality.
- *UC5-Road Traffic Surveillance*: road and railway traffic surveillance, for vehicle detection and recognition, aims at supporting road management, reducing congestions.
- *UC6-Multi Source Streaming Composition*: definition of an embedded high-performance video compositor capable of rendering 8 simultaneous streams on a screen. Tight coordination among the video sources, and with the compositor is required, as well as adaptivity to variations in setting. If the screen layout changes, the compositor needs to dynamically adapt.
- *UC7-Sustainable Safe MRI*: to facilitate MRI use in long interventional procedures, in pediatric imaging, and in the emergency setting, a low energy deposition method for 4D imaging is intended to be developed. Goals are to reduce energy consumption and provide thermal losses models to predict control and minimize thermal load, while maintaining superior image quality.
- *UC8-Robots Calibration*: robots, robotic arms or manipulators require calibration to be done periodically or after each geometry change. This UC is meant to provide fast and automated calibration, getting rid of any mechanically connected measurement device.
- *UC9-Surveillance of Smart-Grid Critical Infrastructure*: active surveillance for the prevention of potential harm to Smart-Grid Critical Infrastructures. The detrimental effects caused by disruptions on the society and citizens must be minimized. This requires a hybrid network that combines information from video surveillance with critical control information on a single.
- *UC10-Autonomous Exploration*: definition of the next generation video processors that will constitute the core of the next earth observation and robotic planetary exploration missions. The challenge is to make the processing adaptable to different non-functional critical parameters (i.e. available power or connection bandwidth) and unexpected functionalities under degraded conditions, or failures of some of the system elements due to the challenging environment.

V. FitOptiVis Implementation

The mission of FitOptiVis implementation (Fig. 4) is: (i) to achieve technical objectives of the proposal, (ii) to maximize collaboration among partners and (iii) to guarantee a solid impact of the project. We aim to have impact in terms of advancements with respect to the state of the art, as well as improvements with respect to current industrial best practices and we intend to foster a concrete market uptake of the project outcomes.

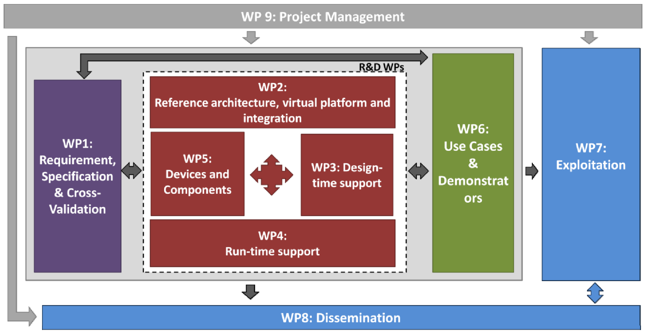

Fig. 4. FitOptiVis Work Plan.

FitOptiVis technical activities are organized as follow:

- *Requirements, specification and cross-validation of the results* – We apply two sources for requirements: the UCs and the industrial/academic knowledge on the state-of-the-art and current issues. Validation, both partial and final, take results from demonstrators on the UCs and the evaluation against the requirements, its outcome is a report with the KPI metrics.
- *Reference architecture, virtual platform and integration* – The reference architecture provides component abstractions for the image and video pipelines and their implementation platforms, as well as platform virtualization techniques, including the QRML DSL. They form the basis for quality- and resource-management for image and video pipelines in budgets for resources like frame rates, power dissipation, required processing, communication and storage. Emphasis is on technology supporting multi-objective optimization for, at least, performance and energy. System-level concerns like distribution and (re-)configuration are addressed. The architecture needs to make sure that the image and video pipelines and the run-time support can work on a heterogeneous network of hardware devices. Each UC is in the process of being described applying the reference architecture.
- *Design-time support* – This work package defines a model-based working methodology involving methods and tools for predicting, simulating and estimating at design-time resource usage. In addition, methods, software libraries, reference designs, HW/SW co-design [2] and compilation techniques are intended to improve the resource behavior of the final system, considering the heterogeneous and changing structure and resource needs of the final system. The development concentrates on the development of all video/image processing building blocks and the run-time support, including hardware IP/accelerators, software applications and sensors.
- *Run-time support* – This work package gathers all the technologies that are intended to implement real time resource management within the system. It is delivering components considering both the actual implementation that will run on the final product and the models to integrate in the system model. It involves monitoring, measuring components [3], and control components implementing the algorithms.
- *Devices and components* – This work packages develops and selects hardware and software devices, components and configurations that are best suitable for optimal energy and performance use. For each component in the FitOptiVis component library a compliant model view will be available at the end of the project, which may present different levels of abstraction, depending on its usage. Components may have different configurations, and being able to support different trade-offs, which could be exploited by WP4 technologies for run-time dynamic support and reconfiguration.
- *Use cases and demonstrators* – The use-case teams are committed in the development of demonstrators, adopting the FitOptiVis reference architecture, along with all the design- and run-time strategies coupled to it. The demonstrators are validated according to the originally defined use-case requirements, considering both design- and/or run-time resource behavior. Partial demonstrators will be available in fall 2020, their assessment is meant to allow last year technology tuning and improvements.

VI. Two Years of FitOptiVis

The following provides a summary of the main technical activities carried on in FitOptiVis in the first 24 months.

A. *WP2 - Reference architecture, virtual platform and integration*

The objective of WP2 is to develop a conceptual reference architecture for the heterogeneous and distributed image pipelines considered in FitOptiVis. It provides template solutions from which concrete solutions for specific platforms, specific components or specific UCs can be derived, or with which they can be compared, and from which common problems and solutions can be identified. The development of the reference architecture is ongoing. It ties together the following elements:

- a conceptual architecture;
- a component interface model that defines the component exterior properties that are relevant for quality and resource management, and their compositions;
- a mathematical semantics that defines component interfaces, compositions, constraints and optimization criteria in terms of a mathematical constraint formulation;
- a DSL (QRML) to specify component models;
- a visualization framework that connects the system model to the FIVIS monitoring framework;
- virtualization techniques to effectively dynamically manage shared resources through virtual platforms;
- a quality and resource management reference framework.

This reference architecture is meant to be evaluated through the project UCs.

The reference architecture distinguishes applications and platforms and captures the mapping of application tasks onto platform resources with an emphasis on application-quality and platform-resource management. The sharing of resources by multiple applications in a predictable and efficient manner is supported by virtualization techniques that support the partitioning of resources into individual budgets that form virtual platforms that can host an application. The challenge behind this is being able to express the budget in independent, predictable and composable form, that the utilization is sufficiently high and that (models of) the resource budgets are simple enough to be effectively used for quality and resource management.

We have developed a component interface model that can coherently describe platform and application components and their interfaces. Compared to existing component models, we emphasize aspects of quality and resource management. From this point of view the characteristic aspects of a component interface are the qualities it offers (covering also costs like power dissipation), the functional inputs and outputs it offers, the provided and/or required resource budgets, and its configuration parameters (see Fig. 3). Components explicitly can have multiple configurations and the interface properties may vary with the configuration or set point of the component. The component model further describes how components can be composed and how the interface of the composition relates to the interfaces of the components from which it is constructed. Fig. 5 shows an example component model in the QRML DSL.

The component model is further equipped with a precise mathematical semantics that defines the concepts in the model [22]. Considering the multi-objective nature of the quality and resource management problem, the semantics is given in terms of Pareto algebra. The semantics relates a component model to a multi-objective constraint optimization problem. In the final year of FitOptiVis, we intend to develop to automatically derive such a constraint problem from a model and use constraint-solving tools, such as Z3, to find optimal solutions. To support the development of component models a domain specific language has been defined, called QRML (pronounced as *caramel*) [21]. The syntax is being developed to allow for a convenient specification of a model by system developers and to support the definition of the interfaces that are important for QRM. Fig. 5 shows an example. The language is supported by tools. In particular an Eclipse plugin can be generated that allows editing of QRML models, model validation and the generation of some derived artifacts, such as a visualization of the component structure. In addition to the Eclipse based tools, to enable low threshold access to the language and tools, a web based tool has been developed that is embedded in the https://qrml.org web site. The language and tool infrastructure are developed with the explicit intention to enable easy refinement of the language and associated tool towards specific application domains. Domain-specific concepts can be embedded in the language and exploited, for instance, for domain-specific code generation. We are investigating this approach for the development and code generation targeting UML/MARTE models and tools to, in addition to the standard QRM related concepts, specify component services and to allow automated generation of C++ code templates for monitoring of properties at run-time. We are also investigating a domain specific extension for timed dataflow models integrating a compositional semantics of timed dataflow models and an analysis model for dataflow graphs that relates their performance to specific abstract processing budget models.

```
component FaceDetection {
   outputs Image out;
   quality Latency lat;
   lat.latency==imgAna.latency;
   requires ImageCapturing imgCap;
   requires ImageAnalysis imgAna;
}
…
component SmartCamera {
   alternatives {
      component SmartCameraNormalMode scnormal;
      component SmartCameraAdvancedMode scadvanced;
   }
}
component SmartCameraNormalMode {
   supports ImageCapturing imgCap;
   supports ImageAnalysis imgAna { latency==25; };
}
component SmartCameraAdvancedMode {
   supports ImageCapturing imgCap;
   supports ImageAnalysis imgAna { latency==50; };
   supports FaceIdentification faceId
      { qual==low; latency==20; };
}
…
typedef Latency {
        integer latency;
}
typedef Quality {
        enumeration {low, medium, high} qual;
}
```

Fig. 5. A QRML snippet of a Biometric Access Control System

The project is also integrating visualization techniques for the component model in the FIVIS visualization framework [25]. FIVIS is also used in WP4 for run-time monitoring of data from the system in operation. This integration allows the run-time data about qualities and resource usage to be intuitively linked to the conceptual component model of the system, to see the configurations that are active, to validate the accuracy of the models. We are investigating virtualization techniques in the context of architecture models and architecture description languages in UML-MARTE, run-time support for dataflow models, and also in the CompSoC [23] composable architecture. It is also investigated how the virtual platforms can be dynamically created, removed and modified in a predictable and composable way. These virtual resources are meant to be coupled to abstract and sufficiently simple budget models that can effectively be exploited for quality and resource management. The methods of implementing virtualization in HW and/or SW are subjects of WP4.

The models allow us to define system models for QRM. We are developing a reference architecture for a quality- and resource-management infrastructure that exploits the models to effectively determine optimal configurations and reconfiguration for a system and initiate reconfigurations when needed. We identify the components that play a role in this architecture, such as brokers, resource managers, application quality managers, etc. We are developing an instance of the reference QRM architecture in the context of the CompSoC platform and intend to verify its suitability also for other platforms in the future.

We are in the process of modelling a number of the UCs in the FitOptiVis project in the component model and in particular in the QRML language to demonstrate how the UCs relate to the overall architecture and to use the feedback from the UCs and from the modelling effort to further refine and improve our models and the QRML language.

B. *WP3 - Design-time support*

The WP3 in FitOptiVis focuses on design-time optimization, deployment and programming strategies, in particular delivered by means of model-driven methods and tools capable of overcoming system design complexity and of predicting, simulating and estimating resource usage at design time. Several issues are tackled here, from HW/SW co-design to SW compilation, made even more challenging by the targeted context: heterogeneous CPS with changing structure and resource needs. To this aim, different methods and tools have been adopted, developed and extended after two years of FitOptiVis, and they can be classified considering the main pursued target (even if some of them are transversal to different classes).

Model-driven engineering for qualities optimization (MDEQ): this class groups tools in charge of estimating and optimizing resources and qualities, such as energy or performance, leveraging on model-driven design. The main MDEQ methods/tools are:

- SD3: a modelling framework for real-time video processing systems distributed from the cloud to the edge, available as free for research. SD3 main features are model capture (based on UML-MARTE), performance analysis and SW code synthesis [6].
- HEPSYCODE: an open source toolchain [7] driving the designer from specification to implementation of heterogeneous parallel dedicated systems. It uses Eclipse MDE technologies, a customized SystemC simulator and an evolutionary genetic algorithm for architecture definition, HW/SW partitioning and mapping activities.
- SAGE: an open–source suite [8] of tools for system formal verification. One of its main features is checking model consistency according to system requirements which, being variable at runtime, need a complex analysis to ensure that they can be always met [4].
- ICON: a framework for closed loop systems with image or data intensive processing. It can support design (Matlab front-end for control and processing), analysis and validation (simulations including physics camera/sensors) and code generation for multi-core platforms.

Programming and parallelization support (PPS): this class groups main FitOptiVis methods/tools that offer to users support for the design and programming of customized processors, as well as for the implementation of embedded systems. Two main methods/tools belong to PPS class:

- TCE: an open application-specific instruction-set toolset for the design and customization of processors, based on the energy efficient Transport Triggered Architecture (TTA). TCE provides a complete retargetable co-design flow from high-level language programs to synthesizable processor RTL and parallel program binaries [10].
- RIE: a methodology and C++ library for component-based implementation of embedded systems [20]. RIE provides support for runtime re-configuration of software components that could have several implementations that are selected at runtime.

Acceleration support (ACCS): this class groups main FitOptiVis tools that offer support for the design and deployment of HW accelerators, including high-level programs to HW tools as well as HW generators oriented to particular applications or specific acceleration techniques. The main tools that belong to ACCS class are:

- MDC: an open-source automated dataflow-to-HW tool for the generation, system integration and management of Coarse Grained Reconfigurable (CGR) datapath-accelerators [11][12][5].
- NeuDNN: a configurable C/C++ library, providing the APIs to seamlessly execute the CNN in SW or accelerated with the NEURAghe solution [13].
- DTRC: an automated dataflow-to-HW tool for configuration and system integration of design time resources for Zynq and Zynq Ultrascale+ systems with Debian OS and HW accelerators [14].
- DTRiMC: a tool for system integration of IPs designed, modelled and validated in Xilinx Model Composer and Xilinx System Generator for DSP [15].

All the presented tools/methods classes are meant to be assessed over different UCs. Table 1 is presenting the preliminary WP3 technologies versus UCs coverage matrix.

TABLE I. WP3 TOOLS ADOPTION IN UCS.

	1	2	3	4	5	6	7	8	9	10
MDEQ	x									
PPS		x				x				X
ACCS	x							x		X

Currently, special emphasis is dedicated to design time, optimization, deployment and programming strategies. Improvements, extensions and advancements on the described design-time tools/methods are expected also for the third year of the project.

C. *WP4 - Run-time support*

WP4 in FitOptiVis aims at developing solutions to provide the runtime management of applications, while considering a diverse set of heterogeneous platform components and configurations. The provided solutions need to offer means for resource managers to control application parameters linked to individual quality attributes and to manage resources assigned to an application. Within the project, such a task has been implemented with reference to MAPE-k loop, which generalizes the concept of a control loop (Ctr-L) for adaptive systems. Such Ctr-L, described in the following, can be nested to form a hierarchy of Ctr-Ls operating at different time scales:

- *Managed-Latency Edge-Cloud Environment (MLEC)* – The goal of this Ctr-L, described in Fig. 6, is to provide soft real-time guarantees to applications executed on top of a virtual environment. A developer is required to provide the application and its soft real-time requirements. For monitoring and analysis FIVIS, a platform for data storage, analytics and visualization, has been developed.

- *CompSoc platform (COMPSOC)* - The goal of this Ctr-L, described in Fig. 7, is dynamically managing quality and resources of composable and analyzable hard real-time applications running on a system-on-a-chip, which is abstracted through a Virtual Execution Platform.
- *Reconfiguration in Processor-Coprocessor systems (RPCP)* – The goal of this Ctr-L, described in Fig. 8, is to deal with runtime adaptability at hardware level [17][27], specifically referring to multi-purpose co-processing units. A developer is required to describe its custom accelerators with a tool named MDC [5][12], based on dataflow models. The Ctr-L exploits coarse-grained functional and non-functional reconfiguration. For monitoring part AIPHS [3][16], a framework to build hardware monitoring systems with controllable intrusiveness, has been developed.

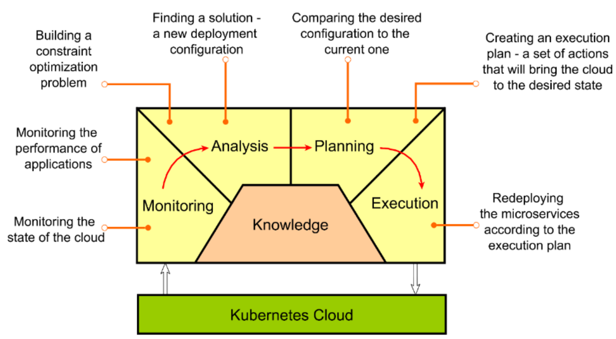

Fig. 6. Ctr-L for reconfiguration in MLEC

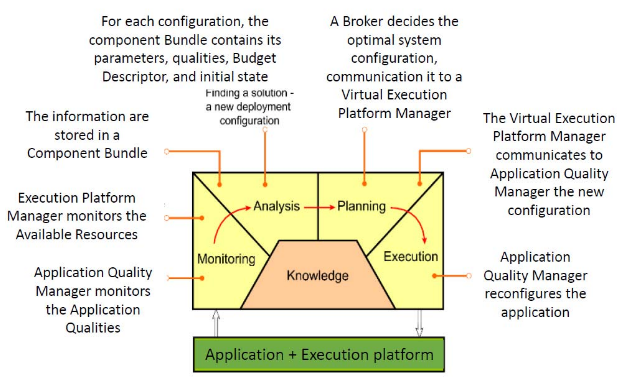

Fig. 7. Ctr-L for reconfiguration in CompSoC.

Similar Ctr-Ls for adaptive systems have already been explored by other studies at the state of the art and projects, as in [18]. What is new here is the assessment at various level of abstraction and the exploitation, in most of the cases, of the Virtual Reference Platform developed in WP2 to take runtime decisions upon adaptation and quality- and resource-management. The support for development of resource managers is enforced with the provision of standardized APIs to harness all available resources. To this end, a distributed OpenCL-centric heterogeneous device runtime software stack (POCL) has been developed [19], which provides a unifying backbone to applications relying on hardware accelerators, both local and remote. This work enabled the possibility to further adapt a high-level programming model, namely OpenMP [26], employing the developed stack. Finally, different platform templates (PTL) with easy-to-use implementations and support to prototyping, tailored for heterogeneous platforms (e.g., Xilinx Zynq), have been provided.

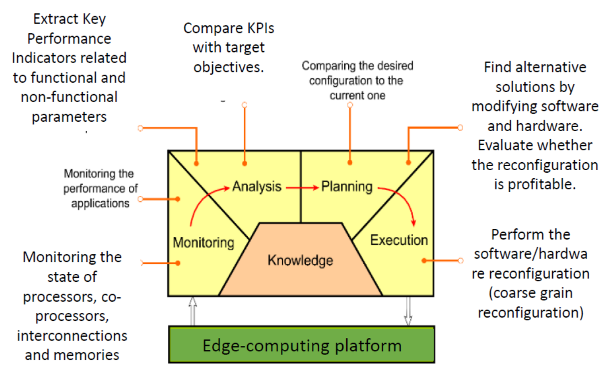

Fig. 8. Ctr-L for RPCP.

The reported solutions are still under development, and the ten use-cases are currently starting to make use of them to satisfy requirements and to further elicit improvements of the same FitOptiVis solutions. During the last year of the project we expect to refine and complete all the described technologies according to the feedback gathered from the UC. Table 2 is presenting the preliminary technologies versus UCs coverage matrix.

TABLE II. WP4 TECHNOLOGIES ADOPTION IN UCS.

	1	2	3	4	5	6	7	8	9	10
MLEC			x	X						
COMPSOC						x				
RPCP	x									
POCL		x	x							
PTL								x		

D. WP5 - Devices and components

The aim of WP5 within the FitOptiVis project is to develop high-performance, energy-efficient processing and communication devices and components that adhere to the reference architecture. In particular, components defined and developed in this WP represent the primary building blocks of the FitOptiVis framework that have to guarantee high-performance edge processing in a scenario characterized by limited resources and/or geographically distributed. Moreover, each component is meant to be characterized, where appropriate, by different levels of abstraction and set points in order to be compatible with the resource management of the run-time support developed in WP4.

For the previously described purposes the WP is carrying out several parallel activities. The first one is devoted to the definition and description of both i) state of the art commercial components and algorithms, capable of addressing the needs of FitOptiVis UCs, and ii) novel software (new algorithms and software IPs, i.e. next generation distributed digital and SW components for MRI control and data processing) and hardware (sensing and actuating components, i.e. smart industrial cameras; enhanced image acquisition devices, with on board processing capabilities) IPs. Since in the context of FitOptiVis, dealing with edge to cloud solutions, the exploitation of low-power and low-energy hardware for high-performance is of paramount importance, the second activity WP5 is carrying out is related to accelerators. Novel co-processing elements, to be used as a part of the heterogeneous processing platforms, are developed within FitOptiVis: i.e. an 8xSIMD floating point accelerator, a license plate detector, coarse-grained application-specific reconfigurable accelerators. The activities are focused on the definition of proper communication mechanisms for edge-to-edge and edge-to-cloud processing pipelines, allowing local pre-processing solutions for minimizing communication load and enabling fast point to point local interactions. Edge distributed processing is also exploited for cutting down time-to-decision and improving security and privacy.

All components proposed in WP5 will bring IPs innovations that can be classified as hardware, software and communication:

- *HW IPs Innovations*: Image and video pipelines (acquisition, composition, transmission, rendering), image and video detection and classification (features extraction, neural network-based classification), MRI application.
- *SW IPs Innovations*: Coding Engines and Image Processing Libraries, Classifiers (as Behavioral Analysis or Face Recognition Systems), Remote/Distributed Monitoring (like Person Tracking or Personal Assistant).
- *Communication IPs Innovations*: Low latency network communication, real-time distributed control systems, communication monitoring support, seamless big data communication, heterogeneous traffic support, secure embedded system-of-systems environment.

TABLE III. WP5 TECHNOLOGIES ADOPTION IN UCS.

	1	2	3	4	5	6	7	8	9	10
HW	x	x		X	x	x	x	x		x
SW	x	x	X					x	x	x
COMM	x	x	X					x	x	

All the previously itemized solutions are exploited in many UCs. Table 3 is presenting the preliminary technologies versus UCs coverage matrix. In summary, the current composition of FitOptiVis components library is the following: 12 Processing and acceleration components; 10 Software components; 4 Communication oriented components; 2 Miscellaneous (Sensor/actuators/composed). Updates, modifications and improvements are expected in the third year, according to the incremental development of the UC demonstrators and their feedback.

VII. REMARKS AND PLAN

FitOptiVis is about to start its final year, partial demonstrators will be available soon and the third year will lead us to consolidate all the innovative technologies, libraries and tools that have been described in this paper. *"This work is part of the FitOptiVis project [1] funded by the ECSEL Joint Undertaking under grant number H2020-ECSEL-2017-2-783162".*

REFERENCES

[1] Al-Ars, Z., et al. (2019). The FitOptiVis ECSEL project: highly efficient distributed embedded image/video processing in cyber-physical systems. ACM Int. Conf. on Computing Frontiers.

[2] Muttillo, V, et al. (2018). Hepsycode-RT: A real-time extension for an ESL HW/SW co-design methodology. ACM Intl. Conf. Proceeding Series.

[3] Moro, A., et al. (2018). Hardware performance sniffers for embedded systems profiling. Intl. Work. on Intelligent Solutions in Embedded Systems.

[4] H. Alizadeh Ara, et al. (2018) Compositional Dataflow Modelling for Cyclo-Static Applications. Conf. on Digital System Design.

[5] C. Sau, et al. (2015) Reconfigurable coprocessors synthesis in the MPEG-RVC domain. Conf. on ReConFigurable Computing and FPGAs.

[6] Single source design framework, http://umlmarte.teisa.unican.es/

[7] The Hepsycode project, http://www.hepsycode.com/

[8] Sage repository, https://gitlab.sagelab.it/sage

[9] Narizzano, M., et al. (2019) Property specification patterns at work: verification and inconsistency explanation. Innovations Syst Softw Eng 15.

[10] TTA-based Co-design Environment (TCE) tools, http://openasip.org

[11] Multi-Dataflow Composer design suite, github.com/mdc-suite/mdc

[12] Palumbo, F., et al. (2017) Power-Awarness in Coarse-Grained Reconfigurable Multi-Functional Architectures: a Dataflow Based Strategy. Jrnl Sign Process Syst.

[13] Paolo Meloni, et al. (2018) NEURAghe: Exploiting CPU-FPGA Synergies for Efficient and Flexible CNN Inference Acceleration on Zynq SoCs. ACM Trans. Reconfigurable Technol. Syst..

[14] Jiři Kadlec, et al. App note and Eval package [Online], http://sp.utia.cz/index.php?ids=results&id=FitOptiVis-te0726-SDSoC-2018_2

[15] J. Kadlec, et al. App note and Eval package [Online], http://sp.utia.cz/index.php?ids=results&id=FitOptiVis-te0808-SDSoC-2018_2

[16] G. Valente et al. (2016) A Flexible Profiling Sub-System for Reconfigurable Logic Architectures. Conf. on Parallel, Distributed, and Network-Based Processing.

[17] G. D'Andrea, et al. (2019) Self-adaptive loop for CPSs: is the Dynamic Partial Reconfiguration profitable? Mediterranean Conf on Embedded Computing.

[18] F. Palumbo, et al. (2019) Hardware/software self-adaptation in CPS: the CERBERO project approach, Intl. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation.

[19] Portable Computing Language, http://portablecl.org/

[20] https://s3d.unican.es/rie/

[21] Quality and Resource Management Language. https://qrml.org.

[22] M. Hendriks, et al. (2020) Interface Modeling for Quality and Resource Management.

[23] K.G.W. Goossens et al. (2017) NoC-based multiprocessor architecture for mixed-time-criticality applications. Handbook of hardware/software codesign (Springer).

[24] http://cyberphysicalsystems.org/

[25] https://github.com/smartarch/ivis-core

[26] http://www.openmp.org

[27] G. Valente, T. Di Mascio, G. D'Andrea and L. Pomante, "Dynamic Partial Reconfiguration Profitability for Real-Time Systems". IEEE Embedded Systems Letters, 2020.