Towards Independent in-Cloud Evolution of Cyber-Physical Systems

Amir Taherkordi and Frank Eliassen
Department of Informatics
University of Oslo, Norway
Email: {amirhost,frank}@ifi.uio.no

***Abstract*—The capabilities of Cyber-Physical Systems (CPSs) are increasingly being extended towards new composite services deployed across a range of smart sensing and controlling devices. These services enable the emergence of multiple end-to-end cyber-physical scenarios, formed dynamically based on their demands, e.g., disaster recovery systems. In such scenarios, each cyber-physical flow may be composed of a large number of physical services with composition challenges such as high dynamism of CPS platforms, rapid development and scalability, and real-time and reliable processing and controlling tasks. Cloud computing enables new perspectives in the design and operation of CPSs, including consolidating and sharing physical services among different applications, auto-scaling computing and communication, and designing and maintaining multiple in-Cloud CPSs dynamically at the same time. In this paper, we present a new architectural approach to address the key concerns of a new generation of CPS services whose functionalities reside in-Cloud (cyber), and on devices and systems (physical). In particular, this design space is focused on principles that allow *in-Cloud evolution* of CPS services, including dynamic in-Cloud service composition and distribution, virtualization of physical services and devices, and the dynamic creation of CPS ecosystems. In this design model, the in-Cloud cyber part may evolve independently, while the on-device cyber and physical platform still work closely together and provide the basic CPS services.**

I. Introduction

CPSs are primarily designed for sensing, processing and controlling purposes locally on devices or in a local private network [1]. The recent applications of such platforms in large scale, mobile and distributed systems, such as disaster, traffic and healthcare systems are introducing a new class of CPSs that fosters the development of global smart applications [2], [3]. In this vision, a CPS application will achieve global visibility to all CPSs on devices deployed across a vast area (e.g., a region or a city), enabling the provision of smarter services and making better control decisions to react to physical phenomena. Moreover, CPS applications can share their services with each other in order to improve functional and non-functional attributes of systems, e.g., in terms of safety, smartness, and rapidness in service provisioning. Beyond these, heterogeneous services offered by different CPSs can be integrated to form new CPS ecosystems while they are cooperating with existing applications.

These can be realized by empowering the device-level CPS services with capabilities to evolve independently in a flexible and powerful computing platform such as the Cloud. Cloud computing enables the consideration of new features in the design and operation of CPSs, including consolidating and sharing physical hardware among different applications, sensing and actuating resources on-demand, customizability through modular composition, rapid development and scalability, resiliency, and performance based on user needs. The prevalence of the Cloud and its benefits allow the cyber-part of CPSs to evolve independently and distribute their services in the Cloud. In this design perspective, cloud computing becomes a metaphor for information acquisition as a service of CPSs, rather than the traditional notion of platform- or software-as-a-service [3].

To efficiently benefit from the development and migration of CPSs in the Cloud, careful attention should be paid to the core design aspects of CPS services, including service types and properties, and potentials for growing new CPS services in the Cloud. This area of research is still in its infancy and there are currently only a few studies in this area. For example, [3] aims to present integration aspects of CPSs into the Cloud from a high-level viewpoint, such as CPS virtualization, interconnectivity between CPS services in the Cloud, and privacy. [4] focuses on cloud-based industrial applications and presents a service-oriented architectural platform to cover the basic needs for monitoring, management, and integration of CPS services. Both above architectural models provide valuable and overall insights into a wide range of design aspects of cloud-based CPS service development, however they do not investigate thoroughly the service design concerns and the evolution opportunities when moved to the Cloud.

In this paper, we focus on Cyber-Physical Cloud Computing (CPCC) and present design choices, addressing the key concerns of a new generation of CPS services whose functionalities reside in-Cloud (cyber), and on-devices (i.e., end devices and network infrastructure devices). In particular, we propose an architectural model and corresponding design patterns focused on principles for *in-Cloud evolution* of CPS services. The concept of evolution, in this paper, refers to the gradual development of CPSs through dynamic in-Cloud service composition and distribution, virtualization of physical services and devices, and the dynamic creation of CPS ecosystems. Additionally, the proposed model is characterized by *independent* evolution of the in-Cloud cyber part, while the on-device cyber and physical platforms still work closely together and provide the basic CPS services. In this way, the architectural framework provides an environment for rapid development, evolution, and integration of CPSs, composed of a set of cloud computing based sensing, processing, and control services. To this end, we study the different types of CPS services (i.e., sensing and control services) and present key design choices for in-Cloud evolution of CPSs and the integration of their services.

The rest of the paper is outlined as follows. Section II discusses the evolution aspects of CPSs in CPCC. In Section III, we focus on the key CPS service types and present architectural concerns and design choices for the in-Cloud evolution of CPSs. We discuss some recent efforts on designing in-Cloud CPSs in Section IV. Finally, Section V concludes this paper and identifies some future work.

II. In-Cloud CPS Evolution

The overall goal for evolving CPSs is to introduce a new class of such systems that can be dynamically created and composed of a wide range of individual and heterogeneous CPSs, while the on-device cyber-part tightly interacts with the physical part with remote interactions to the in-cloud part. In this vision, we expect the in-cloud CPS to evolve in the following dimensions: virtual representation of physical system components and services (e.g., sensing and actuation services), collaborative sensing and actuating through distributed sensing and control, and seamless and cross-layer integration of heterogeneous CPS services. Considering all these design concerns in a single architectural model will enable us to design new CPSs that evolve independently from the original physical and cyber parts. We discuss, in this section, the scope of CPS evolution and the associated design concerns in order to further highlight the benefit of in-Cloud CPS services in the next section.

Virtualized representation of CPS services. Virtualization of CPS services refers to the virtualized representation of CPSs in terms of their components and services, such as sensors, actuators, control and process. In addition, it promises accessing those services and components through a relevant service virtualization mechanism and processing of data collected from a potentially large number of physical devices in a scalable, reliable and on-demand manner. For example, virtual aerial autonomous vehicles form virtual mobile CPSs whose nodes can be created dynamically at flight time and that can aggregate information efficiently and quickly, e.g., by migrating processing from one real vehicle to another [5]. Cloud-based CPS services allow such vehicles to be used in multiple applications at the same time, leading to more efficient use of CPS resources, resiliency during unexpected situations, and flexibility to adapt to and expand the CPS functionalities as end users expect. In such situations, we face new CPSs, *evolved* by absorbing many real vehicles with highly heterogeneous properties, and processing thousands of concurrent sensing and control information such as surveillance, tracking, sending instructions to targeted vehicles, etc. The new level of indirection provided by virtualization can also enable the independent development of CPCC platforms.

Distributed Sensing and Actuating. When moving from local and disconnected physical devices towards a global and integrated smart sensing and controlling system, we need to change our vision from a restricted and localized system to a distributed CPS design model, allowing the distribution of sensing and control tasks. Specifically, a more sophisticated cyber model will emerge in which each sensing or control functionality may reflect a distributed cooperation of a set of actual physical devices. For example, in a medical CPS (MCPS) [2], the system often consists of networks of medical devices and computer systems cooperating to provide care to patients from home to the hospital. The set of device types and the algorithm which defines how those devices should distribute and coordinate the treatment tasks will introduce new requirements in terms of CPS evolution. In-cloud CPS development allows processing of complex distributed coordination models at the same time for multiple end-user applications. Distributed control is of particular interest in this context as it might introduce new challenges such as dependency between autonomous control entities and decentralized algorithms for coordination.

Multi-level composition of CPS services. To efficiently use the CPS capabilities, we need a model of compositionality that cuts across different abstractions of CPS services provided by heterogeneous cyber and physical platforms of CPSs. By different abstractions, we mean that a composition process in CPS environments not only includes the interactions among atomic services or composite services, but also embraces a more complex model of composition between virtualized platforms and other typical service types. Moreover, this model should address the composition of sensing and control services at different levels of CPS ecosystems, e.g., a service that is composed of a control service and a sensing service *collectively provided* by a set of co-located services. Introducing this level of composition diversity and abstraction helps services evolve rapidly and independently in the Cloud. Figure 1 depicts these two composition models. Realization of such models requires a flexible and adaptive approach that mitigate composition challenges of both subsystems and systems-of-systems.

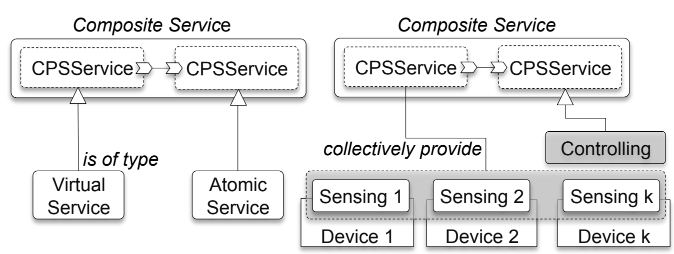

Fig. 1: Samples for multi-level compositional models

Dynamic CPS ecosystems. The integration of co-existing CPSs can provide new evolution opportunities by rapidly forming a new class of CPSs which reuses the existing functionalities and resources, and introduces cross-domain CPSs. For example, a dynamic disaster recovery system can be formed as a combination of existing CPS infrastructures, such as healthcare systems, transport systems, and power management systems. As another example, homecare scenarios may involve not only home-based medical devices, but also tiny body sensors that measure for example heart beat rate. The integration of such sensor devices and MCPS will introduce a new homecare ecosystem in which both the physician and the patient can control locally and remotely the care procedure, in addition to automating the control of MCPS. Therefore, we will need new distributed business processes, integrating different CPS services from multiple application domains for a given purpose—CPS ecosystems. The in-cloud availability of CPSs significantly reduces the required effort to develop such ecosystems thanks to the global and rapid accessibility of CPS services in the Cloud.

III. PROPOSED ARCHITECTURAL MODEL

In this section, we investigate the main architectural components and design choices for introducing in-Cloud CPSs and evolution of their services. A typical architecture for CPCC may address numerous concerns and the design scope can be broad. For example, integration, discovery, process monitoring, life cycle management and security are some of those concerns [4], to name a few. In this paper, we do not intent to present a generic architectural approach to tackle all these concerns. Rather, we aim to explore the aspects of such an architecture that contribute to the efficient evolution of CPS systems in the Cloud, founded on a service-based vision of CPSs. In particular, the proposed architecture attempts to analyse CPS services and enhance their capabilities in order to facilitate the evolution of CPSs in the Cloud.

This requires first to study the different types of services a typical CPS can provide, ranging from sensing and processing, to controlling and actuating. We categorize these services into three main types: *sensing*, *processing*, and *controlling*. We believe that any CPS functionality, accessible to the outside world, can be described by one of these service types. The aspects of evolution, described in the previous section, will reveal new design choices for the above service types when envisioned at a large scale and hosted in the Cloud. In this paper, we study the two key service types: sensing and control.

A. Sensing Service Design

The first service type is the sensing service, monitoring physical conditions and encapsulating single or aggregated measurements as a web service. As an example of sensing services, a pulse oximeter is a sensor that continuously monitors the heart rate and blood oxygen saturation (SpO_2) of a patient, and transmits the readings to a controller of a Patient Controlled Analgesia (PCA) infusion pump [2]. The latter is a kind of device widely used for pain control of post-operative patients. Generally, a sensing service can be either pull-based or push-based. The former refers to the traditional way of calling web services, while the latter makes more sense in embedded and event-based environments where a third party system subscribes to the receipt of service data. There are a number of popular WS-based solutions for CPSs (generally for embedded systems) that support the latter, such as Devices Profile for Web Services (DPWS) with the support of WS-Eventing over SOAP [6].

Cloud-level visibility of sensing services will allow us to access a huge number of different type of cross-domain CPS sensing functionalities in a scalable, on-demand, efficient, and reliable manner. This also encourages the development of new sensing service abstractions to ease the integration, sharing, management, and improvement of the quality of CPS sensing services. To achieve this, we introduce three types of sensing services: *physical sensing*, *virtual sensing*, and *distributed sensing*. Whereas the first model refers to the traditional way of invoking sensing services from known devices (i.e., address-based invocation), the other two types are generally focused on collaborative models of sensing service provision.

Virtual sensing services promise an encapsulated and abstract sensing model created out of variety of sensing services that are semantically correlated. In this way, we obtain a scalable service model that can combine heterogeneous sensing services spread out in a monitored area. For example, a vehicle location tracking service can be composed of a car navigation service, and the GPS-based location service offered by the driver's smart phone, while an *IsDriving* sensor service could be composed of a combination of the GPS and the accelerometer of the smart phone, and even using its microphone. There are a number of works addressing virtual sensor in WSN application domains. For example, in [7], a *sensor cloud* platform is proposed to virtualize wireless sensors based on the covered geographical area and provide sensing as a service to users. [8] presents a sensor cloud infrastructure which virtualizes physical sensors to enable sharing of them. Furthermore, end users can create virtual sensor groups dynamically by selecting the templates of virtual sensors or virtual sensor groups, prepared by infrastructure administrators.

In our architectural model, we aim to devise a generic and extensible technique for virtual sensing abstraction. A virtual sensing service is defined as a product of spatial, temporal, or semantic transformation of raw sensing services. To this end, the notion of ontology is used to capture the information about physical sensing services and their relations. Figure 2 gives an overview of semantic-based virtual sensing services. Having a domain-specific ontology for each virtual service helps unifying virtual service presentation and allows service and information reuse. To define a virtual sensing service, we suggest to use Sensor Model Language (SensorML) [9]—a standard model to describe sensor systems, sensor groups, and processes associated with sensor observations in an XML-based schema. The in-Cloud processing of the semantic description and realization of virtual services will allow the development and dynamic maintenance of a huge number of such services for cross-domain use cases. For example, in a vehicle monitoring CPS, we can associate the location data to a domain-specific location ontology, e.g., in the disaster recovery system. This enables navigation through a set of linked sensing services and querying services based on their deployment and other attributes as well as their physical attributes.

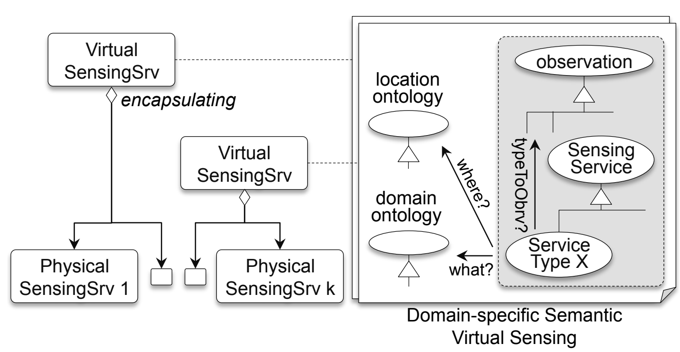

Fig. 2: Overview of the design of virtual sensing services

The other form of sensing service is the distributed collaboration and interaction among different service types to achieve an overall goal. For example, *homecare* scenarios may involve not only home-based medical devices, but also tiny body sensors that measure for example heart beat rate. The integration of such sensor devices and MCPS will transform the homecare sensing services to a distributed sensing model through which different stakeholders (e.g., physicians, other medical devices, and hospital systems) can observe different sensing conditions and take appropriate actions (i.e., control and actuation). Thus, such services essentially represent a distributed model of collaboration, which occurs either locally between physical devices, or remotely in the Cloud.

This will lead to a model that binds the physical sensing services into a *distributed service* instance and orchestrates interaction among the services to achieve a given distribution goal. Figure 3 provides an overview of the proposed model, where the distribution manager can be deployed either on the local network of devices, or remotely in the Cloud. The distribution manager provides an execution environment for distributed sensing services, featuring components such as coordination model, orchestration, and algorithms required for describing and handling distributed interactions between physical sensing services. The in-Cloud execution of the distribution manager will bring more flexibility to the system as we can have multiple instances of this component with different distribution concerns running in parallel for different applications. However, moving this process to the edge of local network as a middleware framework will impose less communication overhead.

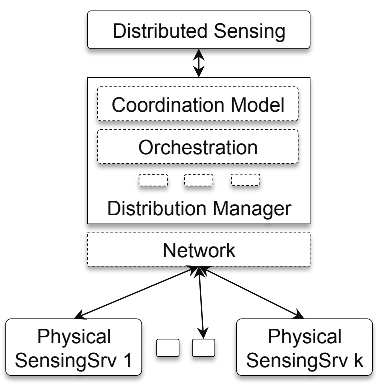

Fig. 3: Overview of core components for distributed sensing

B. Control Service Design

Monitoring and control are considered key factors for achieving the visions in several CPS dominated areas such as automotive electronics, smart-grid, building control, and medical and health infrastructures. The traditional design approaches for control systems take into account different concerns such as uncertainties, physical disturbances, robust control, and adaptive control [10]. CPSs reflect the growing level of integration of control systems with computing technologies and make the above challenges of particular interest when the cyber space is incorporated into the physical world, such as in control theory, algorithms, and control computation models. In this work, we focus on the system-level and service design scope of control in CPSs, as well as the benefits of in-Cloud control services distribution and evolution.

From a system viewpoint, control services in CPSs can be classified into [11]: i) local control, ii) distributed control, and iii) supervisory control. *Local control loop* refers to the low-level automated control that handles a certain control logic locally with a relatively low number of actuators and sensors. The logic of control may be continuous or discrete, and in many cases it may require low latency and short sample times. In contrast, *distributed control* cannot be executed by a single device (controller). It refers to a form of control in which the control process is divided into sub-processes and executed on platforms that are far away from each other, geographically or architecturally. *Supervisory control* is an enhanced, yet slower model of a local control loop, executed at a higher system level based on information from more than one subsystem. This model of control typically uses aggregated process values as input and actuates through changing the set point of a local control loop, without direct access to sensors or actuators.

From an architectural viewpoint, the control model in CPSs is similar to that of industrial control systems such as power systems. Inspired by these systems [10], we propose a layering perspective toward the design of control services in the context of this work. Figure 4 illustrates the different design aspects of control services, including the architectural layers, control-specific design concerns, as well as the part of the architecture residing in the Cloud. At the physical control level, multiple physical and control components are introduced, including sensors, actuators, and other intelligent control components. On top of this, the communication layer, along with network components are in charge of addressing network-level controlling concerns such as collaborative reasoning and actuating. Supervisory layer, as part of the intelligent control level, coordinates all lower layers by issuing appropriate commands in cooperation with the management layer, making higher level decisions based on predefined criteria such as CPS resource models or human-made factors. In-Cloud control services can evolve using network-level and intelligent control components with respect to the control design concerns discussed below.

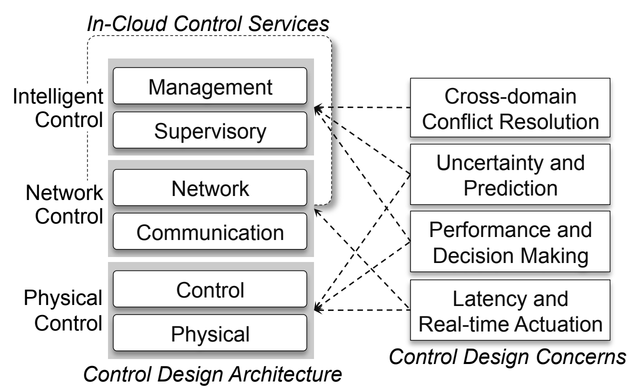

Fig. 4: Generic architecture of in-Cloud CPS control system and associated design concerns

Each of the above architectural components is associated with crucial design concerns for developing the control software. A prime concern is related to latency and real-time control operations. Real-time actuations in CPSs require a combination of different operations such as sensing, processing, communication, and actuation. This raises the need for a unified theory of *real-time operation* that includes existing results and novel solutions from these research areas. When it comes to supervisory, several trade-offs among sub-objectives (e.g., reliability versus efficiency of actuation) will occur. Therefore, components in the intelligent control should be empowered with relevant modelling and control concepts that allow meeting the sub-objectives and coordinating system-wide *performance*. In particular, distributed control algorithms and decision making models should be central to the design of physical and intelligent control services, e.g., in Supervisory Control and Data Acquisition (SCADA) systems—used for monitoring and controlling power grids [12]. CPS control services must consider, reason about and compensate for *uncertainty* based on the potential effects of the cyber part. This requires special attention to be given to service verification based on uncertainty, control/computing co-design for safety, and feedback control loops. Finally, with multiple in-Cloud CPS systems with different control models and decisions, we need to carefully consider the conflicts arising due to this difference and resolve them through techniques such as operation prioritization, semantic reasoning or human feedback.

By migrating control services to the Cloud, CPSs can enable broad and distributed controls over complex and large industrial processes through a heterogeneous network architecture of control and sensing services. Considering the aforementioned design concerns, the role of the Cloud and its benefits are highlighted even more when offloading heavy reasoning, prediction, and coordination tasks to the Cloud.

C. Service Composition and CPS Ecosystems

To integrate heterogeneous sensing and controlling capabilities with each other, as well as with third-party services, CPS services should be open, interoperable, and standard-compliant. This also facilitates the generation of new CPS-enabled distributed business processes that further integrate the real world with the digital world. With a service-oriented design of CPS functionalities and their in-Cloud accessibility, business processes can be designed independently from local CPS processes in large-scale infrastructures that accommodate a diversity of IT systems and CPS devices. As discussed before, we consider two concrete goals for composing CPS services based on their characteristics. First, the proposed architecture should address multi-level composition requirements and integration of virtual services. Second, to foster the development of CPS ecosystems, the service composition framework should facilitate the in-Cloud integration of different services in order to create new on-demand CPSs.

To address the first aspect of composition, we propose to rely on service *orchestration* and adopt a generic composition framework that supports dynamic service composition. Additionally, the service composition solution should support semantic service discovery and composition in order to allow dynamic and flexible creation of virtual services. By describing the specification of virtual or other composite service types (by the CPS service developer), the composition framework should perform automatic discovery, matching and composition of a set of services that together fulfill the service requirements. In addition, the adopted framework should be enriched with meta information about non-functional requirements for composite services which are crucial in CPCC, e.g., communication overhead and timing. There are a few popular approaches for addressing dynamic composition of web services, such as eFlow [13], METEOR-S [14], and DynamiCoS [15]. Among these, we believe that DynamiCoS can better meet the above requirements, such as semantic definition of composite services and specifying non-functional requirements.

To create on-demand CPS ecosystems, we need a highly flexible and distributed model of service composition and interaction. Service *choreography* is known as a standard approach for service compositions in which each involved party describes its own part in the interaction of services. CPS ecosystems can rely on the principles of service choreography, where CPS services are distributed across heterogeneous applications and devices, and the the overall behaviour of the target ecosystem is choreographed. The key design challenge in this context is to consider the special requirements and constraints of each participating CPS application or service in a choreography flow, such as service execution deadline and reliability (i.e., service level agreement). The above concerns are summarized in Figure 5.

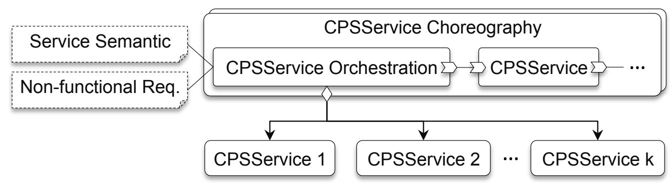

Fig. 5: The composition levels of CPS services

D. Overview of Proposed Architecture

Figure 6 shows the main components of the proposed architecture in the Cloud, as well as their relationships. CPSService is the key element of the architecture which can be reified into three different service types: Sensing, Controlling and Processing (we studied the first two types in this paper). Each of these service types is further broken down into concrete types discussed earlier in this section. Whereas Physical Sensing and Selective Control refer to invoking atomic services provided by certain devices, the other service types deal with collaborative service models such as virtualization or distribution. At the top of the figure, CPS applications are located, where each application has its own service composition and business process model, as well as its own constraints for configuring and using a CPSService. Moreover, this allows modeling and implementing cross-sector CPS ecosystems by integrating dynamically co-existing applications and their in-Cloud CPSServices.

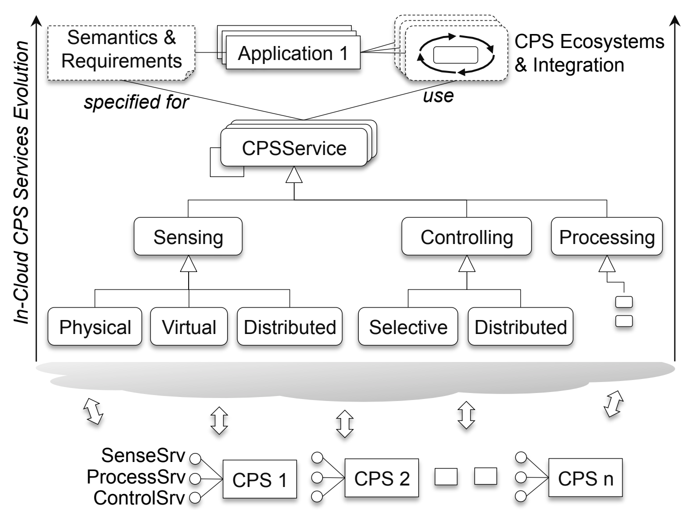

Fig. 6: Proposed architecture for in-Cloud CPS evolution

Considering the interactions between the architectural components and physical platforms, the key requirement is seamless, widely-accepted and potentially cross-sectoral interoperability at application-, service- and device level. At the same time, a right interoperation choice serves as an application enabler for cross-domain CPS systems (i.e., CPS ecosystems) as envisioned, e.g., for disaster management and smart grid. Therefore, as we implicitly mentioned before, the architecture is built on the principles of Service-Oriented Architecture (SOA). With the benefits offered by modern SOA, CPS functionalities can be offered as one or more services of varying complexity, hosted in the Cloud and composed by other (potentially cross-layer) services [4]. This leads to a flat information-based architecture that depends on a big variety of CPS services and their composition [16], contrary to the traditional hierarchical view.

IV. Recent Attempts on CPCC Architecture

CPCC is increasingly gaining attention thanks to its scalable, on-demand, and reliable provision of physical services. This has triggered several lines of research to address challenges, such as architectural patterns for integration, virtualization of physical components in the Cloud, security, privacy, and Cloud-assisted situation-awareness and decision support.

The first and the most significant issue, in this context, is the software architectural model for CPCC. A few research efforts have been carried out to explore the core architectural components for migrating CPSs to the Cloud. [3] investigates a diverse set of requirements for CPCC (e.g., timeliness, reliability, and security) and proposes a conceptual architecture framework for CPCC with an special emphasis on information modeling from low-level sensing and actuating data to the cloud-level interpretation of those data elements. The other prominent work is done in the context of the IMC-AESOP research project [4], presenting main architectural services for cloud-based industrial applications (i.e., a type of CPCC), such as monitoring, management, data handling, and integration. Both these works deal with high-level design challenges, while the architecture presented in this paper focuses more on service design aspects of CPCC and their evolution in the Cloud.

Additionally, there are some works that address particular research challenges in the integration of manufacturing and industrial systems into the Cloud, e.g., vehicles and robots. For example, in [5], the V-Cloud architecture is presented to enable cloud computing system of vehicles in order to meet safety and comfort requirements for the driver. As another work in this category, in [17], Jiafu et al. design a vehicle CPS and mobile cloud computing integration architecture to provide mobile services for potential users such as drivers and passengers to access mobile traffic cloud. Both these architectures are particularly designed for addressing challenging in vehicle and transportation domains, such as verification of intelligent automobiles, driver-vehicle interactions, and dynamic vehicle routing. In [18], the authors introduce the notion of information-acquisition-as-a-service and the associated software design challenges for CPSs based on virtualized versions of aerial vehicles deployed on a fleet of helicopters. The concept of virtual vehicle in their work refers to the virtualization of the complete system including system software and I/O, which is different from the goal of our work.

V. Conclusions and Future Work

In this paper, we focused on the migration of CPS services to the Cloud and opportunities for such services to evolve in the Cloud independently from the physical services. To achieve this, we studied the in-Cloud evolution aspects of CPS services and proposed an architectural model which highlights design choices for different CPS service types, including sensing and controlling services. The architecture also supports on-demand development of CPS ecosystems through standard, dynamic and flexible service composition models such as service orchestration and choreography. In our future work, we plan to study the performance and optimization concerns for realization of cloud-level virtual sensors and distributed control, which are central to the in-Cloud evolution of CPSs.

References

[1] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10. ACM, 2010.

[2] I. Lee *et al.*, "Challenges and research directions in medical cyber physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, Jan 2012.

[3] E. Simmon *et al.*, "A vision of cyber-physical cloud computing for smart networked systems," NIST Interagency 7951, Tech. Rep., 2013.

[4] S. Karnouskos *et al.*, "A soa-based architecture for empowering future collaborative cloud-based industrial automation," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct 2012.

[5] H. Abid *et al.*, "V-cloud: Vehicular cyber-physical systems and cloud computing," in *Proceedings of the 4th Symposium on Applied Sciences in Biomedical and Communication Technologies*, 2011.

[6] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," *Industrial Informatics, IEEE Transactions on*, vol. 1, no. 1, pp. 62–70, Feb 2005.

[7] S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors," *Software, IEEE*, vol. 31, no. 2, pp. 70–77, Mar 2014.

[8] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing," in *Network-Based Information Systems (NBiS), 13th Int. Conf. on*, 2010.

[9] Open Geospatial Consortium, Inc., "Sensor Model Langua g e (SensorML) for In-situ and Remote Sensors," http://www.opengeospatial.org/standards/sensorml.

[10] Q. Zhu and T. Basar, "Robust and resilient control design for cyber-physical systems with an application to power systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, Dec 2011, pp. 4066–4071.

[11] A. W. Colombo *et al.*, Eds., *Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach*. Springer, 2014.

[12] M. Ilic, L. Xie, U. Khan, and J. Moura, "Modeling of future cyber physical energy systems for distributed sensing and control," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 40, no. 4, 2010.

[13] F. Casati *et al.*, "Adaptive and dynamic service composition in eflow," in *Proceedings of the 12th International Conference on Advanced Information Systems Engineering*, ser. CAiSE '00, 2000.

[14] K. Verma *et al.*, "The meteor-s approach for configuring and executing dynamic web processes," University of Georgia, Tech. Rep., 2005.

[15] E. G. da Silva, L. F. Pires, and M. van Sinderen, "Towards runtime discovery, selection and composition of semantic services," *Computer Communications*, vol. 34, no. 2, 2011.

[16] acatech, Ed., *Cyber-Physical Systems: Driving Force for Innovation in Mobility, Health, Energy and Production*. Munich, Germany: acatech – National Academy of Science and Engineering, 2011.

[17] J. Wan *et al.*, "Vcmia: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Networks and Applications*, vol. 19, no. 2, 2014.

[18] S. S. Craciunas *et al.*, "Information-acquisition-as-a-service for cyber-physical cloud computing," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10, 2010.