Platforms for industrial cyber-physical systems integration: contradicting requirements as drivers for innovation

Michael Heiss, Andreas Oertl, Monika Sturm
Siemens AG Österreich
Corporate Technology
Vienna, Austria
{firstname.name}@siemens.com

Peter Palensky, Stefan Vielguth, Florian Nadler
AIT Austrian Institute of Technology
Energy Department
Vienna, Austria
{firstname.name}@ait.ac.at

***Abstract*—The full potential of distributed cyber-physical systems (CPS) can only be leveraged if their functions and services can be flexibly integrated. Challenges like communication quality, interoperability, and amounts of data are massive. The design of such integration platforms therefore requires radically new concepts. This paper shows the industrial view, the business perspective on such envisioned platforms. It turns out that there are not only huge technical challenges to overcome but also fundamental dilemmas. Contradicting requirements and conflicting trends force us to re-think the task of interconnecting services of distributed CPS.**

Keywords—cyber-physical systems, IT platforms, software integration, distributed systems, complexity management

I. INTRODUCTION

A variety of businesses are experiencing a boost in innovation due to cyber-physical systems. Consumer electronics, the health sector, the energy sector, or the automotive industry are examples of domains where the design and operations of highly integrated embedded systems are closer to physical processes than ever. New methods for describing, verifying, designing, and operating such integrated and often autonomously acting systems are required [1]. However, the biggest challenge likely lies in between these domains: the integration of distributed, multi-domain cyber-physical systems [2].

The combination and integration of diverse CPSs lead to an entirely new magnitude of applications and design challenges regarding the resulting CPS. An autonomous, electric vehicle may serve as an illustrative example: In order to perform a smooth, safe, comfortable, and efficient shopping tour, various CPSs that were previously operated independently need to interoperate:

- The steering and navigation controls,
- The charging electronics,
- Vehicle-to-infrastructure, vehicle-to-vehicle, and vehicle-to-pedestrian communications,
- The charging station directory,
- The route planner, using traffic including data from social networks,
- List of available parking lots,
- The shopping app on the smartphone with electronic shopping,

and so on. In the sense of Adam Greenfield [3], the reader might argue that this is an overspecified luxury scenario and does not meet the real needs of smart citizens. In addition to those luxury scenarios there are scenarios like the collaboration of building automation systems and smart grids (e.g., buildings offer their energy storage capacity to the grid) which are key enablers for the energy revolution [4].

The key substrates between these independent systems are integration platforms that combine the power of the cloud with distributed, autonomous systems at the edge. Traditional data-intensive businesses like Google and Facebook and traditional infrastructure businesses like Siemens and General Electric are both moving towards one goal: the integration of massively distributed CPSs in order to implement new services.

The key challenges are evident once control applications are implemented via such platforms: real-time requirements, interoperability, scalability, and flexibility are hard to achieve when a large variety of different CPSs are expected to work together.

II. INDUSTRIAL NEEDS

From start-ups, academia, and research institutes to multinational enterprises and governments: the awareness of CPSs is high [5, 6, 7]. Most industries are currently investigating the potential benefits of CPSs for their business: from manufacturing and process automation to energy, building automation, mobility, smart cities, and health care. Examples of well-known CPS use-cases are given in Table I.

In all those use-cases, we are confronted with (1) more interconnected and therefore more complex solutions, (2) the desire of cross-domain operation and optimization, (3) more autonomous behavior, and (4) the need for a better human-in-the-loop integration. Nevertheless, high awareness of business

978-1-4799-7358-3/15/$31.00 ©2015 IEEE

and academia does not automatically lead to commercial success. We identify three main barriers industry faces with CPSs: (1) technological challenges, (2) regulatory frameworks conflicting with the new realities of CPS development and deployment, and (3) high impact on current business models. The limits and boundaries of the current businesses will be shifted. New players will be part of the future business.

TABLE I: EXAMPLES OF CPS USE-CASES IN DIFFERENT INDUSTRIES

Smart City
Resilient city
Mega events
CO_2 reduction
Smart Grid
Energy efficiency
Smart Mobility
Autonomous driving
Smart travel services
Smart Health
Ambient living
Integrated hospital information systems
Urban health
Smart Living / Smart Home
Integrated energy management
Smart Comfort
Smart Industry Automation
Smart manufacturing
Smart logistics and sourcing

The resulting industry challenges have been analyzed by [6, 8] and most recently by [9, 10]. In this paper, we present a new perspective on industrial challenges: the contradicting requirements.

Most CPS challenges mentioned in the literature are individually approached research tasks. Research fields like predictability, validation techniques and security issues are some of the most frequently mentioned [8, 11].

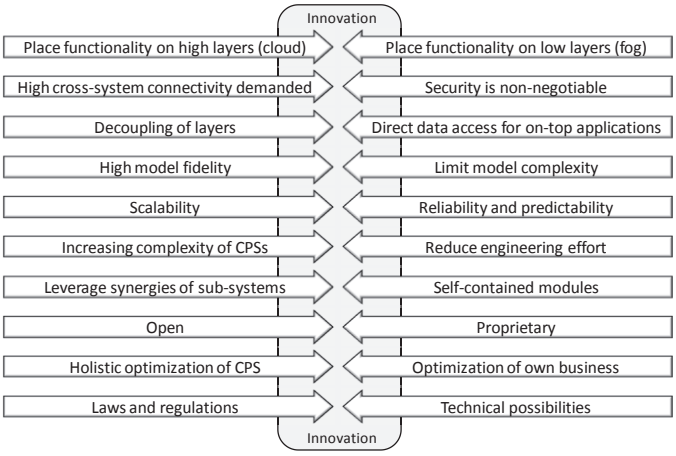

Fig. 1 Contradicting requirements are challenges but also drivers for innovation.

In the following Sections, we focus on requirements which lead to contradictions. If two requirements for the same system are in conflict with each other (Fig. 1), there are three options: give up, change the requirements, or innovate. The first option seems to be no option for CPSs, the second option might be an option for some sub-systems or special domains where some of the requirements are not applicable. The third option, to innovate, is the most interesting and, therefore, the focus of this paper. Conflicting requirements are drivers for innovation.

A. Placement of functionality: cloud vs. fog

CPSs are typically managed and controlled on multiple levels of their system architecture, leading to a nested multi-loop control structure. The highly dynamic control is usually done directly at the field or control level (see Fig. 3) in order to fulfill real-time demands. Cisco introduced the term *fog computing* [12] for computations at the edge, i.e. on lower levels of the system architecture.

Control functionality with the aim of a global optimization (e.g., cross-domain optimization between the smart energy grid and the building energy automation) is typically done at higher levels of the system architecture, e.g., at the cloud.

The following trends can be observed (Fig. 2):

1) Computational power and communication bandwidth is increasing: More data can be transferred to in-time and analyzed at central servers. Therefore, real-time or near-real-time requirements can also be fulfilled on higher architecture levels.

2) Network and field devices are becoming more intelligent: More computational power is available in the field. Therefore, more functionality can be performed on lower architecture levels, e.g., applying the decentralized and distributed divide and conquer principles for big data.

3) Demand for global optimization: The interconnected components of large CPSs are in most cases distributed over more than one site. CPSs connect components from different application domains. This enables the CPS designer to perform overarching optimization of the controlled processes and systems for the whole CPS, e.g., by aggregating data on the dependencies of the participating sub-systems. In classical architectures, this global optimization functionality is assigned to higher architectural levels.

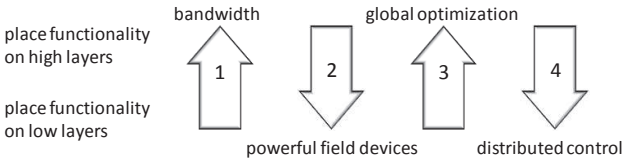

Fig. 2: conflicting requirements for the placement of functionality on higher or lower layers of the architecture. The arrow numbers correspond with the paragraph numbers above.

4) Infrastructure independence: There is a strong demand to be independent of centralized infrastructure. Centralized infrastructure is seen as being costly and makes the system dependent on the failsafe operation of this central infrastructure. If central infrastructure is used, a common requirement

demands that the system at least continues to operate in an autonomous mode if the central infrastructure fails. Examples like the architecture of the Internet itself demonstrate the potential resilience of de-centralized systems. This leads to the demand for more functionality being placed on lower architecture levels.

Currently, there is a revolution taking place in industrial IT. The classical industrial IT, e.g., the Totally Integrated Automation (TIA) pyramid [13], consists of the field level, the control level, the SCADA level, the planning level, and the management level (Fig. 3, left part). This industrial IT is increasingly being accompanied by the Internet IT (Fig. 3, right part).

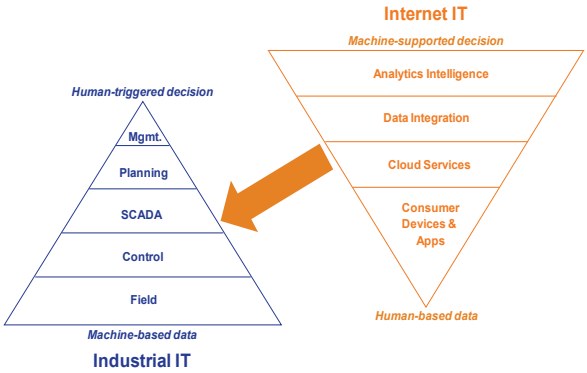

Fig. 3: Industrial IT will be accompanied by Internet IT

The two are growing together and the boundaries between them tend to blur. In future architectures [14], software and hardware will likely be largely decoupled, which might result in a physical structure as shown in Fig. 4: the major functionality will be placed either at the edge, or in the cloud. The platform efforts will then take place in all those layers: an embedded stack at the things-level, a fog-platform at the edge, a cloud platform in the cloud and, a cross-CPS platform for a dynamic orchestration of the ecosystems. This is both disruptive for the business models and a radical innovation of the underlying technologies.

In the following paragraphs, we present barriers which need to be overcome in order to realize this vision.

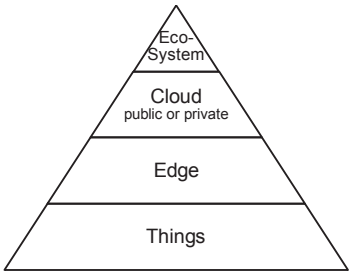

Fig. 4: Digitalization may result in a new structure of the IT and operational technology. All functions may be re-assigned or re-imagined.

The resulting research questions are as follows: Which CPS architecture addresses these demands in an optimal way? Will the future architecture be the one shown in Fig. 4? Will there be any central unit or will future concepts avoid central units? What are the decision criteria to place functionality at the edge or in the cloud? How dynamic will the placement of the functionality be within those layers and across those layers? Does the dynamic placement of functionality only make sense for resilience modes or are there other use-cases of similar relevance? How and how reliably can a global optimization task be performed if there is no central computing? What are the properties of a distributed middleware?

B. Networking and cloud vs. security

A CPS is by definition highly interconnected. From a digital customer experience point of view, the smooth interoperability of the sub-systems, the central monitoring, and the holistic system operation optimization is highly appreciated and demanded. Massive interconnection is required in order to meet these demands. Nevertheless, data security, data privacy, and data trust are non-negotiable requirements in almost all industries. Integration platforms need to address those security issues properly.

The resulting research questions are as follows: How to design the security architecture of a CPS, so that the IT-security metric of the whole system is equal to or better than the metrics of individually operated sub-systems? How are these new IT-security metrics defined? Under which conditions would, e.g., a manufacturing company be willing to have its production data executed in the cloud?

C. Decoupling of layers vs. direct data access for on-top applications

One of the main achievements of the last decade is the high reliability and predictability of industrial IT. A significant contributor to this achievement is the decoupling of the architectural layers according to the pyramid in Fig. 3. A higher layer has no direct access to the data and commands of a lower layer, but only hands over demands via well-defined interfaces. The rights management of each layer defines who is allowed to get access via which interface.

This well-defined, layered access makes gaining access to required resources impossible, if a lower layer does not permit the access. The reason could be that the system architect did not envision the need for this permission for on-top applications that were unknown at the time of the system design.

The broad accessibility and interconnection of sub-systems but also the interoperability of third-party on-top applications enables the development of applications which have never been envisioned at the time of the system design. Compare how the Google Maps app had a significant impact on the app ecosystem of our smartphones.

There is a big difference between the smartphone and a large cross-domain CPS: the number of resources. The smartphone mostly has everything once: one location, one identity, usually one calendar, one Wi-Fi connection, etc. A large CPS system may have thousands of sensors and actuators. Let us assume that the on-top application wants to have access to a particular temperature sensor in a particular room. How does it get the unique name of this room? If there is more than one temperature sensor in this room, how is the one most appropriate for the application selected? The demand is that access to the resources

(including data) is protected and available at the same time [10]. Currently, there are some concepts and technologies well known in the IT field – namely ontologies and the related techniques such as RDF, OWL, and SPARQL – that might pave the way in terms of implementation and address the issues of resource identification and access as well.

The resulting research questions are as follows: semantic search and ranking mechanisms for the selection of the right resources; advanced configuration technologies [15] considering not just the access permissions, but also the rules and constraints within a large CPS, e.g., which control command may, under certain system conditions, not be overwritten by a higher level command due to safety reasons? What is the most effective access rights architecture for such large multi-layered CPS systems? How should multi-tenancy concepts be designed?

D. High model fidelity vs. complexity of models

The classical challenge of modeling is to choose the appropriate model fidelity. Which properties of a sub-system are relevant for the particular purpose of the model? What is relevant depends very much on the context in which the model is applied. In other words, the goal is to design models with the least possible complexity, just enough to fulfill the purpose.

This design criterion changes if the purpose of the model is not known at the time of model design. A *digital twin* [16] should behave like the system itself even in critical or unusual situations. This leads to more and more complex models, which are interconnected with other models, (1) on different levels of granularity, (2) in different modeling paradigms, and (3) for different components of the system [17].

Ideally, the application should select the model quality, while a platform offers several types of accuracy and granularity (e.g., dynamic, quasi-static, and static models for energy consumption). An important factor in this context is error estimation. Any executed model should provide an upper bound of its errors, and the associated process (e.g., an optimization run or a forecast of events) can migrate between different model grades depending on its accuracy needs.

A further challenge is the determinism of models [18]. Programming languages like C or Java have no mechanism to express timing behavior. Even languages like PEARL that include timing constructs rely on the capability of the underlying computers to provide mechanisms for controlling timing in their instruction set architectures [18]. Lee has demonstrated based on PRET and Ptides [18] that deterministic CPS models with faithful physical realization are possible and practical.

The resulting research questions are as follows: How can we set up multi-modal models in a way that ensures their complexity is manageable?

E. Scalability vs. reliability

During the life cycle of a large CPS, the system will change, due to either a growing or shrinking number of nodes – i.e., a node is assumed to be a participating or managed complex physical system to be incorporated [14] – or because the addition of new or the removal of old components, sub-systems, or systems to the CPS.

Customers expect the reliability and predictability of the CPS to remain untouched.

The resulting research questions are as follows: How can we predict the system behavior and reliability of the scaled system via modeling before we actually scale it? Are there architecture and configuration concepts which guarantee reliable scalability within certain limits?

F. Increased complexity vs. reduction of engineering effort

This is one of the key challenges. The higher interconnection of sub-systems in CPSs leads to a higher complexity of the CPS [5]. High interconnection is required to fulfill the customer requirements. Higher complexity classically leads to increased engineering effort for such systems. This is one of the underlying principles of the function point analysis for effort estimation [19].

However, systems requiring less engineering effort typically have a lower total cost of ownership and are more reliable. Both qualities are of high interest for the customer.

This leads us to the desire to handle the complexity automatically. The idea is to develop a platform in such a way that the platform has well-tested *self-** mechanisms. These include self-organization, self-configuration, self-learning, self-healing, self-optimization, self-protection, and self-explaining. Those *self-** mechanisms reduce the engineering effort, ideally to zero, e.g., with a plug-and-play (PnP) mechanism. The demand is to make the handling of a CPS as simple as possible and to enable customers to act without IT experts.

Most of today's known PnP mechanisms solve complicated but not complex problems for us. The connection of a memory-stick to a computer via USB is an example for such a task. Without the standardized USB protocols, this would be a complicated, but not a complex task, as the solution is straightforward if the technical specifications on both sides are known. Computer programs do well supporting complicated tasks like the finite elements simulation for the simulation of physical systems, but they are no silver bullet for solving complex tasks. Lacking a consistent system theory or practicable complexity metric [20], practitioners postulate that complexity can only be handled by a system of similar complexity. Otherwise, it is not complex but just complicated. Even the well-known so-called complexity metrics in software engineering from McCabe [21] and Halstead [22] do not sufficiently consider the complexity generated by interconnections.

We do not believe in the existence of a generic, "one-size-fits-all" self-engineering mechanism which dissolves all system complexity. Nevertheless, we are convinced that engineering tools for specific application domains will significantly reduce the engineering effort. Additionally, new basic concepts will be discovered that will boost the whole community. The process of development, modeling, and operation will be tightly coupled and overlapping in terms of timeframe in the case of CPSs.

Research questions related to system theory: Is it possible to derive a no-free-lunch theorem for managing complexity like the no-free-lunch theorem for optimization problems [23]? Are there systematic approaches for CPSs to separate complications from complexity? Is there a formal abstraction model for describing and measuring complexity in the above-mentioned sense? Is there something like a difference between troublesome complexity and irrelevant complexity? Can we then shift complexity from a troublesome area into an irrelevant area?

More pragmatic research questions: robust concepts for *self-** mechanisms in selected application domains.

We know from the field of swarm intelligence [24, 25] that very simple individual devices, in a high number and highly interconnected, can lead to very complex system behavior. Now we need it the other way around: The CPS is already a very complex system. It seems to be impossible to manage its complexity with a single central unit. Can we implement in each node a simple add-on that manages the CPS complexity in a distributed way?

G. *Leverage synergies vs. modularity of self-contained modules*

In conventional systems, each sub-domain has its own self-contained system. The evolution of those systems and the demand for interconnecting those systems led to a situation where much functionality is implemented redundantly in different sub-systems. Due to the different use-cases and requirements in the different sub-domains, the redundant functionality of the legacy systems is often not directly compatible and difficult to synchronize. Typical examples of such multiply implemented functionalities are: user management, asset management, authentication/registration mechanism, databases, event handling, and resilience mechanisms.

Nevertheless, customers like modular concepts. Modularity makes them independent from single suppliers and gives them more flexibility to adapt their CPS to their needs.

In a Greenfield situation, the architecture of such a CPS would make use of the well-established architectural patterns for system of systems integration [26] and therefore leverage the synergies between the different sub-domains. When legacy systems need to be handled, the applicable patterns for Brownfield or Closed Source scenarios [26] will have less impact on leveraging the synergies. A complete re-design or a replacement of selected sub-systems might be necessary for harvesting the full synergy potential.

From a daily business perspective, the migration from the heterogeneous landscape of legacy systems to a future CPS is one of the most important challenges.

The resulting research questions are as follows: Which migration strategies have the potential to become best practices? How can the modularity – e.g., the easy replacement of sub-systems – be kept and at the same time the synergies between sub-systems be optimized? How can resilience mechanisms work across sub-systems? How can we deal with the insight that standardization will be the answer to most of those questions, but that those standards are not sufficiently established today?

H. *Standardized open interfaces vs. proprietary reliable systems*

Standardization and open interfaces are key success factors for the integration of heterogeneous sub-systems into large multi-domain CPSs [27]. Nevertheless, we observe some inertia of successful companies to open the interfaces of their currently proprietary systems.

Open interfaces stimulate competition. Customers like the competition of their suppliers, which typically leads to a reduction of costs. On the other hand, customers need the reliability of their CPS and, in the event of troubles, they prefer to have one responsible contact person. In highly interconnected complex systems, it is difficult to prove, or sometimes even impossible to prove, the root cause of a failure. Therefore, the legal situation for a proprietary system is clearer than for an orchestrated assembly of independent open systems.

Standardization and openness lead to a separation of hardware and software. Today, a company can still gain higher margins when offering intelligent hardware (e.g., an intelligent sensor), where hardware and software is offered as a unique package that adds value for the customer and cannot be bought anywhere else. If the hardware and software need to follow open standards, then hardware and software can be separated; in other words: the software is replaceable by the software from other providers. These other providers can be small startups or established IT providers. They argue that their service is better integrated into the CPS-IT as they might be the provider of other significant parts of the CPS-IT.

In recent years, we have observed a shift in the perception of IT security aspects of open source platforms. Today, well-established open source platforms are perceived to be more secure than proprietary products because of the full transparency of open source platforms and the large open source community [28]. This community has the ability to detect new security issues much faster and to fix those problems faster than in proprietary products.

The resulting research questions are as follows: Open standards and reliability are not necessarily contradictions: How should a framework be designed, where all components of a CPS are delivered with a digital twin [16] that enables the predictability of the CPS's system behavior and supports the detection of unexpected behavior and the root cause analysis in case of a failure? How could a root cause analysis in a CPS be standardized to a certain degree?

I. *Holistic optimization vs. optimization of own business*

The optimized operation of a CPS is not just a technical challenge – as discussed in the previous paragraphs – but also an organizational challenge for both the solution provider and the customer. The main reason is that a cross-domain optimization (and sometimes also optimization across sub-domains) has an impact on the business model.

A good example is a smart city: Today, the thermal energy provider, the gas provider, the electricity provider, the lighting company, the building operators, the car-sharing companies,

the bicycle sharing companies, the taxi companies, the parking space operator, the local public transportation company, and the railway companies all separately optimize their businesses. More progressive companies have agreements with their relevant partners, but today there are just a few living lab scenarios [3] around the globe where those businesses are digitally connected to a large CPS and are optimized in a holistic way.

Imagine a solution provider who has developed such an overarching CPS. Who is the customer for this solution? The mayor of the city is highly interested in such a solution, but he or she is not the business owner. Neither is each of the separate companies a customer of the overarching CPS. All those separate companies could start a joint venture and then this joint venture could be a customer. But why should they do it? In the best case, the CPS is capable of saving resources. The end customers would like it as this would save them money. In other words: the CPS reduces the revenue of those separate companies, or even worse, it could shift the revenue from one company to another.

On the solution provider side, the situation is similar: There are a lot of separate solution providers which offer the "perfectly optimized" IT solutions for each of the above-mentioned companies, respectively. Only the very large solution providers would have the in-house capability to build an overarching CPS (assuming that they manage to solve their internal cross-divisional cannibalism). As long as they do not see a customer for such an overarching smart city CPS, their activities remain research projects.

Therefore, there is now an urgent demand to invent new business models that have the power to mix up the established businesses. The established businesses are not the main drivers, as they have no guarantee that they will win in this new situation. The drivers are startups that have nothing to lose, but if they create a new successful business model, they can win a lot. Well-known patterns for such business models [29] are pay-per-use or freemium models. The challenge lies in identifying those value-adding services that motivate customers to pay for them.

Those new business models will more likely evolve if the underlying CPS platforms have open interfaces and the capability for third-party applications to access all allowed resources (like we know it from the smartphone app business). The vision is to connect businesses like people are connected via social media today [10].

The resulting research questions are as follows: How to identify new value adding services and develop the corresponding business models? What kind of cross-sub-domain or cross-domain services will be the new "killer applications" that stimulate the CPS market? How can a CPS be monetized? What is the difference between business models for the consumer-oriented smartphone app market and the B2B-oriented CPS application market, and which new business models will be appropriate for the latter case? How can we connect businesses as easily as people are connected via social media today? How can consumer apps stimulate the B2B CPS application market? What would a roll-out strategy for the overarching CPS look like, and how would it differ for large companies and startups? What are the main motivations for companies or individuals to contribute to a holistic optimization instead of optimizing their own business – is it only the fear of losing market share if others do it?

J. Regulated environment vs. technical possibilities

CPSs are often included in safety-critical systems such as traffic control, trains, rail automation, airplanes, and health care solutions. Safety-critical applications are strictly regulated. Standards like SIL or FDA regulations have strict regulations for the development process and the commissioning of the CPS solution, e.g., every change of the software requires a new (typically external) certification by the respective authority.

Those regulated environments are in conflict with the demand of the dynamic tailoring of the CPS. The CPS should be able to add new components or remove old components or to install third-party applications during the live operation of the system. The current regulations for safety-critical systems prohibit such live changes of the system properties [30, 31].

To sum it up, much more is technically feasible than it is permitted by law. Another example is the very strict data privacy regulations within the European Union that prohibit certain analyses of consumer behavior, even if it is intended solely for the benefit of the consumer [32].

The resulting research questions are as follows: How can a safe continuous certification process [33] be defined which is acceptable for the regulatory bodies? How can the country-dependent legal requirements be managed in a global CPS application?

III. PRACTICIONERS' NEEDS

After the analysis in Section II, we want to compare those results with the needs of engineers who are currently developing cyber-physical systems. To this end, we asked the following simple open question on the Siemens-internal knowledge sharing platform TechnoWeb [34]: "What are your three main challenges for CPS integration – your daily pain points?" 25 managers and engineers around the globe contributed by relating their own experiences.

Answers were posted in an open forum, so mentioning the same problems multiple times was indirectly discouraged. Therefore, the frequency and severity of the pain points cannot be measured by counting how often they were mentioned. The advantage of the applied method is that it encourages respondents to add a new aspect which was not already mentioned by other engineers before.

That said, problem variants concerning the issues of (1) security, (2) communication between heterogeneous systems, and (3) the lack of standardization were broached fifteen separate times, making a strong argument that these topics are universal pain points.

When viewed as a whole, the practitioners' needs touch all major challenges of Section II and the CPS literature [e.g., 7].

In Table II, we present problems not commonly found in CPS literature and shed some light on everyday problems. To summarize, practitioners know that the technological problems will be solved (and are looking forward to contributing with

their own expertise) but are very aware of the risks of having too high or naive expectations and know about the challenges of transforming their own organization into a *CPS-ready* organization.

TABLE II: PRACTITIONERS' PAIN POINTS (SINGLE NOMINATIONS – FOR MULTIPLE NOMINATIONS SEE TEXT)

Knowledge protection of intellectual property rights that are included in digital twins
Managing the life cycle of complex CPSs
Lack of a business model and owner
Changing the mindset to facilitate sharing and collaboration between previously unrelated departments
Meeting customer expectations with a CPS portfolio that offers high prices for high performance and moderate prices for moderate performance.
Risk of diversifying products too much, making them untestable and less robust.
Risk of developing inflexible and un-scalable solutions
Social impact of ubiquitous technology
Experimentation and early prototyping hindered by security policies
Increased need for electrical energy
Training of the workforce to develop a security culture and awareness necessary for CPSs
Ability to "hotfix" and apply incremental updates
Performance guarantees on heterogeneous hardware

IV. CONCLUSION

The paper has shown that the task of integrating diverse CPSs with some universal and flexible platform spans a large field of requirements. Technical requirements like handling large amounts of data or keeping services and information secure and protected are considered as standard challenges.

The much more interesting aspect is the list of decision dilemmas when integrating CPSs. The standard way of overcoming such dilemmas is deciding for one side, e.g., reliability is more important in a particular case than scalability. This, however, would sacrifice all that we expect from such platforms: openness, preparedness for unknown future applications, flexibility, and sustainability.

It is therefore necessary to come up with innovative, radically new approaches that do not suffer from these dilemmas. For instance, traditional security concepts cannot cater to cloud-based systems. Something new needs to be applied, where data and services can safely migrate in potentially unsafe environments. Similarly, the other dilemma of standards vs. proprietary is not new, but in the context of finding a solution for CPS integration platforms we need a radically new solution for it: a solution that not only works with software design methods but takes the entire life cycle of such systems into consideration, including all of their legal, financial, and ownership issues.

Integrating distributed cyber-physical systems is a promising topic. For example, smart cities with optimized energy systems that interact with transport and other domains are expected to make our lives safer, cleaner, and more comfortable. However, there are fundamental obstacles ahead of us that require radical innovation and new solutions.

V. ACKNOWLEDGEMENTS

The authors like to thank Bernd Rosauer for his visionary leadership as well as the anonymous reviewers and the many colleagues who contributed with their valuable comments and inputs, in particular Martin Lehofer and Herwig Schreiner.

References

[1] E. A. Lee, "Cyber physical systems: Design challenges" Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on. IEEE, 2008.

[2] J. Sztipanovits et al., "Toward a science of cyber–physical system integration" *Proceedings of the IEEE* 100.1 (2012): 29-44.

[3] A. Greenfield, A, "Against the smart city (The city is here for you to use)". New York City, Do Projects, 2013.

[4] J. Rifkin, "The Third Industrial Revolution: How Lateral Power Is Transforming Energy, the Economy, and the World", Palgrave Macmillan Trade, ISBN-13: 978-0230341975, New York 2013.

[5] "Cyber-Physical Systems: Situation Analysis of Current Trends, Technologies and Challenges." In Foundations for Innovation in Cyber-Physical Systems (NIST CPS-Workshop), March 2012. Prepared by Energetics incorporated for the National Institute of Standards and Technology (NIST).

[6] E. Geisberger and M. Broy, "*acatech STUDIE",* Springer, 2012.

[7] M. Törngren, "Cyber-Physical European Roadmap & Strategy", "Cyber-Physical European Roadmap & Strategy D5.1", 2014

[8] S. Ying, "Foundations for Innovation in Cyber-Physical Systems." *Workshop Report, Energetics Incorporated, Columbia, Maryland, US*. 2013.

[9] V. Gunes et al., "A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems", *KSII Transactions on Internet and Information Systems*, vol. 8, no. 12, Dec. 2014.

[10] P. Bellavista et al., "Cyber Physical Systems: Opportunities and Challenges for Software, Services, Cloud and Data," Networked European Software and Services Initiative (NESSI) Whitepaper, 2015

[11] E. Geisberger et al., "agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems. " Springer, Berlin, 2012.

[12] F. Bonomi et al. "Fog computing and its role in the internet of things." Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012.

[13] Totally Integrated Automation, www.siemens.com/tia [Accessed: 12-Feb-2015].

[14] A. Giordano et al., "Rainbow: an Intelligent Platform for Large-Scale Networked CPS." [Online]. Available: http://ceur-ws.org/Vol-1156/paper6.pdf. [Accessed: 20-Jan-2015].

[15] A. Falkner, A. Haselböck, G. Schenner, H. Schreiner, "Modeling and solving technical product configuration problems", AI EDAM 2011 25(2): 115-119.

[16] E. H. Glaessgen and D. Stargel. "The Digital Twin paradigm for future NASA and US Air Force vehicles." *53rd Struct. Dyn. Mater. Conf. Special Session: Digital Twin, Honolulu, HI, US*. 2012.

[17] P. Derler et al., "Modeling Cyber-Physical Systems". *Proceedings of the IEEE (special issue on CPS)*, 100(1):13-28, January 2012.

[18] E. A. Lee, "The Past, Present, and Future of Cyber-Physical Systems: A Focus on Models", Sensors, 15(3), p. 4837-4869, doi:10.3390/s150304837, February, 2015.

[19] M. Kjetil and M. Jorgensen. "A review of software surveys on software effort estimation". *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*. IEEE, 2003.

[20] F. Heylighen, "Complexity and Self-Organization." Encyclopedia of Library and Information Sciences. CRC. ISBN 978-0-8493-9712, 2008

[21] T. J. McCabe, "A Complexity Measure". IEEE Transactions on Software Engineering: 308–320, 1976.

[22] M. Halstead, "Elements of Software Science", Amsterdam: Elsevier North-Holland, Inc. ISBN 0-444-00205-7, 1977.

[23] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization." Evolutionary Computation, IEEE Transactions on 1.1: 67-82, 1997.

[24] M. Rubenstein et al., "Programmable self-assembly in a thousand-robot swarm." Science 345.6198 (2014): 795-799.

[25] E. A. Lee et al. "The Swarm at the Edge of the Cloud," IEEE Design & Test, vol. 31, no. 3, pp. 8–20, Jun. 2014.

[26] R. Kazman et al., "Understanding patterns for system of systems integration," in *2013 8th International Conference on System of Systems Engineering (SoSE)*, 2013, pp. 141–146.

[27] Z. Fan et al., "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 21–38, 2013.

[28] J.-H. Hoepman and B. Jacobs, "Software Security Through Open Source," Technical report, Institute for Computing and Information Sciences, Radboud University Nijmegen, 2005.

[29] A. Osterwalder et al. "Business Model Generation: A handbook for visionaries, game changers and challengers." African Journal of Business Management 5.7 (2011).

[30] E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?", Position Paper for NSF Workshop On Cyber-Physical Systems:Research Motivation, Techniques and Roadmap October 16 - 17, 2006

[31] L. I. Millett et al., Software for Dependable Systems:: Sufficient Evidence?. National Academies Press, 2007.

[32] "EUR-Lex - 31995L0046 - EN," Official Journal L 281 , 23/11/1995 P. 0031 - 0050;. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:31995L0046. [Accessed: 12-Feb-2015].

[33] "About Open-DO." [Online]. Available: http://www.open-do.org/about/. [Accessed: 03-Feb-2015]

[34] C. Wiener et al. "Targeting the right crowd for corporate problem solving-a siemens case study with TechnoWeb 2.0." Technology Management Conference (ITMC), 2012 IEEE International. IEEE, 2012