Project Deliverable

Project Number: 611115	**Project Acronym:** CPSoS	**Project Title:** Towards a European Roadmap on Research and Innovation in Engineering and Management of Cyber-Physical Systems of Systems

Instrument: COORDINATION AND SUPPORT ACTION	**Thematic Priority** ICT

Title D2.3 Report on the Second Meeting of Working Group 3

Due Date: Month 13	**Actual Submission Date:** Month 13 (October 2014)

Start date of project: October 1st, 2013	**Duration:** 30 months

Organization name of lead contractor for this deliverable: TUE	**Document version:** V2.1

Dissemination level (Project co-funded by the European Commission within the Seventh Framework Programme)

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group defined by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract :

This document provides the minutes of the second meeting of Working Group 3 (Workshop on Tools and Methods for CPSoS) that took place on September 12th in Bertinoro, Italy, and the breakout session on tools that was organised as part of the second Working Group 2 meeting on October 1st in Zürich, Switzerland.

Authors (organizations):

Michel RENIERS, Wan FOKKINK (TUE)

Reviewers (organizations):

Sebastian ENGELL (TUDO)

Keywords :

Working Group 3, Cyber-physical systems of systems, engineering, tools, research challenges

Disclaimer :

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppels or otherwise, to any intellectual property rights are granted herein. The members of the project CPSoS do not accept any liability for actions or omissions of CPSoS members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice

Revision History

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V1.0	24/10/2014	Creation	Michel RENIERS (TUE), Wan FOKKINK (TUE)
V1.1	25/10/2014	Initial Review	Sebastian ENGELL (TUDO)
V2.0	27/10/2014	Update	Michel RENIERS (TUE)
V2.1	28/10/2014	Review	Sebastian ENGELL (TUDO)

Table of Contents

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

Acronyms and Definitions

Acronym	Defined as
CPSoS	Cyber-physical Systems of Systems
CPS	Cyber-physical Systems
SoS	System of Systems
STReP	Specific Targeted Research Projects
DYMASOS	Dynamic Management of Physically Coupled Systems of Systems
iFM	Integrated Formal Methods
FACS	Formal Aspects of Component Software
COMPASS	Optimized Co-modal Passenger Transport for Reducing Carbon Emissions
HYCON2	Highly-Complex and Networked Control Systems
DSS	Decision Support System

1. Executive Summary

CPSoS, funded by the EC (FP7 programme), is a 30-month Support Action that provides a forum and an exchange platform for systems-of-systems related communities and ongoing projects, focusing on the challenges posed by the engineering and the operation of technical systems in which computing and communication systems interact with large complex physical systems. Its approach is simultaneously integrative, aiming at bringing together knowledge from different communities, and applications-driven.

The findings of the project will be summarized in a concise strategic policy document "European research and innovation agenda on Cyber-physical Systems of Systems" supported by a set of in-depth technical papers, presented at a symposium "Cyber-physical Systems of Systems Meeting Societal Challenges".

The second meeting of the Working Group 3 on "Methods and Tools for Engineering and Management of Cyber-physical Systems of Systems" was held as a public workshop "Workshop on Tools and Methods for Cyber-physical Systems of Systems" on September 12th, 2014, at the University Residential Center, Bertinoro, Italy. The programme consisted of invited presentations from Working Group members and submitted presentations.

The second part of the programme consisted of a presentation of key research challenges derived from a study into the state-of-the-art of methods and tools for engineering and management of CPSoS, and a discussion with the participants thereof.

Outcome:
The discussion confirmed that the current state-of-the-art description is valid mostly. Relevant suggestions for improvement were given. Furthermore, in the discussion of the identified key research challenges these were confirmed as key challenges and some of these were identified as having a high priority.

As a part of the second meeting of Working Group 2 (Physically Connected Systems of Systems) which was organized as a public event jointly with STReP DYMASOS (Dynamic Management of Physically Coupled Systems of Systems) in Zürich on October 1st, 2014, a breakout session on tool support was organised. The goal of this session was to validate and prioritize key research challenges. For the breakout session on tools, the results of this prioritization are reported in this deliverable.

Outcome:
The prioritized list of key research challenges for different time horizons will be used in discussions with tool experts and will be used for updating the state-of-the-art description and as input for the European research and innovation agenda.

2. Workgroup Meeting and Public Workshop

The second meeting of Working Group 3 took place in the form of a "Workshop on Tools and Methods for CPSoS" that was co-located with the scientific conferences iFM 2014 (The 11th International Conference on Integrated Formal Methods) and FACS 2014 (The 11th International Symposium on Formal Aspects of Component Software). In total 15 participants gathered for the workshop on September 12th in Bertinoro, Italy to discuss the state-of-the-art and future challenges in the engineering of cyber-physical systems of systems.

The programme of the workshop consisted of three types of presentations: invited presentations, presentations by Working Group members from the CPSoS project, and submitted presentations. The submitted presentations were solicited by means of a Call for Presentations. Proposals for presentation consisted of a paper of at most four pages that clearly shows the relationship to the topic of the workshop and gives an overview of the material to be presented at the workshop. Five such papers were submitted. See Appendix 4 for abstracts of those.

Submitted presentation proposals have been evaluated by the organisers of the Workshop (John FITZGERALD, Wan FOKKINK, and Michel RENIERS) and were accepted for presentation at the workshop. The programme of the workshop is provided in Appendix 4.

The last part of the programme (16:00 – 17:30) consisted of a presentation of the "State-of-the-art and future challenges in tools and methods for engineering and management of cyber-physical systems of systems" and discussion thereof. The literature review that forms the basis for this presentation is included in Appendix 4 and the presentation that was used is included as Appendix 4 as well. The following key challenges were presented

- **(Efficient) Modelling and simulation of large-scale heterogeneous complex systems**
  - formulation of detailed models of the constituent systems, incl. human operators and environment
  - availability of simulation engines capable of dealing with the scale dimension of CPSoS. This requires clear interfaces between heterogeneous models and abstraction methods
  - system-wide simulation techniques that allow to assess the properties of the system prior to effectuating of evolution steps (in case these can be controlled). These should also aid in detecting emerging behavior
- **Abstraction and approximation methods** for reducing model complexity for system-wide functionality and performance analysis
- **Development of control strategies and methods for decision making**
  - that deal well with reconfiguration and partial autonomy of parts of the CPSoS
  - for which reconfiguration and evolution have less impact on the system-wide behavior
  - including methods to detect significant evolutions in the CPSoS in order to react timely with adapted control
- **Techniques for modelling and analyzing threats** to system functionality and performance induced by communication infrastructure
- **Model-based systems engineering approach** is needed that does full justice to the shift from design-time to run-time engineering
- **Tools for the management of models** and relationships between models that allow to keep track of past, current and planned system configurations at the architectural level and provide linkage with the associated models.

The discussion on the state-of-the-art was moderated by Michel RENIERS. Minutes were made by Christian SONNTAG and Wan FOKKINK. These were used as the basis for the following summary of the discussion.

In the discussion of the state-of-the-art the participants indicated that the following issues are to be considered very important:

- Co-simulation is a very important technology for the future. In this context, definition of standardized interfaces for co-simulation that are not linked with or restricted by commercial providers is needed. Another issue that requires attention in the area of co-simulation is the preservation of physical balances (e.g., mass and energy balance) over the boundaries of the individual simulation tools during co-simulation.
- Better algorithms are needed for solving optimization problems. Although solutions for optimization problem formulations are already very general, for real-time optimization, and in particular for such large systems as CPSoS, their application is infeasible.
- More integrated and easier-to-use tools for full life-cycle management is very important.
- In handling the evolutionary aspects of systems of systems there are two issues that need to be separated and that both deserve attention. One is "How do deal with the fact that future evolutions of the system are not known at all?" and the other is "How to decide NOW what actions to apply to a system (e.g., in economic optimization) based on expectations of the future?"
- Uncertainty is an important future challenge in hierarchical and distributed control.
- Methods and models for architectural reconfiguration representation and execution are missing. Consensus-building between many different agents is an important topic in dynamic reconfiguration.

During the discussions the following remarks were also collected that may be used for improving the state-of-the-art document:

- Contract-based design actually helps if there is a lack of trust between agents. Agents do not need to publish their internal designs in this approach.
- Several examples of positive emergence have been mentioned and links to papers that may be useful have been made available.
- The COMPASS project has started developing models to represent and analyze cyber-security issues.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

3. Breakout session on Tools from WG2 Meeting in Zürich

As part of the second Working Group 2 meeting parallel breakout sessions were organised to discuss and prioritise research and development challenges. One of the sessions focused on tool support. In this section this breakout session will be reported on. Further details on this Working Group meeting may be obtained from Deliverable D2.2.

The breakout session as attended by 8 participants (1 from industry and 7 from academia). The goal of the session was to validate and prioritize the key challenges in four areas that are seen as crucial for the future research and innovation in CPSoS. These areas are

- Modelling , Simulation, and Model Management
- Engineering and Run-time Platforms
- Model- and Data-based Engineering Tools
- Integration and Deployment of Advanced Solutions

These areas were introduced by means of a short presentation (See Appendix 4). The following list of key research topics was introduced and discussed:

Modelling, Simulation, and Model Management

1. Keeping all models up to date and consistent (model management)
2. Reducing the effort and cost of modeling by model re-use (object-oriented or modular modelling) and predefined and adaptable standard models.
3. Coupling of many different simulation tools of different strengths (co-simulation)
4. Dynamic on-the-fly reconfiguration of simulation models
5. Integrated modeling and simulation with distributed management schemes, failures, and abnormal states
6. Large-scale, faithful, efficient simulation algorithms for CPSoS with different time scales and on-the-fly reconfiguration

Engineering and Run-time Platforms

7. Development of new engineering frameworks that support the requirements specification, adaptation, evolution, and maintenance of CPSoS not only during design, but over their complete life-cycle
8. Collaborative engineering and run-time environments that enable providers to jointly work on aspects of the CPSoS while competing on others
9. Engineering platforms that support an integrated cross-layer design of fault-resilient management architectures, and early testing facilities to detect errors as soon as possible

Model- and Data-based Engineering Tools

10. New (distributed/hierarchical/decentralized) methods and tools that take CPSoS properties (autonomy, dynamic reconfiguration, …) into account
11. More powerful optimization algorithms and tools that enable real-time optimization of large-scale CPSoS
12. New algorithms and tools for stochastic optimization and risk management

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

13. New algorithms and tools for large-scale CPSoS validation and verification, including reconfiguration (e.g. hybrid simulation/verification approaches, assume-guarantee / contract-based reasoning, …)
14. (Real-time) processing, synchronization, and management of large data sets for monitoring, optimization, fault detection, …

Integration and Deployment of Advanced Solutions

15. Consistently integrating engineering and operational data and engineering artefacts from heterogeneous, structured and unstructured data sources with advanced solutions
16. Integrating new engineering and operational software and hardware tools with existing infrastructure

Part of the discussion was centered around clarification of the contents of the presented list of key research topics.

- The challenges are formulated from the point of view of industry.
- Modelling in practice is not only restricted to modelling of behavior. Nevertheless, in the key research challenges modelling mostly refers to modelling of the dynamic behavior of the system.
- Automatic tearing and algebraic loop breaking should also be considered under challenge 6 (Large-scale, faithful, efficient simulation algorithms)
- Combine challenges 4 and the second part of challenge 5 (Integrated modelling and simulation of failures and abnormal states) from the above list into a new challenge named “Dealing with Unmodelled Aspects”. The first part of challenge 5 (Integrated modelling and simulation with distributed management schemes) is considered part of challenge 1 (Model management).
- Enlarge the scope of challenge 6 (Large-scale, faithful, efficient simulation algorithms) to include model execution as a generalization of model simulation.
- Challenge 9 (Integrated Cross-layer Design) also includes computing hardware problems.
- A challenge Conceptual Model Alignment is added as an important key research area.

An anonymous voting was conducted at the end of the session. Each participant prioritized the research topics in three different time horizons:

- Short-term horizon: less than 4 years
- Medium-term horizon: 4 to 8 years
- Long-term horizon: more than 8 years

Each participant could use two votes for each time horizon. The result of the voting is summarized as follows (see the Appendix for a picture of the result).

Short-term horizon

Industry	votes	Academia	votes
Model Re-use and Predefined and Adaptable Standard Models	1	Model Management	5

This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

Large-scale, Faithful, Efficient Simulation Algorithms	1	Co-simulation	2
		Large-scale, Faithful, Efficient Simulation Algorithms	2
		Consistent Integration of Data and Engineering Artefacts	2
		Model Re-use and Predefined and Adaptable Standard Models	1
		New CPSoS Engineering Frameworks	1
		Collaborative Engineering and Run-time environments	1

Medium-term horizon

Industry	**votes**	**Academia**	**votes**
New CPSoS Engineering Frameworks	1	More Powerful Optimization Tools	4
More Powerful Optimization Tools	1	New System-wide Management and Coordination Tools	3
		Collaborative Engineering and Run-time environments	2
		Model Management	1
		Co-simulation	1
		Large-scale, Faithful, Efficient Simulation Algorithms	1
		Stochastic Optimization and Risk Management	1
		Consistent Integration of Data and Engineering Artefacts	1

Long-term horizon

Industry	votes	Academia	votes
New System-wide Management and Coordination Tools	1	Dealing with Unmodelled Aspects	3
Stochastic Optimization and Risk Management	1	Large-scale Data-based Methods	3
		Stochastic Optimization and Risk Management	2
		Large-scale, Faithful, Efficient Simulation Algorithms	1
		Conceptual Model Alignment	1
		New CPSoS Engineering Frameworks	1
		Collaborative Engineering and Run-time environments	1
		New System-wide Management and Coordination Tools	1
		Validation and Verification of Large-scale CPSoS	1

4. Appendices

4.1 Programme of the Workshop on Tools and Methods for CPSoS

9:30	-	10.30	A process calculus framework for dynamic component structures with sharing Jean-Bernard Stefani
11:00	-	11:30	Towards a security model for cyber physical systems V. Sassone
11:30	-	12:00	Hierarchical control of large complex plants S. Cristea, R. Mazaeda, C. de Prada
12:00	-	12:30	Optimisation methods for recoverable smart electrical grids M. Kamali, M. Kolehmainen, M. Neovius, L. Petre, M. Rönkkö, P. Sandvik
13:45	-	14:30	A formal approach to the design and operation of complex systems A. Cimatti
14:30	-	15:00	A decision support system approach for systems of systems management M.P. Fanti, M. Clemente, W. Ukovich
15:00	-	15:30	A vision for future model-based support for dependable cyber-physical systems of systems C. Ingram, K. Pierce
16:00	-	16:30	State-of-the –art in tools and methods for engineering of CPSoS M. Reniers and W.J. Fokkink
16:30	-	17:30	Discussion on state-of-the-art in tools and methods for engineering of CPSoS

4.2 Abstracts of Submitted Papers

The submitted papers were allowed to be 4 pages each. Below abstracts of these are provided.

4.2.1 Towards a Security Model for Cyber Physical Systems

Towards a Security Model for Cyber Physical Systems (Vladimiro Sassone)

Abstract

In the past few months, we have been working on a computational model of Cyber-Physical Systems (CPS) –and more generally in the Internet of Things – conceived with the intention of developing a framework for the analysis of cyber security. This is based on identifying the key components of a generic object as consisting of: a processing module; a sensing/actuation module; a communications module; and an energy module. The approach is mathematical in its reliance on formal definitions and its attempt to prove security properties conclusively. Our ambition is to build a universal model able to capture all the relevant aspects in a modular, flexible, computational framework. Ultimately, we aim at developing tools to validate the security of (key aspects of) large-scale critical national infrastructures, such as the UK smart metering system.

The talk will report our first steps towards these objectives, and illustrate with selected examples the potentiality of the approach. We shall discuss several challenging issues currently under development, such as the needs to rely on a robust stochastic engine, to formulate a comprehensive and realistic attacker model and, to develop a risk model for the generic CPS component. Our vision is that this will ultimately allow us to deploy realistic quantitative analyses which take into account specific risks associated with specific devices.

This work being still preliminary, our immediate objective is none other than to trigger the interest of the CPSoS community to the cyber security issues of CPS, and hopefully start a dialogue about how to best develop models, techniques and tools so as to inform our future work on this topic.

4.2.2 Hierarchical Control of Large Complex Plants

Hierarchical control of large complex plants (S. Cristea, R. Mazaeda, C. de Prada)

Abstract

Modern process plants exhibit many characteristics of cyber-physical systems of systems: They are composed of different interconnected sections where complex material processing takes place, each one devoted to a specific task and having its own control room, or area within it, with operators performing decisions through a distributed control system that gather information from the process units and, after computing the corresponding control actions, acts on them.

Daily management of these plants is not easy, with decisions organized in different layers, following what is known as the control pyramid. Basic or advanced control of the process units is located in the bottom and can be considered in general as a mature field. Nevertheless, none of these units functions in isolation, so that an efficient operation of the whole plant requires coordinated information processing and the proper global actions as a key factor for the productivity and smooth behaviour of the factory. Many aspects can be considered at this level: bottleneck avoidance with respect to the product flows, efficient energy use among the plants, environmental fulfilments of the regulations, product qualities satisfaction, etc. that require considering the plant

as a whole system where complex decisions involving topics of different nature must be taken and implemented finally through the control system.

A variety of model based real-time optimization methods have been proposed to cope with these problems. Nevertheless, when the process plant involves not only continuous but also batch sections, the problem is much more challenging and not many practical solutions have been suggested in the literature.

This contribution to the Workshop presents an approach for dealing with the optimal operation of processes with mixed continuous and batch dynamics and sharing common resources at the same time. The aim is to generate an architecture that combines local advanced control with a coordination layer able to cope with the global operation goals and easy to be extended to a large number of components from a computational point of view.

The paper takes as reference system a benchmark proposed as case study in the EU, Network of Excellence HYCON2: the crystallization section of a sugar factory, where sugar crystals are made in vacuum pans operating in semi-batch mode and sharing syrup and heating steam.

4.2.3 Optimisation Methods for Recoverable Smart Electrical Grids

Optimisation methods for recoverable smart electrical grids (M. Kamali, M. Kolehmainen, M. Neovius, L. Petre, M. Rönkkö, P. Sandvik)

Abstract

An electrical grid is a structured framework of interconnected nodes. The purpose of this electrical grid is to connect the producers (generators) of energy, via a path of substations and disconnectors, with the consumers. The term smart electrical grid refers to an enhancement of the traditional grid, that provides and uses information from the cyber-physical control system by bi-directional cyber-secure communication. Thus, ‘smart’ refers to utilising the available information for self* properties when the grid is considered an entity in its own right. Typically the bi-directional communication is assured by mobile telecommunication technology. A categorisation on the physical realisation of the bi-directional communication and a survey on smart grids may be found elsewhere.

Research on smart electrical grids can be classified with respect to (i) infrastructure, (ii) management, and (iii) cyber-security. Smart infrastructure systems (i) are further divided into communication infrastructure, focusing on the communication technology and protocols; the information infrastructure, concerned with information interoperability; and the energy subsystem, concerned with, among others, small scale energy production such as solar panels. Second, the management system (ii) considers energy efficiency, operation costs reduction, demand and supply balance, emission control and utility maximisation. Third, the smart protection system (iii) is concerned with user errors, equipment failures, natural disasters and deliberate cyber-attacks. Our research is focused on the smart protection system and its feature of failure recovery, as well as on the management system, with respect to operation cost reduction. The former is studied via formal methods, whereas the latter is analysed through graph optimisation. Both of these are of outmost importance to increase the smart grids reliability and are strongly motivated by the annual costs of outage; for example, in 2002 these were estimated to 79 billion dollars in the US and we can only assume these figures have since risen.

In this paper, we are concerned with the number and position of redundant links. The paths made up of these links support the recovery of the smart electrical grid, which, at any moment, has the logical topology of a tree. Having the configuration of the smart electrical grid as a tree, reconnecting any compromised node or

subnetwork requires these idle redundant paths. A successful reconnection is characterised by having a grid with all consumers connected. Obviously, these idle redundant paths are constructed to minimise blackouts and circumvent causes of brownouts, i.e. to optimise the power uptime. The scale of redundancy for any node or subnetwork is a trade-off between the consequences in case of a failure and the costs of maintaining redundant paths, i.e., possible network configurations connecting the node or subnetwork to a producer. In this paper, we propose methods to find the critical nodes in the grid, i.e., the most reasonable places to attach redundancy to.

4.2.4 A Decision Support System Approach for Systems of Systems Management

A Decision Support System Approach for Systems of Systems Management (Maria Pia Fanti, Monica Clemente and Walter Ukovich)

Abstract

Systems of Systems (SoSs) are “large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal”. They find practical application in different fields of great interest nowadays, such as logistics and transportation, automotive and aerospace applications, smart grids, personal health management, domotic systems, integrated business processes, and so on. More and more, in fact, there is the need of integrating and coordinating several previously autonomous elements in order to better exploit their features and reach an overall level of efficiency otherwise not possible. However, due to the heterogeneity and the multi-disciplinarity of the involved systems, the management of SoSs is a very complex process that requires knowledge and technologies from many domains. Therefore, the application of advanced modelling techniques and systematic quantitative methods is a necessity.

In this work we formalize a methodology devoted to the management and control of a SoS based on the concept of Decision Support System (DSS). In particular, we describe the main components of the DSS and the tools employed in order to design the management strategy of such heterogeneous systems.

We highlight that such a methodology is completely general and can be applied regardless the specific context considered. However, with the aim of giving an example of the proposed approach, we consider a DSS devoted to management of Electric Vehicles (EVs) charging operations. In particular, we formalize the core of such a DSS: the Optimization and Decision Modules.

4.2.5 A Vision for Future Model-Based Support for Dependable CPSoS

A Vision for Future Model-Based Support for Dependable Cyber-Physical Systems of Systems (Claire Ingram, Ken Pierce)

Abstract

The development of dependable cyber-physical systems (CPS) and systems of systems (SoS) presents distinct challenges to stakeholders. When developing systems that exhibit both cyber-physical and SoS characteristics — cyber-physical systems of systems (CPSoS)— meeting the combination of these challenges may be beyond the current state of the art.

We present a rationale for adopting a model-based approach in this emerging field and present a vision of how model-based techniques could evolve to support the development of dependable cyber-physical systems of systems. We outline key challenges to be tackled when modelling CPSoS, which include the verification of emergent behaviour, coping with autonomous components and the integration of semantically different paradigms and concepts.

We will describe the state of the art in modelling techniques for CPS and SoS and consider these techniques can be leveraged to address the challenges seen in CPSoSs.

4.3 List of Participants Workshop Bertinoro

In total there were 15 participants in the workshop among which the following members of Working Group 3:

Fokkink	Wan	(VU Amsterdam, Netherlands)
Reniers	Michel	(TU Eindhoven, Netherlands)
Sonntag	Christian	(TU Dortmund, Germany)
Copigneaux	Bertrand	(inno, France)
Cimatti	Alessandro	(Bruno Kessler Foundation, Italy)

and the following participants that were identified by name:

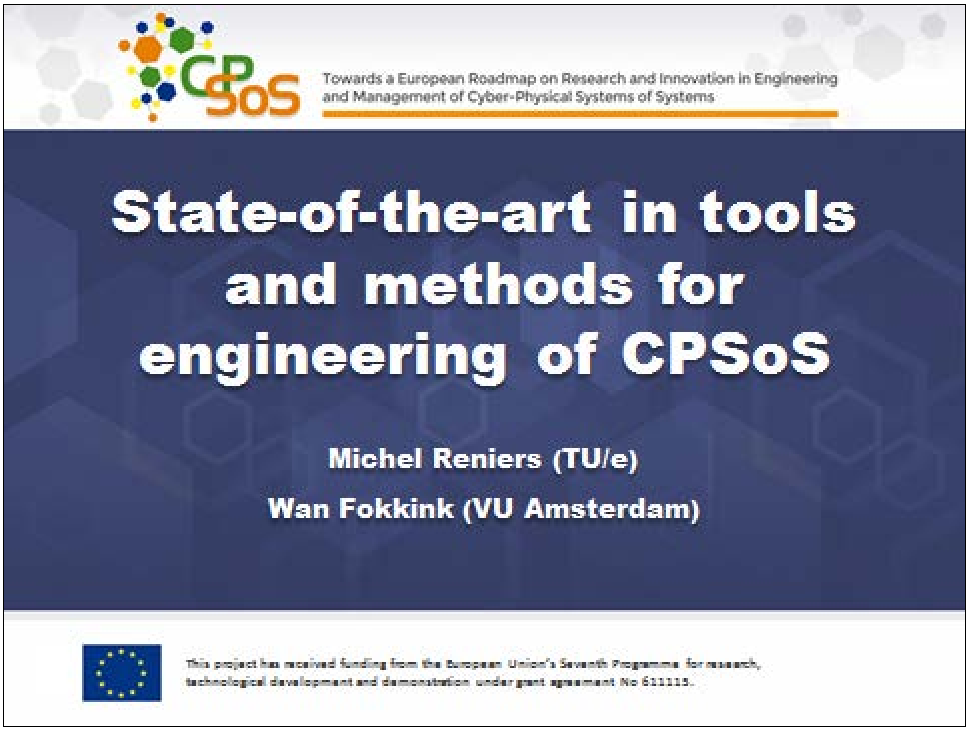
Sassone	Vladimiro	(University of Southampton, United Kingdom)
de Prada	Cesar	(Universidad de Valladolid, Spain)
Petre	Luigia	(Åbo Akademi , Finland)
Sandvik	Peter	(Åbo Akademi & Turku Centre for Computer Science, Finland)
Fanti	Maria	(University of Trieste, Italy)
Ingram	Claire	(Newcastle University, United Kingdom)

4.4 Photos from the Meeting

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

4.5 Presentation of the State-of-the-art

The presentation consists of a part introducing the CPSoS project to the audience and of a part describing the state-of-the-art.



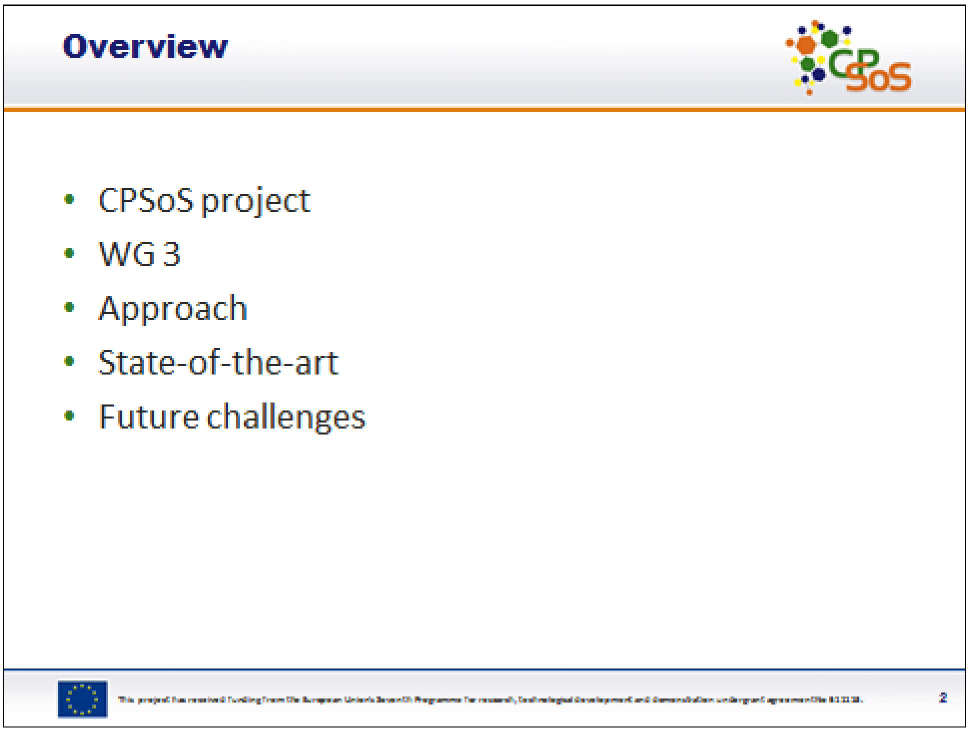
Towards a European Roadmap on Research and Innovation in Engineering and Management of Cyber-Physical Systems of Systems

State-of-the-art in tools and methods for engineering of CPSoS

Michel Reniers (TU/e)

Wan Fokkink (VU Amsterdam)

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115.

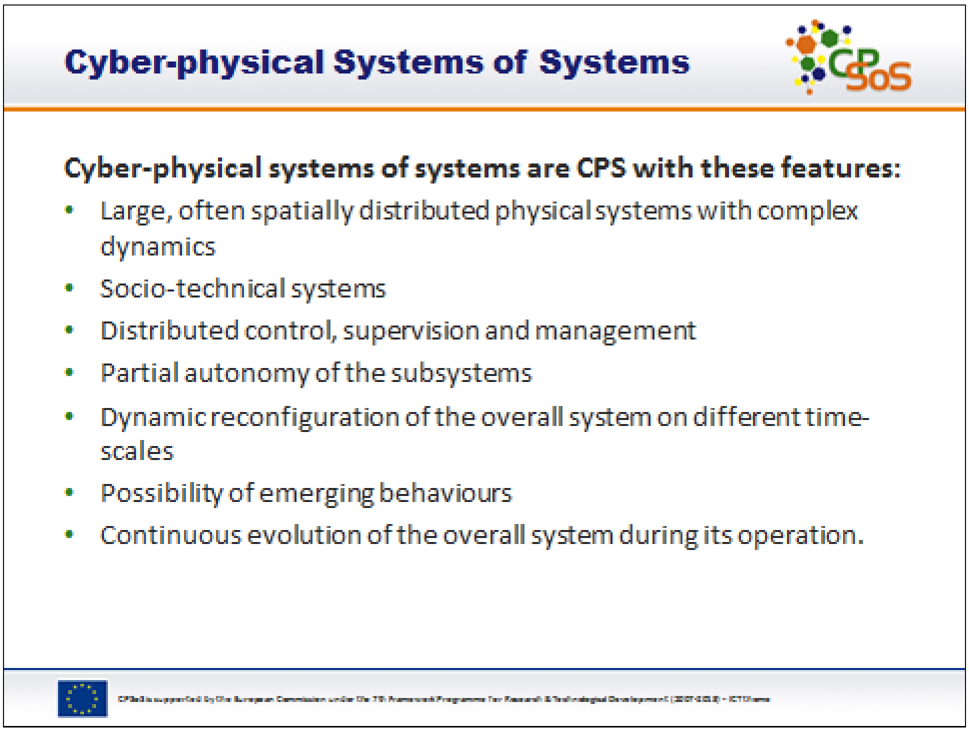


Overview

- CPSoS project
- WG 3
- Approach
- State-of-the-art
- Future challenges

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115.

2

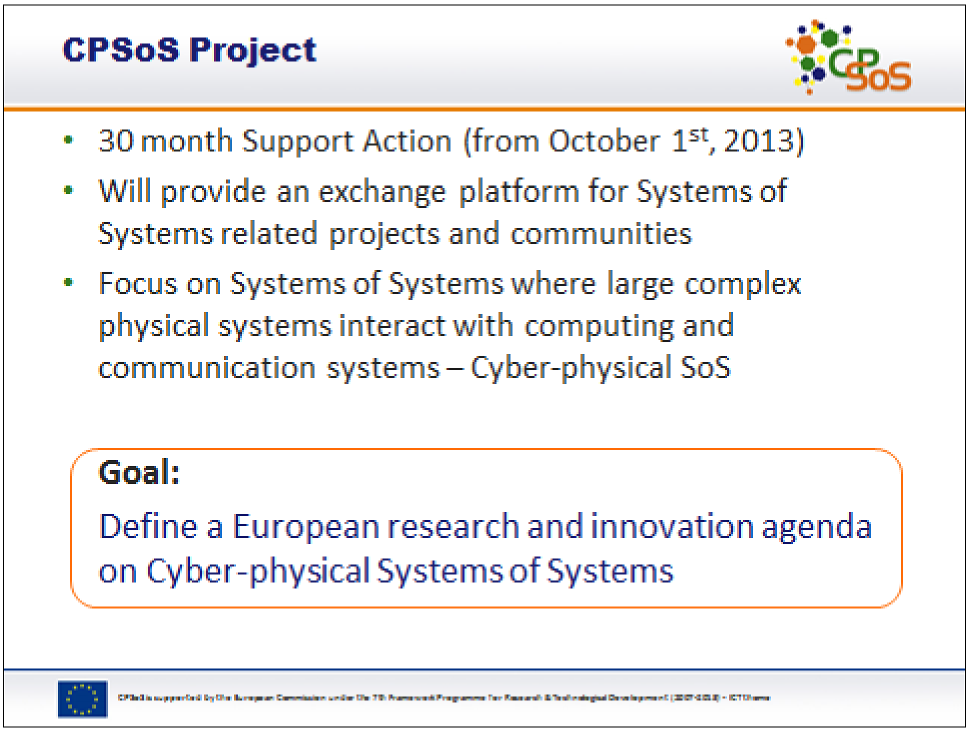


Cyber-physical Systems of Systems

Cyber-physical systems of systems are CPS with these features:

- Large, often spatially distributed physical systems with complex dynamics
- Socio-technical systems
- Distributed control, supervision and management
- Partial autonomy of the subsystems
- Dynamic reconfiguration of the overall system on different time-scales
- Possibility of emerging behaviours
- Continuous evolution of the overall system during its operation.

CPSoS is supported by the European Commission under the 7th Framework Programme for Research & Technological Development (2007-2013) - ICT theme



CPSoS Project

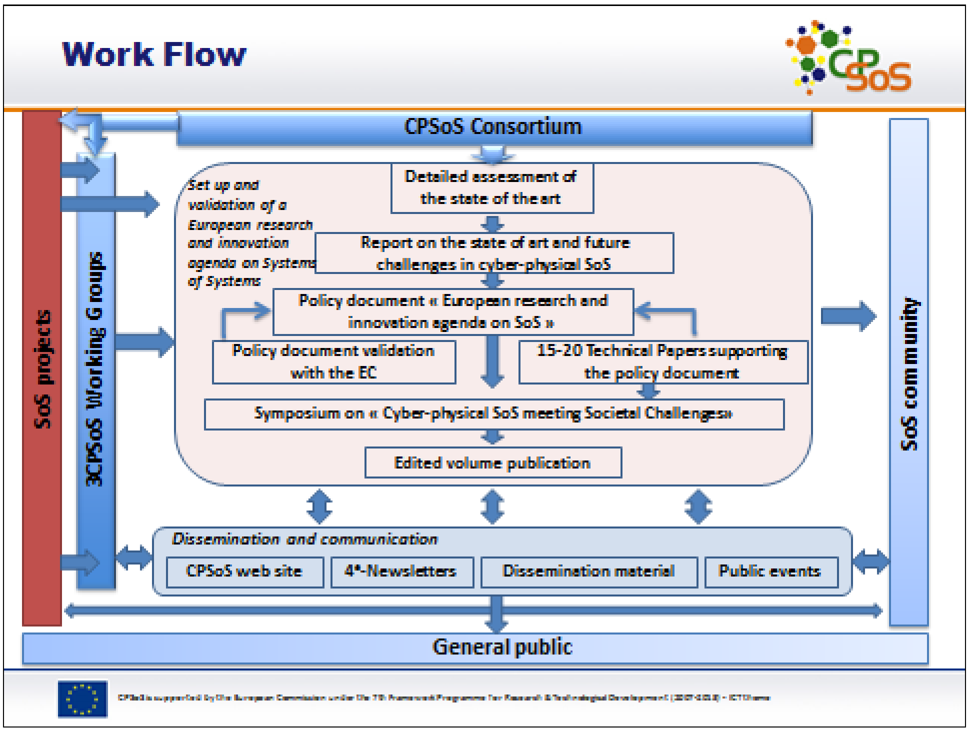
- 30 month Support Action (from October 1st, 2013)
- Will provide an exchange platform for Systems of Systems related projects and communities
- Focus on Systems of Systems where large complex physical systems interact with computing and communication systems – Cyber-physical SoS

Goal:

Define a European research and innovation agenda on Cyber-physical Systems of Systems

CPSoS is supported by the European Commission under the 7th Framework Programme for Research & Technological Development (2007-2013) - ICT theme

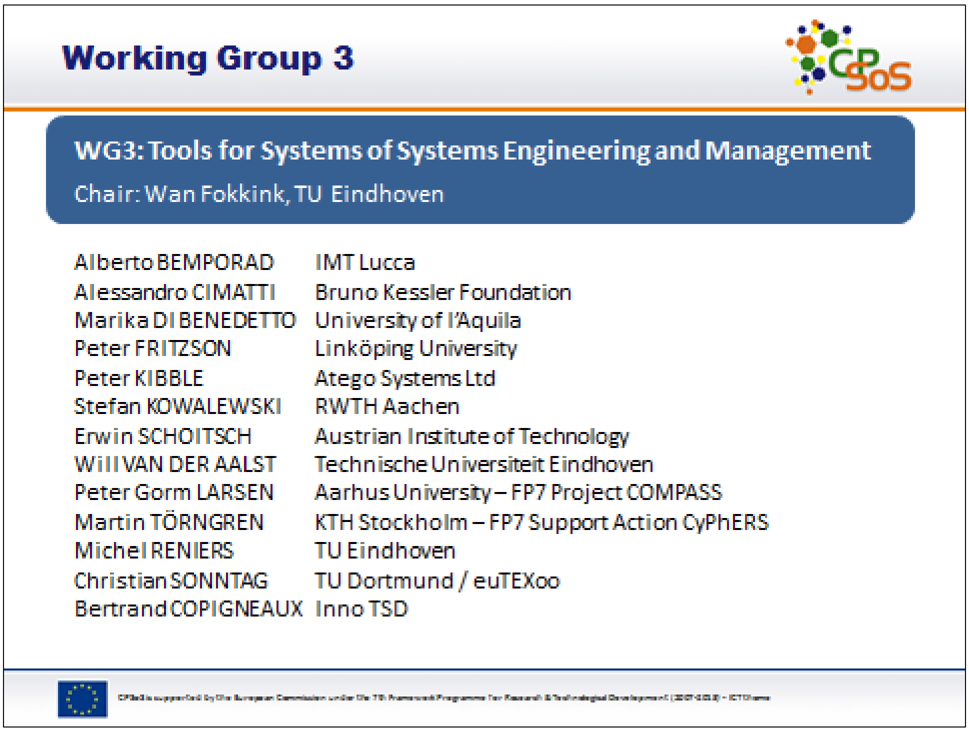
This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



Work Flow

CPSoS Consortium
SoS projects
3 CPSoS Working Groups
SoS community
Set up and validation of a European research and innovation agenda on Systems of Systems
Detailed assessment of the state of the art
Report on the state of art and future challenges in cyber-physical SoS
Policy document « European research and innovation agenda on SoS »
Policy document validation with the EC
15-20 Technical Papers supporting the policy document
Symposium on « Cyber-physical SoS meeting Societal Challenges»
Edited volume publication
Dissemination and communication
CPSoS web site
4*-Newsletters
Dissemination material
Public events
General public

CPSoS is supported by the European Commission under the 7th Framework Programme for Research & Technological Development (2007-2013) - ICT theme



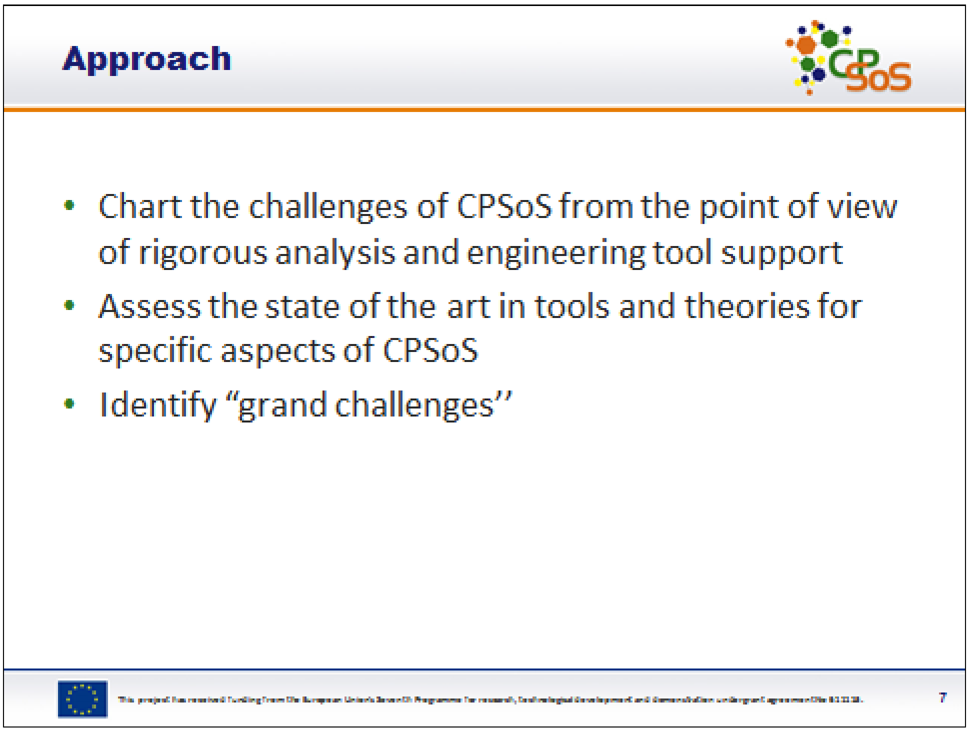
Working Group 3

WG3: Tools for Systems of Systems Engineering and Management

Chair: Wan Fokkink, TU Eindhoven

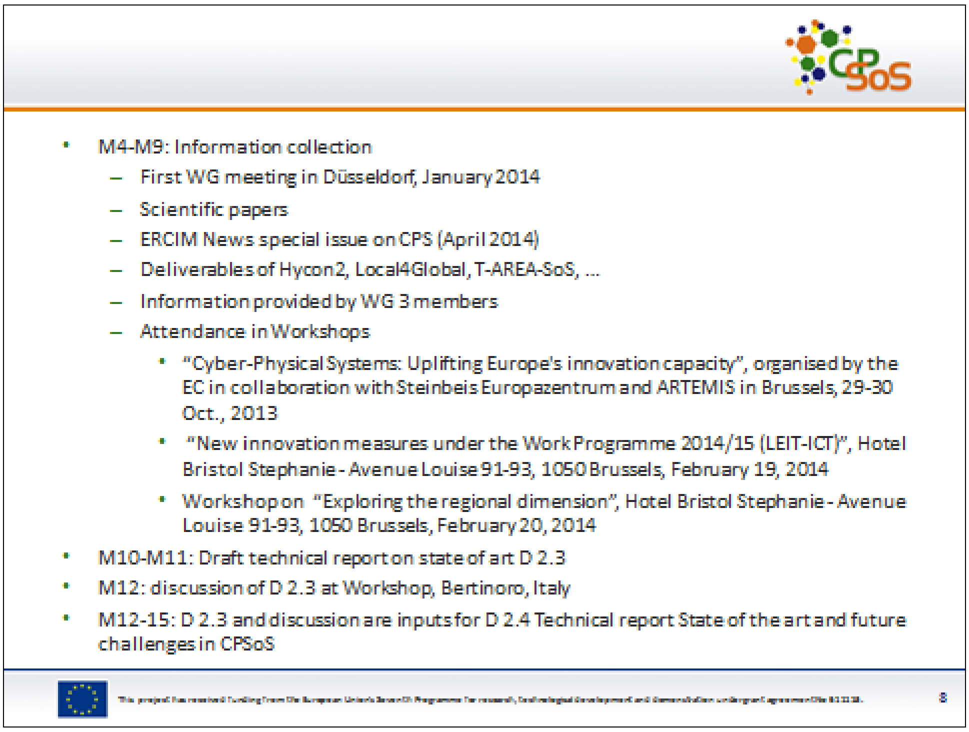
Alberto BEMPORAD	IMT Lucca
Alessandro CIMATTI	Bruno Kessler Foundation
Marika DI BENEDETTO	University of l'Aquila
Peter FRITZSON	Linköping University
Peter KIBBLE	Atego Systems Ltd
Stefan KOWALEWSKI	RWTH Aachen
Erwin SCHOITSCH	Austrian Institute of Technology
Will VAN DER AALST	Technische Universiteit Eindhoven
Peter Gorm LARSEN	Aarhus University – FP7 Project COMPASS
Martin TÖRNGREN	KTH Stockholm – FP7 Support Action CyPhERS
Michel RENIERS	TU Eindhoven
Christian SONNTAG	TU Dortmund / euTEXoo
Bertrand COPIGNEAUX	Inno TSD

CPSoS is supported by the European Commission under the 7th Framework Programme for Research & Technological Development (2007-2013) - ICT theme



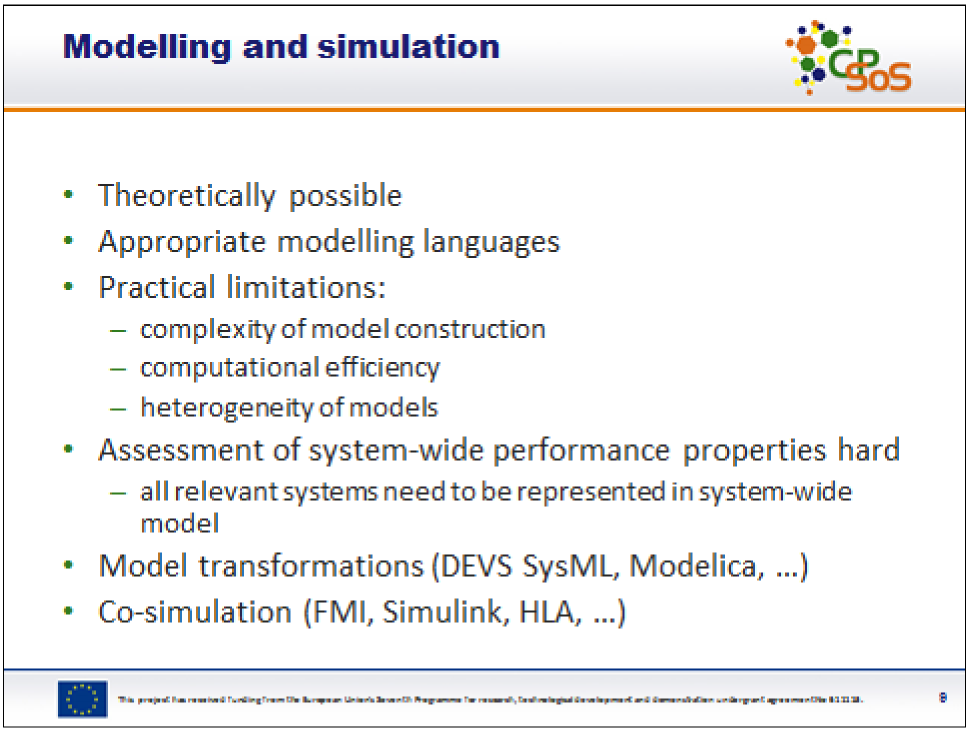
Approach

- Chart the challenges of CPSoS from the point of view of rigorous analysis and engineering tool support
- Assess the state of the art in tools and theories for specific aspects of CPSoS
- Identify "grand challenges"



- M4-M9: Information collection
  - First WG meeting in Düsseldorf, January 2014
  - Scientific papers
  - ERCIM News special issue on CPS (April 2014)
  - Deliverables of Hycon2, Local4Global, T-AREA-SoS, …
  - Information provided by WG 3 members
  - Attendance in Workshops
    - "Cyber-Physical Systems: Uplifting Europe's innovation capacity", organised by the EC in collaboration with Steinbeis Europazentrum and ARTEMIS in Brussels, 29-30 Oct., 2013
    - "New innovation measures under the Work Programme 2014/15 (LEIT-ICT)", Hotel Bristol Stephanie - Avenue Louise 91-93, 1050 Brussels, February 19, 2014
    - Workshop on "Exploring the regional dimension", Hotel Bristol Stephanie - Avenue Louise 91-93, 1050 Brussels, February 20, 2014
- M10-M11: Draft technical report on state of art D 2.3
- M12: discussion of D 2.3 at Workshop, Bertinoro, Italy
- M12-15: D 2.3 and discussion are inputs for D 2.4 Technical report State of the art and future challenges in CPSoS

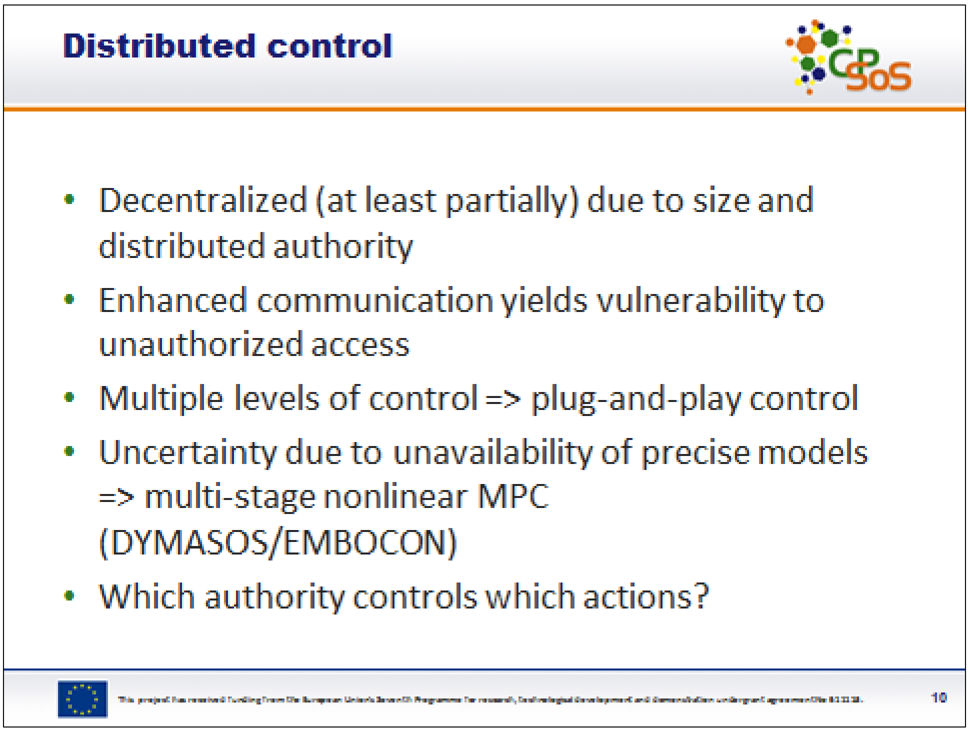
This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



Modelling and simulation

- Theoretically possible
- Appropriate modelling languages
- Practical limitations:
  - complexity of model construction
  - computational efficiency
  - heterogeneity of models
- Assessment of system-wide performance properties hard
  - all relevant systems need to be represented in system-wide model
- Model transformations (DEVS SysML, Modelica, …)
- Co-simulation (FMI, Simulink, HLA, …)

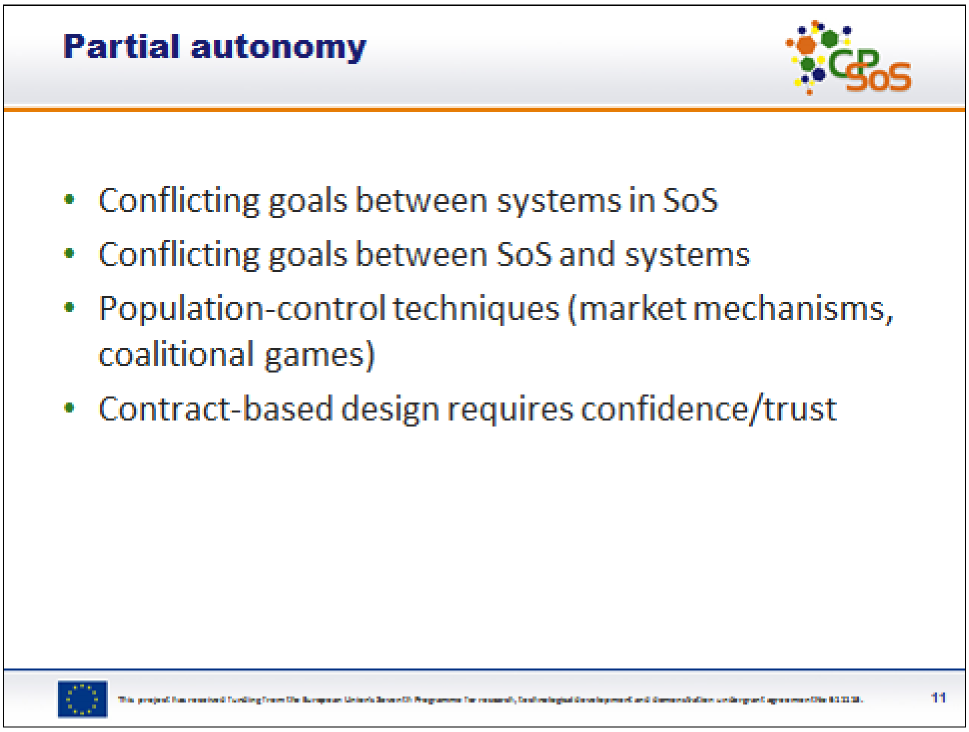
This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115. 9



Distributed control

- Decentralized (at least partially) due to size and distributed authority
- Enhanced communication yields vulnerability to unauthorized access
- Multiple levels of control => plug-and-play control
- Uncertainty due to unavailability of precise models => multi-stage nonlinear MPC (DYMASOS/EMBOCON)
- Which authority controls which actions?

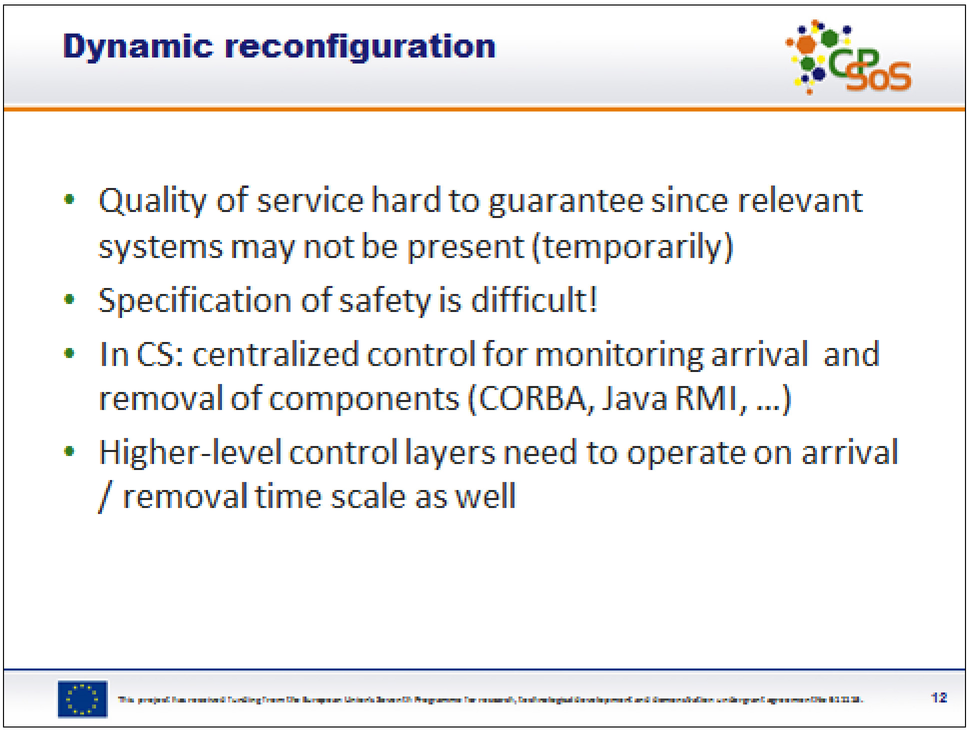
This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115. 10



Partial autonomy

- Conflicting goals between systems in SoS
- Conflicting goals between SoS and systems
- Population-control techniques (market mechanisms, coalitional games)
- Contract-based design requires confidence/trust

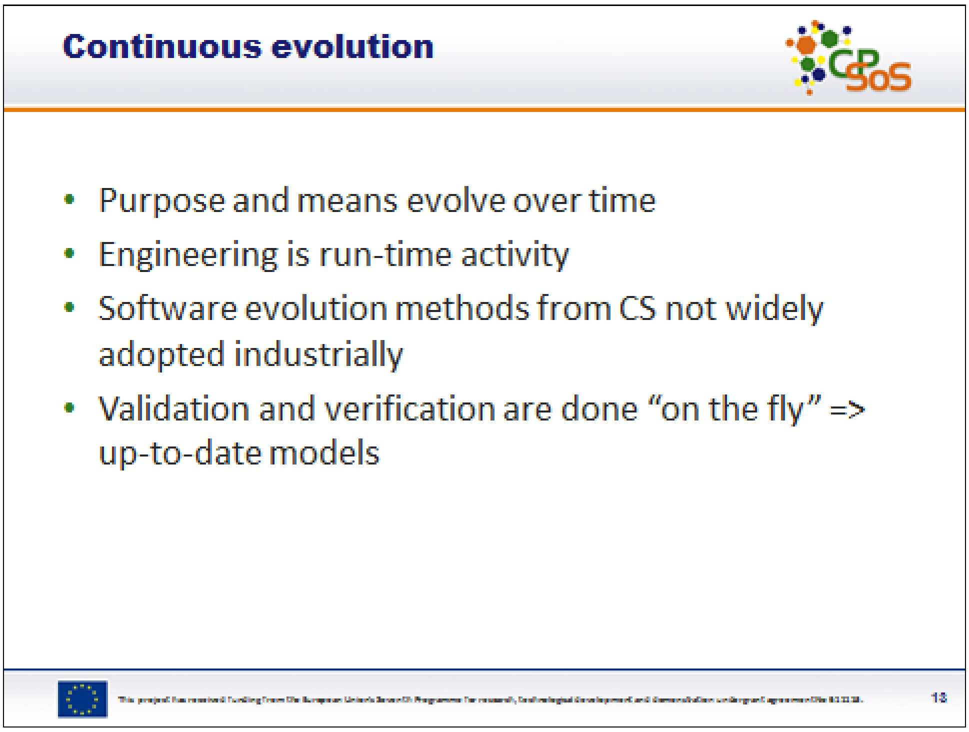
11



Dynamic reconfiguration

- Quality of service hard to guarantee since relevant systems may not be present (temporarily)
- Specification of safety is difficult!
- In CS: centralized control for monitoring arrival and removal of components (CORBA, Java RMI, …)
- Higher-level control layers need to operate on arrival / removal time scale as well

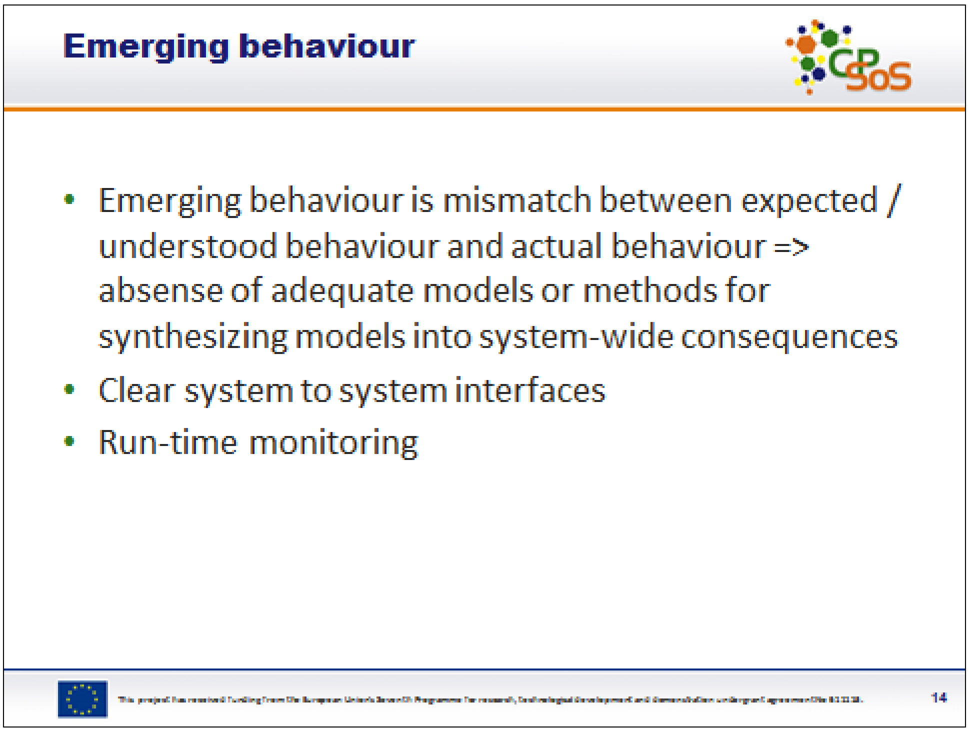
12



Continuous evolution

- Purpose and means evolve over time
- Engineering is run-time activity
- Software evolution methods from CS not widely adopted industrially
- Validation and verification are done “on the fly” => up-to-date models

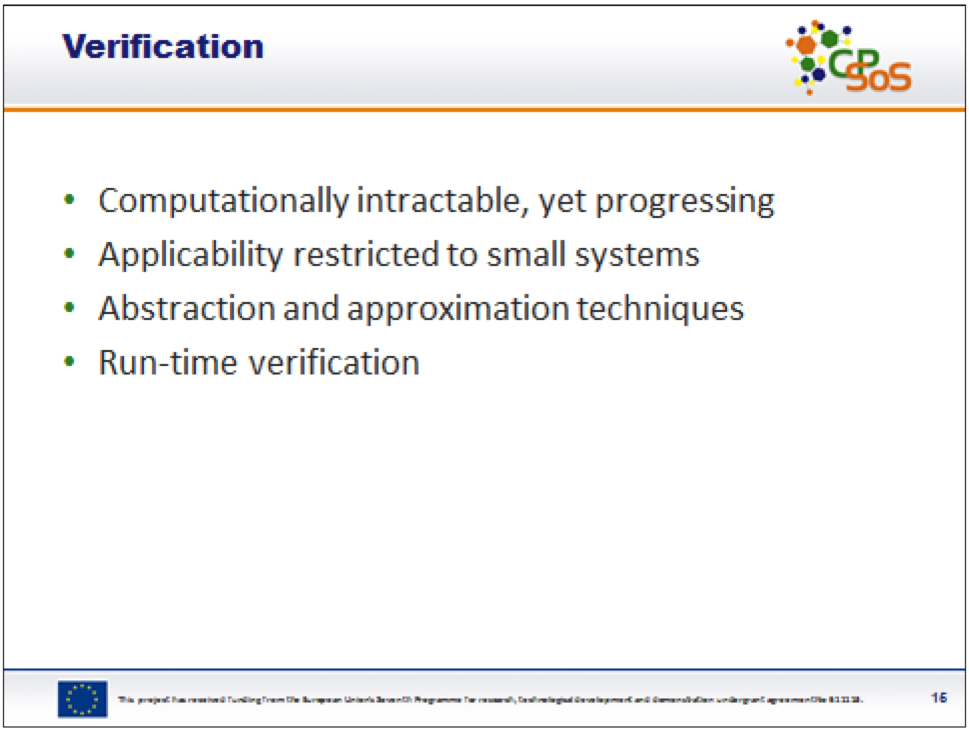
13



Emerging behaviour

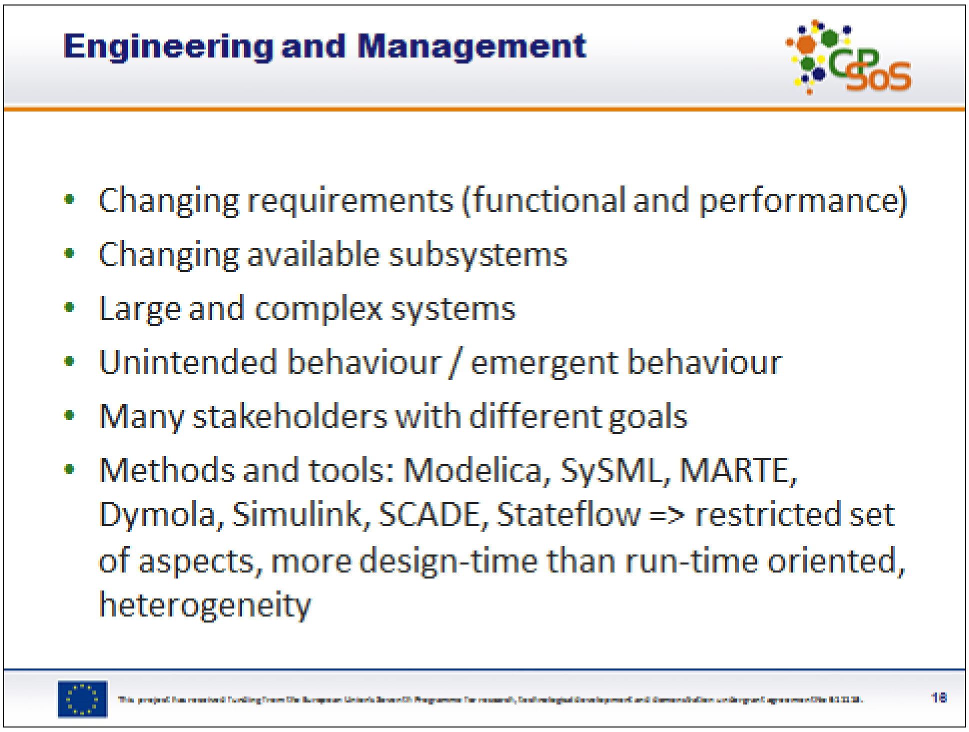
- Emerging behaviour is mismatch between expected / understood behaviour and actual behaviour => absense of adequate models or methods for synthesizing models into system-wide consequences
- Clear system to system interfaces
- Run-time monitoring

14



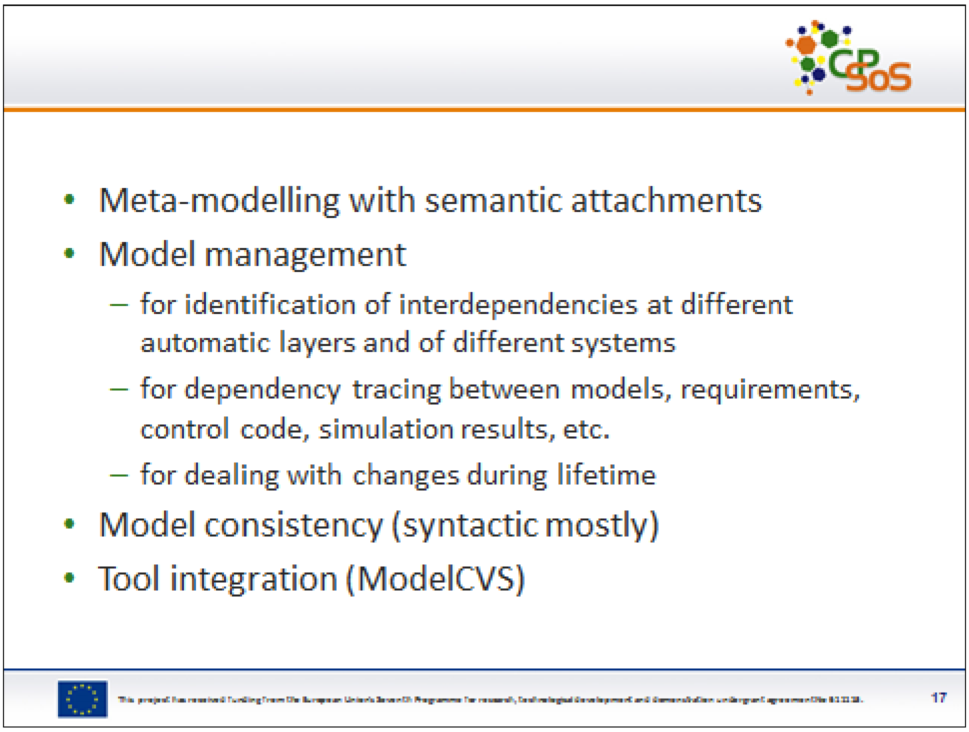
Verification

- Computationally intractable, yet progressing
- Applicability restricted to small systems
- Abstraction and approximation techniques
- Run-time verification



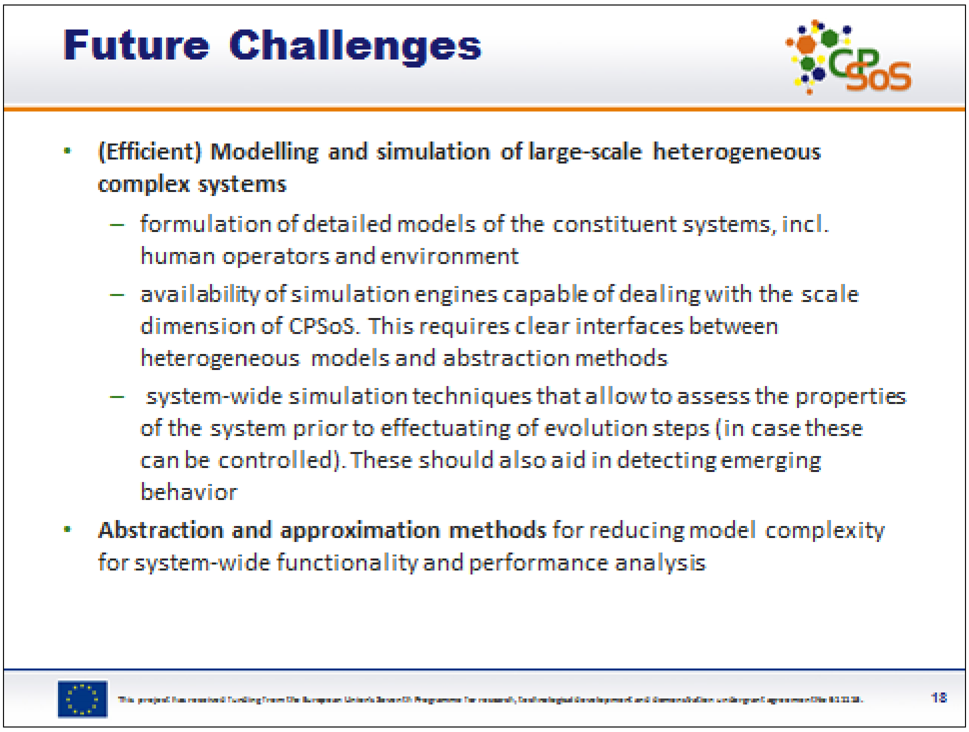
Engineering and Management

- Changing requirements (functional and performance)
- Changing available subsystems
- Large and complex systems
- Unintended behaviour / emergent behaviour
- Many stakeholders with different goals
- Methods and tools: Modelica, SysML, MARTE, Dymola, Simulink, SCADE, Stateflow => restricted set of aspects, more design-time than run-time oriented, heterogeneity



- Meta-modelling with semantic attachments
- Model management
  - for identification of interdependencies at different automatic layers and of different systems
  - for dependency tracing between models, requirements, control code, simulation results, etc.
  - for dealing with changes during lifetime
- Model consistency (syntactic mostly)
- Tool integration (ModelCVS)

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115.

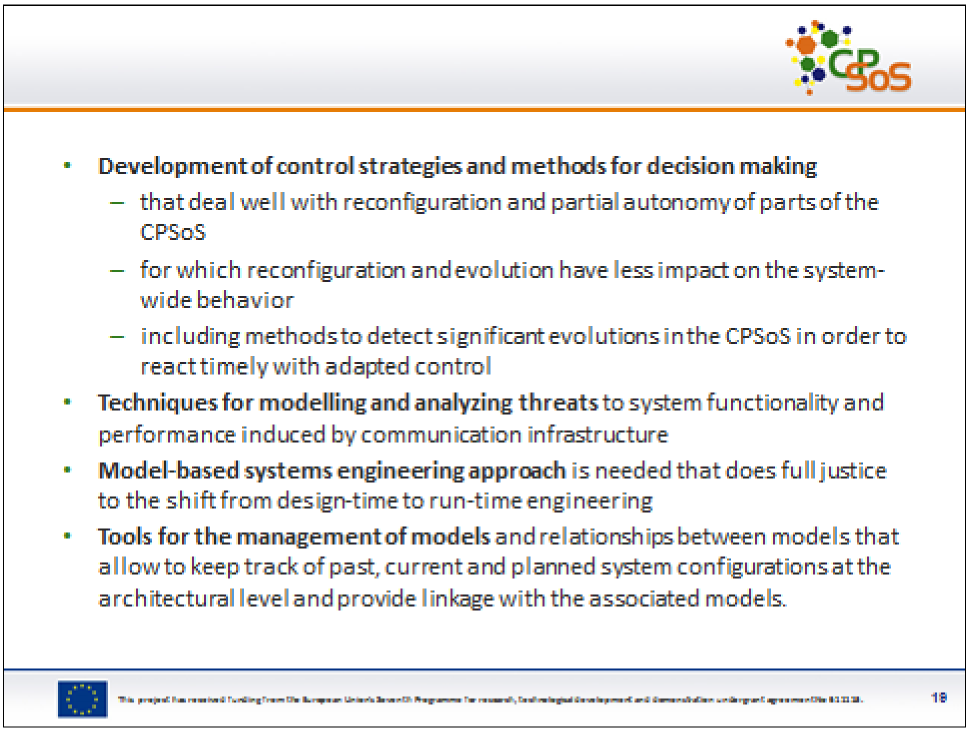


Future Challenges

- **(Efficient) Modelling and simulation of large-scale heterogeneous complex systems**
  - formulation of detailed models of the constituent systems, incl. human operators and environment
  - availability of simulation engines capable of dealing with the scale dimension of CPSoS. This requires clear interfaces between heterogeneous models and abstraction methods
  - system-wide simulation techniques that allow to assess the properties of the system prior to effectuating of evolution steps (in case these can be controlled). These should also aid in detecting emerging behavior
- **Abstraction and approximation methods** for reducing model complexity for system-wide functionality and performance analysis

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



- **Development of control strategies and methods for decision making**
  - that deal well with reconfiguration and partial autonomy of parts of the CPSoS
  - for which reconfiguration and evolution have less impact on the system-wide behavior
  - including methods to detect significant evolutions in the CPSoS in order to react timely with adapted control
- **Techniques for modelling and analyzing threats** to system functionality and performance induced by communication infrastructure
- **Model-based systems engineering approach** is needed that does full justice to the shift from design-time to run-time engineering
- **Tools for the management of models** and relationships between models that allow to keep track of past, current and planned system configurations at the architectural level and provide linkage with the associated models.

This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement No 611115.

19

4.6 Input Paper for State-of-the-art Presentation

State-of-the-art and Future Challenges in Tools and Methods for Engineering and Management of Cyber-Physical Systems of Systems

In this paper the state-of-the-art is described with respect to the methods (and tools) that are currently used explicitly for the characteristics associates with cyber-physical systems of systems as identified in the working paper [1] that was used as an input document for CPSoS Working Groups Kick off Meeting which took place in Düsseldorf/Germany on January 31st, 2014 [2].

For completeness the relevant parts of the mentioned definition are iterated here.

Cyber-physical Systems of Systems are cyber-physical systems which exhibit the features of systems of systems:

- Large, often spatially distributed physical systems with complex dynamics
- Distributed control, supervision and management
- Partial autonomy of the subsystems
- Dynamic reconfiguration of the overall system on different time-scales
- Possibility of emerging behaviors
- Continuous evolution of the overall system during its operation.

In the next sections, first, for each of these characteristics a short discussion of accepted state-of-the-art approaches is presented. Research challenges are introduced.

Large-scale Systems with Complex Dynamics

From the working paper "Cyber-physical Systems of Systems – Definition and core research and development areas" [1] the characteristic of large-scale systems with complex dynamics is explained as follows: Cyber-Physical Systems of Systems consists of a *significant* number of *interacting* components that are (partially) physically coupled and together fulfill a certain function, provide a service, or generate products. The components can provide services independently but the performance of the overall system depends on the "orchestration" of the components.

Here, complex dynamics refers to systems having continuous-time behaviour that is described by means of systems of differential equations that are hard to solve or approximate algorithmically. Such examples arise among others in the domain of chemical batch processes. Complex dynamics also arise in systems having many different modes of continuous behaviour with complex switching between these, and in large-scale systems where the geographic (or functional) distribution requires communication mechanisms that introduce information loss and time delays.

In order to assess the functional and performance properties of a cyber-physical system of systems model-based methods are used that allow for modelling and simulating the complex dynamics on a system-wide level. For model-based analysis it is needed to have models that are capable of expressing complex dynamics. There are several approaches towards the description of such complex dynamics. With respect to the continuous dynamics of such systems these are dominated by methods that use differential and difference equations. It is well recognized that in cyber-physical systems (of systems) besides dynamics also the switching between different dynamics needs to be considered. This has resulted in many formalisms with well-established semantics such as hybrid automata [3], hybrid Petri Nets [4], and other hybrid process theories [5] [6].

Research in systems and control in networked control systems [7] is focused on extending techniques for control systems to also deal with the network that facilitates non-physical interaction between components. Distributed control of such networked control systems will be discussed in the next section. Research in the area of networked control systems is focused in the following areas [8]: (1) evolution and research in networking technologies, (2) effect of network delays (modelling and analysis, compensation for delays), (3) fault-tolerant control, (4) bandwidth allocation and scheduling, (5) network security, and (6) integration of components.

In the FP7 project FeedNetback [9] a co-design framework has been proposed to integrate architectural constraints and performance trade-offs from control, communication, computation, complexity and energy management. FeedNetback contributed in mastering complexity, temporal and spatial uncertainties such as delays and bandwidth in communications and node availability. This approach enables the development of more efficient, robust and affordable networked control systems that scale and adapt with changing application demands. The outcome has been demonstrated on underwater inspection systems using fleets with autonomous underwater vehicles and on a system of smart camera's for surveillance and security.

Modelling and simulation of large-scale systems with complex dynamics is theoretically possible. However, severe practical limitations are the complexity of the construction of the models, computational efficiency, and the heterogeneity of these models in different dimensions. Construction of the large number of involved models is time-consuming and needs deep domain knowledge (from very different areas). Models that are encountered in cyber-physical systems of systems are different in modelling paradigm that is used (continuous-time, discrete event, stochastic, computation, …) as well as in formalisms that are used for expressing models within a certain paradigm (e.g. automata-based descriptions versus process algebraic specification languages).

To enable assessment of system-wide performance properties of a cyber-physical system of systems it is needed to represent all relevant (contributing) systems in a system-wide model. This is not always possible. There may be different reasons for the lack of appropriate models:

- The efforts needed for obtaining a detailed model are not in line with the anticipated gain of having these models. This may result in models that only approximate the behavior of the system coarsely.
- Different (legal) entities are involved that may not desire or are not allowed to provide an accurate model of the system.
- Scientific knowledge has not reached the level of understanding in order to explain the causalities that result in the behavior anticipated as emergent. As an example of these we may consider the human as a system in a system of systems.

Simulation of cyber-physical systems of systems suffers from computational challenges such as the numerical integration of very complex continuous behavior, execution of discrete model parts, and re-initialization of continuous quantities after discrete events [10] [11] [12].

There is a large variety of modeling formalisms and simulation tools around, which. are often designed specifically for particular domains [13] [14] [11] [12] [15] [16] [17]. Following [17], the aggregate models can be obtained either via automated model transformation or via co-simulation. Automated model transformation requires a semantics-preserving mapping between different formalisms which can be executed by means of an algorithm. Such transformations may then be used to map models to the same modelling language. Limitations of the applicability of this approach on cyber-physical systems of systems are the very different uses of time in the modeling formalisms, and the very rich palette of modeling constructions offered by the modelling languages such as Modelica [18], gPROMS [19] and EcosimPro [20] encompassing object-orientation, hierarchical modeling, and complex dynamics (Integral/Partial Differential Algebraic Equations), and algorithmically specified computations. A number of languages have been developed that aim to provide a common model exchange format:

- DEVS (Discrete Event System Specification) [21] is an exchange format for discrete event models
- SysML [22] is a customization of UML for systems engineering applications with hierarchical high level models
- UML extensions [23] with hybrid model representations
- ModelCVS [24] allows ontology-based tool integration
- Modelica [25] [26] for exchange of equation-based models
- CIF (Compositional Interchange Format) [27] for hybrid systems modeling with complex continuous DEA dynamics, hierarchy and modularization

In many cases [17] model transformation is not an option. In these cases co-simulation may be applied. Co-simulation is the synchronized simulation of different models in their original formalisms and with their own simulation tools. An example is the Crescendo tool which combines 20-sim for continuous-time modelling and uses the VDM notation for discrete-event modelling [10] [11]. Due to increased computing powers contained in even reasonably standard desktops one may approximate real-time simulation of complex dynamics, as demonstrated for cardiac dynamics [12].

Co-simulation platforms, which adapt already existent tools, can be classified as follows [28]:

- Extending physical systems simulators to also simulate the events and dynamics of communication networks. Examples are the original TrueTime [29] version that is based on Simulink, the more recent Modelica-based versions of TrueTime [30] and VisualSense [31]. These tools lack support for routing, transport, and application protocols.
- Extending a network simulator to support physical systems simulations. An example is Agent/Plant [32]. Physical dynamics and control algorithms are modeled explicitly by differential–algebraic equations (DAEs) that are to be solved within the simulation script or via a call to an outside utility such as Matlab.
- Marrying a full-blown network simulator with a full-blown physical systems simulator. Examples include the ADEVS/ns-2 integrated tool [33], the Simulink/ns-2 combined tool [34], and the Modelica/ns-2 integrated tool [28].

Since each of the contributing realms to cyber-physical systems of systems design has its own specialized and mature simulation tools, it seems appropriate to combine domain-specific tools into one that utilizes the best features of individual simulators.

There are many academic co-simulation frameworks such as the Simantics Open Simulation Platform, TrueTime, CHEOPS, and SIMCAN. These lack industrial relevance and/or are focused on a very specific domain. Also the commercial frameworks dSpace SystemDesk, qTronic Silver, SKF, CAPE-OPEN, OPC and TISC are targeted for very specific domains. The following are a number of co-simulation frameworks that are both of industrially relevant and general purpose.

- FMI (Functional Mockup Interface) is developed in the EU project MODELISAR and provides an open tool-independent standard for co-simulation of hybrid dynamic models [35] [36]. The FMI only supports continuous dynamics in ODE form. Well-established tools such as Dymola and OpenModelica [37] support FMI.
- Simulink is a tool for the modeling and simulation of nonlinear dynamic systems [38]. Discrete event models can be integrated using Stateflow and SimEvents and acausal components can be added using the SimScape toolbox. Simulink uses the s-function interface that must be implemented by external software components to be integrated into Simulink models.
- HLA (High Level Architecture) [39] is an IEEE standard for distributed simulation. HLA establishes a framework in which simulation components interact via services from the Runtime Infrastructure (RTI).

It is well known that verification of (discrete) systems is computationally intractable, yet progress is made in the development of methods, algorithms and tools for the verification of both discrete event and continuous-time systems [40]. In [41] the authors report a new method (continuization) for computing accurate over-approximations of reachable sets of states for hybrid systems which may have a large number of discrete switching.

The tool KeYmaera [42] is a formal verification tool for cyber-physical systems. It has seen many real-world applications, e.g. verifying non-collision in the European Train Control System. KeYmaera’s capability comes from Platzer’s differential dynamic logic [43]. Differential dynamic logic generalizes very elegantly to dynamic logics of multi-dynamical systems. Multi-dynamical systems, which are systems that combine multiple dynamical aspects such as discrete, continuous, stochastic, nondeterministic, or adversarial dynamics, generalize hybrid systems and provide the dynamical features of cyber-physical systems that hybrid systems are missing.

Although promising techniques are developed for the verification of multi-dynamical systems, currently the size of the systems to which verification by means of model checking can be applied successfully is still very limited and not adequate for application on heterogeneous large-scale systems.

It is expected that ongoing research will generate improved methods and tools for simulating and verifying a cyber-physical system of systems, but given the current state of the art it may not be expected that these are applicable immediately to the heterogeneous large-scale systems, involving also computational and communication elements that fall within the scope of cyber-physical systems of systems. Given this observation, the need arises for abstraction and approximation methods that allow to consider only the details of the system that are of relevance to the analysis questions at hand, such as the stability of the cyber-physical system of systems under both normal and exceptional conditions.

Research challenges are:

- (Efficient) Modelling and simulation of large-scale heterogeneous complex systems
- Integration and management of large collections of heterogeneous models
- Abstraction and approximation methods for reducing model complexity

Distributed Control, Supervision and Management

In a typical cyber-physical system of systems the control, supervision and management tasks are usually not performed in a completely centralized or hierarchical top-down way. This is partly due to the size of the system and also due to present distribution of authority in the system.

In a cyber-physical system of systems communication between the constituent systems takes place both between the physical subsystems by exchanging material and energy and at all levels of the control hierarchies that are in place via sensors and actuators and communication channels. These enhanced possibilities for communication on the one hand allows for more possibilities for management and control but on the other hand induces many spots in the system where unauthorized access may be gained to the system. Challenges in this respect are attack modelling, and simulation of such attacks on the system, detection of attacks, and development of control structured with self-healing defence mechanisms [44].

Management of cyber-physical systems depends more on socio-technical performance indicators than on technical criteria. The management of these systems requires translation of these performance indicators into control on a technical level of the cyber-physical system of systems.

A recent roadmap published in the context of the Network of Excellence HYCON2 [45] shows that there is much emphasis on the development of methodological approaches for the coordination and control of cyber-physical systems of systems. The common denominator in these approaches is the use of multiple levels of control which are usually inspired by geometrical or time/spatial considerations. A drawback of these approaches is that they require a centralized off-line design phase which requires the knowledge of the whole system. This does not go particularly well with some of the characteristics of cyber-physical systems of systems as provided in [1], most notably dynamic reconfiguration and continuous evolution. Research in Plug-and-Play control overcomes this drawback by using only models of a system and its neighbours for developing local control [46].

In cyber-physical systems of systems there is generally much uncertainty as a consequence of unavailability of precise models and the openness of these systems. In order to deal with uncertainty, multi-stage nonlinear model predictive control has been proposed [47]. This approach makes it possible to take into account that in the future new information may be available. This work is continued in European projects such as DYMASOS (www.dymasos.eu) and EMBOCON (www.embocon.org).

It is common practice to discriminate different classes of systems of systems based on the control structure that is in place (section 3.3 of [48]): directed, acknowledged, collaborative and virtual. For the different classes there are different challenges as to how to supervise the system of systems. For directed systems of systems this is relatively straightforward. This situation is best compared to the local supervision mentioned before. For other classes it may be problematic to decide where the authority lies on actions that are allowed or disallowed for the systems of the systems of systems. Engineering the control and supervision requires that it is clearly stated what are the (agreed) supervision goals (avoiding collisions, optimization goals). It is also required that it is clearly stated which entity has which authority to enforce joined behavior.

The EU FP7 project Local4Global (www.local4global-fp7) aims to develop and extensively test and evaluate a generic, integrated and fully-functional methodology/system for technical systems of systems (TSoS) with

- a self-learning mechanism for the identification of the TSoS dynamics
- a situation-awareness mechanism for the constituent systems
- a distributed optimizer to determine local control actions
- a control-for-learning and learning-for-control mechanism

The system will be deployed and tested for a traffic case and an efficient building case. The advantages and disadvantages of the developed system for application on cyber-physical systems of systems may deliver interesting ideas for further research. Related projects are AGILE (rapidly-deployable, self-tuning, self-configurable, nearly-optimal control design), HYDRA (middleware platform for heterogeneous devices), BEAMS (buildings energy advanced management system), and EPIC-HUB (fully interoperable middleware solution.

Research challenges are:

- Development of control techniques for which reconfiguration and evolution have less impact on the system-wide behavior
- Techniques for modelling and analyzing threats to system functionality and performance induced by communication infrastructure

Partial Autonomy

In cyber-physical systems of systems where (some of) the involved systems are partially autonomous any centralized control should be aware that these systems cannot always be made to comply with global decisions. Conflicts between goals of systems and between goals of systems and global system of system goals will arise. Selfish behavior of systems may even result in degradation of quality of services throughout the system of systems since each constituent system insists in fulfilling its own objectives.

In many Cyber-Physical Systems of Systems partial autonomy is present because the managerial units that are responsible for the existence of the constituent systems are partially autonomous. As a consequence the possibilities for exercising centralized control may be restricted. In the ongoing FP7 project DYMASOS, but also in the Network of Excellence Hycon2, research is performed into population-control techniques, market-like mechanisms and coalitional games. Examples of practical cases where such mechanisms may be applied are chemical production sites, electric vehicle charging and electric distribution grids.

The constituent systems in a cyber-physical system of systems join forces, deliberately or not, to reach a common system-wide goal. Since the constituent systems are partly autonomous, they are free to negotiate agreements (contracts) that bind all partners to cooperate. Agreements between systems are based on a certain level of confidence that the constituent systems that are involved in the agreement will comply. The establishment of a high level of trust among autonomous constituent systems is an issue of utmost importance in the engineering and management of a CPSoS [49].

In the context of design of embedded systems contract-based design [50] has been coined as a technique that addresses some of the concerns in the design of cyber-physical systems of systems. In this sense a contract represents both the assumptions on the environment and the guarantees of the system under these assumptions. The use of contracts supports open system development as contracts abstract from details from the possibly unknown aspects of the environment and focus on the properties that the context is supposed to deliver. Contracts are useful in managing requirements and in fusing viewpoints. Finally, contracts, at least in principle, allow contract composition and successive refinement steps in a design process. Assume/Guarantee contracts were developed in the SPEEDS project [51] and the CESAR project [52]. They have been extended to real-time and stochastic settings in [53] and [54], respectively.

Research challenges are:

- Development of methods for decision making in a cyber-physical system of systems context, involving constituent systems which are partially autonomous and may act unpredictably.

Dynamic Reconfiguration

Dynamic reconfiguration refers to the addition or removal of components on different time scales, depending on the nature of the system and the reasons for the changes of the structure and changes of the way the system is operated. It includes systems where components come and go (like in air traffic control) as well as the handling of faults and the change of system structures and management strategies following changes of demands, supplies or regulations.

The presence of possibilities for dynamic reconfiguration as defined above implies that quality of service can only be guaranteed insofar the services that are needed to guarantee this QoS are present in the system of systems, and can be located and used by the requesting system.

To guarantee safety of the system of systems one needs to specify the allowed or disallowed global system states in terms of global concepts. It is not useful to state safety in terms of a local system states since these systems may be removed.

Approaches advocated in the engineering of computer systems address dynamic reconfiguration at the architectural level and mostly assume that a centralized control is available for monitoring arrival and removal of components [55]. In computer science much research has been performed in dynamic reconfiguration and this has resulted in many commercially available solutions that aid in the engineering of such systems such as CORBA, Java RMI and DCOM [56].

In computer science literature, there is a distinction made into foreseen reconfigurations, and unforeseen ones. The latter type is much harder to deal with in engineering of CPSoS as these are not anticipated at design-time. Engineering of CPSoS in which unanticipated reconfiguration appears, requires that accurate models of the entire CPSoS are maintained at run-time in order to assess functional and performance properties of the system as a whole.

Dynamic reconfiguration techniques are widely used for designing efficient System-on-Chip (SoC) architectures. Many promising reconfiguration methods exist including dynamic scaling of processor voltage levels,

This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

reconfiguration of cache hierarchy and communication architectures to improve both energy consumption and overall performance in SOC architectures. Although they receive considerable attention in various domains recent years, dynamic reconfiguration techniques haven't widely been employed in real-time systems. This is due to the fact that such systems consist of tasks with time constraints and missing task deadlines may lead to catastrophic effects in safety-critical systems. Dynamic reconfiguration normally will change the task's execution time. Moreover, additional computation required for making decisions at runtime may adversely affect the task schedulability. The problem is further aggravated in the presence of aperiodic or sporadic tasks where task attributes are not known in priori. The goal of this research is to exploit the advantages of dynamic reconfigurations in real-time embedded systems. See http://esl.cise.ufl.edu/reconfiguration.html.

A consequence of reconfiguration is also that the higher level control layers such as system-wide supervisory control and optimal control need adaptation at a similar time scale as the (possible) occurrences of the reconfigurations in order to guarantee that the QoS of the system as a whole remains within acceptable levels. For this it is needed to have means, either inside or outside the CPSoS of detecting that reconfiguration occurred.

In control theory plug and play control has been introduced as a technique that performs well in systems where reconfiguration issues play a role [46]. The method has been applied amongst others to the design of the AGC layer in power networks. A tool for the implementation of this technique is integrated in the HYCON2 toolbox.

Research challenges are:

- Development of control strategies that deal well with reconfiguration of the system of systems

Emerging Behaviours

There is much debate on the use and abuse of the term emergent behaviour [49]. The point of view adopted in the CPSoS project is that the emerging behaviour is to be restricted to the occurrence of patterns, oscillations or instabilities on a systems level and the formation of structures of interaction in a way that had not been anticipated in the construction of the subsystems and in the design of their interactions.

In the context of the AMADEOS project [57], in [49] work is done on emergent phenomena in a system-of-systems. Reasons for the occurrence of such phenomena are presented. It is shown how emergence manifests itself in an SoS and elaborated on the diverse causes of emergence. Guidelines for the system designer that should reduce the occurrence of detrimental emergent phenomena in a System of Systems are presented.

Emergent behavior is not so much a property of the cyber-physical system of systems, but rather a mismatch between expected and understood consequences of the cyber-physical system of systems and the actual behavior it displays. This mismatch is closely related to a lack of a complete enough understanding of the system and the complexity of the cyber-physical system of systems (both in terms of size and complex dynamics). In other words, it indicates absence of adequate models or methods to synthesize models of the systems into its system-wide consequences.

In [58] it is argued that clearly defining system-to-system interfaces is key to understanding emergent behavior. Another method to deal with emerging behaviour not at design time but rather at run time is to equip the cyber-physical system of systems with monitors which may then be used to reconfigure or adapt the system behaviour [59]. Run-time monitoring and verification is discussed previously.

Research challenges are:

- Formulation of detailed models of the constituent systems, including human operators and models of the environment.
- Availability of simulation engines that are capable of dealing with the scale dimension of cyber-physical systems of systems appropriately. This requires clear interfaces between heterogeneous models and abstraction methods that can be used for representing complex models with adequate simplified ones.

Continuous Evolution

Cyber-physical systems of systems are systems that evolve continuously over long periods of time both in terms of the purpose they serve as the means they have for achieving those. As a consequence the engineering of such a system has to be performed at run-time. The waterfall paradigm “Requirements – modelling – model-based design – verification – commissioning – operation - dismantling” is not applicable to systems of systems where the requirements change during operation.

A systematic review of software architecture research [60]identified that most methods and tools that are developed for the purpose of software evolution are not widely established in industrial practices. Among others, the development of foundation theories with practical value to software architecture evolution is identified as a research challenge. Novel methods and tools are needed to design ultra-large-systems that integrate and orchestrate the evolution of thousands of platforms, decision nodes, organizations and processes.

In the DEECo project [61] a framework is presented that allows software engineering in the context of smart cyber-physical systems and addresses the characteristics autonomy, dynamic reconfiguration and continuous evolution to some extent. Component (software units of development and deployment) may be assembled dynamically into collaboration groups. Interaction between components is handled by a centralized execution environment. Global system invariants are maintained by component coordination and are decomposed into component processes or collaboration group interactions. This approach is referred to as the Invariant Refinement Method [62]. Reported unresolved issues in the use of this framework are uncertainty of knowledge and verification in the presence of dynamicity.

Validation and verification has to be done “on the fly”. This strengthens the role of models in the engineering processes. Up-to-date (because continuously updated) models of the running operation can be used for both purposes. If they are adapted to the real operational practice, they reflect reality better than the original engineering models and can be used to investigate options for modifications as well as improved operational policies without modifications.

The behaviour if cyber-physical systems of systems depends heavily on the environment and changes over time, which makes their behaviour hard to analyse prior to execution. Runtime verification [63] is a lightweight verification technique which is performed in runtime and as such may suffer less from the evolutionary aspects of a cyber-physical system of systems. By means of a monitor process it is decided check whether expected and actual behaviour are well enough aligned. Runtime verification may be an interesting technique in cases where certain information is only available at runtime, in cases where precise models are not available, and in safety-critical systems for increasing confidence that the system behaves safely. Run-time verification continuous signals is studied in [64]. Runtime monitoring for stochastic cyber-physical systems is discussed in [65]. Compared to other verification approaches, runtime verification is able to operate on concrete values of system state variables, which makes it possible to collect statistical information about the program execution and use this information to assess complex quantitative properties. More expressive property languages that will allow us to fully utilize this capability are needed.

Partly due to the presence of evolution of a cyber-physical system of systems both in terms of the functionality offered and in the means available for delivering the requested functionality with appropriate performance, the focus must shift from the use of models in the design and implementation phase to the operation and maintenance phase [66]. We identify the following challenges:

- A model-based systems engineering approach is needed that does full justice to the shift from design-time to run-time engineering
- Tools for the management of models and relationships between models that allow to keep track of past, current and planned system configurations at the architectural level and provide linkage with the associated models.
- System-wide simulation techniques that allow to assess the properties of the system prior to effectuating of evolution steps (in case these can be controlled). These should also aid in detecting emerging behavior.
- Methods to detect significant evolutions in the cyber-physical system of systems in order to react timely with adapted control.

Engineering and Management of CPSoS

The engineering of cyber-physical systems of systems is a challenging activity as it has to deal with changing requirements (both functional and performance requirements may chance over time), changing available constituent systems. On top of that the cyber-physical system of systems is large and complex.

In T-AREA-SoS [48] a number of aspects have been put forward that complicate the engineering of cyber-physical systems of systems:

- Representation of constituent systems is not easy as these vary over time as well: architecture should have as little as possible impact on constituent systems
- Gain of participation in system of systems is unclear from a constituent systems perspective
- Risks of unintended behaviour
- Reluctance (by stakeholders) to make changes to constituent systems that are needed from a system of systems point of view

A large number of methods and tools have been proposed for the engineering of cyber-physical systems. Below we mention some. It is impossible to deliver a complete list.

- Modelica [67] is a relatively new language for hierarchical object oriented physical modelling. The language has been designed to allow tools to generate efficient simulation code automatically. The language supports exchange of models and the use of model libraries.
- SysML (www.sysml.org) [68] is a standardized general purpose graphical modeling language for capturing complex system descriptions in terms of structure, behavior, properties and requirements. SysML offers systems engineers several noteworthy improvements over UML, which tends to be software-centric. SysML reduces UML's software-centric restrictions and adds two new diagram types, requirement and parametric diagrams for requirements engineering and for performance analysis and quantitative analysis. SysML is able to model a wide range of systems, which may include hardware, software, information, processes, personnel, and facilities. SysML furnishes flexible allocation tables that support requirements

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

allocation, functional allocation, and structural allocation. This capability facilitates automated verification and validation and gap analysis. SysML model management constructs support models, views, and viewpoints. These constructs extend UML's capabilities and are architecturally aligned with IEEE-Std-1471-2000 (IEEE Recommended Practice for Architectural Description of Software Intensive Systems).

- MARTE (Modeling and Analysis of Real-Time and Embedded Systems, [69]) provides support for specification, design and verification/validation stages.
- Dymola (www.3ds.com) is a tool for modelling and simulation of integrated and complex systems for use within automotive, aeroscape, robotics, and other application areas. Dymola supports multi-engineering solutions for modelling and simulation. The Dymola environment uses the Open- Modelica modeling language which means that the users are free to create their own modle libraries or modify existing model libraries to better match the users needs.
- Simulink (www.mathworks.nl/products/simulink) is a block diagram environment for multidomain simulation and Model-Based Design. It supports simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.
- SCADE (http://www.esterel-technologies.com/products/scade-suite/) is a model-based development environment dedicated to critical embedded software. With native integration of the Scade language and its formal notation, SCADE Suite is an integrated design environment for critical applications spanning model-based design, simulation, verification, qualifiable/certified code generation, and interoperability with other development tools and platforms, including requirements traceability.
- Stateflow (www.mathworks.nl/products/stateflow) is an environment for modeling and simulating combinatorial and sequential decision logic based on state machines and flow charts. Stateflow allow to model how a system reacts to events, time-based conditions, and external input signals. With Stateflow you can design logic for supervisory control, task scheduling, and fault management applications. Stateflow offers state diagram animation, state activity logging, data logging, and integrated debugging for analyzing the design and detecting run-time errors, static and run-time checks for transition conflicts, cyclic problems, state inconsistencies, data-range violations, and overflow conditions

These tools are offering very useful and important functionality to support the engineering process. However they are mainly restricted to conveniently model a small number of the relevant aspects. These tools provide mainly functionality for design-time activities and seem to lack proper functionality for managing and engineering the run-time engineering aspects associated with reconfiguration and evolution. Also, keeping in mind that cyber-physical systems of systems are mostly modeled in a rich palette of different modelling formalisms (even for the same aspects), these tools currently cannot deal with this heterogeneity.

Derived from computer science and engineering, meta-modelling is the activity that describes the permitted structure to which models must adhere [70]. It entails the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for modeling a predefined class of problems. As such meta-modelling may be used to capture the specific structures of an application domain. The availability of a meta-model may be used beneficially for construction of model editors (textual and graphical), parser, and in cases where semantics are attached to the models to obtain simulation and verification tools. Meta-models also provide the right level of abstraction for defining model transformations (for example a code generator).

According to a recent state of the art in meta-modelling [71] research is needed as to what methods are appropriate and intuitive in attaching semantic information to meta-models. The expected benefits of semantic attachments are among others documentation, verification benefits, and automatic translations between tools. In [72] meta-modeling is used to relate models from different domains which is beneficial in capturing the correspondence in model elements of the different views.

CyPhy [73] is a model integration language which integrates model from different domains in a semantically sound manner that enables reasoning for correctness of models. The language is supported by the META design flow and a tool suite. The goals of the CyPhy tool suite development effort were to

- support design flow through levels of abstraction with early, incremental, and continuous analysis of designs and design spaces that enable system designers to efficiently navigate the design trade-space,
- provide system and subsystem verification at different levels of abstraction including probabilistic analysis, and
- enable semantic integration through the lifecycle; compositional design tools; design verification tools; and the generation of detailed manufacturing directives spanning machine instructions, human work instructions, and logistics flow. This semantic integration ensures a seamless and coherent design flow.

Requirements engineering of cyber-physical systems of systems is faced with the following challenges [74]:

- New requirements engineering processes, management methods, techniques and tools that can dynamically respond to continually changing requirements are needed. Emphasis will need to be placed on handling competing stakeholder demands, dynamic evolution and impact of emergent behaviors on the stability of requirements
- Effective requirements engineering methods, tools and techniques for managing emergent effects with predictable results are required.

As a step in this direction [75] proposes a content model that facilitates collaboration between stakeholders. Reported benefits of the approach are consistency and eased communication between stakeholders. In [76] it is concluded that the approaches towards security requirements engineering lack explicit support for managing the effects of software evolution.

In [77] factor in systems of systems that complicate requirements engineering are reported to be (i) scale, in terms of number of constituent systems, number of interactions, number of stakeholders, (ii) multi-domain, (iii)n varied operational context, (iv) decentralized control, (v) rapidly evolving environments (vi) multiple life-cycle phases. In the paper a requirements engineering approach for systems of systems is proposed that consists of identifying the system of systems context, identification of system-wide and constituent system goals, understanding interactions, identifying individual system capabilities and constraints, and analyzing the gap between capabilities offered and capabilities required for system-wide goals.

The systematic consideration of interactions between management and control layers in the design and validation of cyber-physical systems of systems requires a methodology for the management of models and requirements which enables engineers

- to identify and capture interdependencies between the elements on different automation layers and of different constituent systems
- to trace dependencies between requirements, models, control code, simulation results, etc.,
- to deal with changes of these artefacts during the lifetime.

Existing approaches for models and requirements management from the field of model-based design in software engineering focus on models and requirements for software, often in the sense that the models are a representation of the requirements which are then refined during the design [78]. Models of physical processes and systems usually are not considered in this domain. If they are, management most often only refers to version control, translation using exchange formats, tool integration, and similar activities [79] [24].

Consistency of the models is an issue for which support is available in terms of model-management systems such as the Design Framework (df.esi.nl) that link all design activities to concrete design artefacts and to track consistency of these artefacts in a multi-disciplinary environment. These model-management systems fall short of charting semantic relations between modelling artefacts. For meaningful analysis of the models the semantic relationships of the concepts that are exchanged over interfaces must be made unambiguously clear.

The tool Artshop [80] provides a framework for managing artefact dependencies. The analyses currently implemented in Artshop cover consistency and conformity checking of related artefacts while management operations include traceability, variant management and auditing of persisted models. Each artefact and its containing elements can be annotated with arbitrary meta-information, such as the results of test cases, comments, simulation parameters or other information.

Tool integration can be divided into either conceptual or mechanical [81]. Research efforts at the mechanical level of tool integration has resulted in standardization on middleware efforts that suffer from high maintenance overheads and poor scalability [24].

ModelCVS [24] is a system which enables tool integration through transparent transformation of models between different tools' modelling languages based on their meta-models. ModelCVS provides versioning capabilities. The integration of tools is based on ontologies to express the semantics of modeling languages. In order to specify the bridging between two meta-models ModelCVS offers the following functionalities:

- Meta-model lifting: creation of an ontology, which entails a mapping of elements in the meta-model to concepts in the ontology.
- Ontology-level integration: based on relations between concepts in an ontology relations between concept in the meta-models may be deduced.
- Derivation of bridging: provide operators that express the desired integration behavior on the meta-model level. The bridging operator may be used express semantic correspondence and translation.
- Derivation of transformation: bridging is used to obtain transformation code.

The authors report that integration of ontologies which are very heterogeneous is challenging.

In a presentation for the Modelica Conference, 2011, Peter Schwarz has provided a list of items that are needed in order to be able to efficiently simulate a heterogeneous system. In a condensed form these are

- Model interfaces for model exchange between different simulation environment based on the same modeling language. The FMI is a well-recognized example of this.
- Co-simulation for coupling simulators of different modeling languages and for embedding special simulation algorithms
- Co-existence of different modelling languages such as Modelica, VHDL-AMS, Verilog-AMS, Matlab/Simulink/Stateflow, SystemC-AMS

Research challenges are:

- A model-based systems engineering approach is needed that does full justice to the shift from design-time to run-time engineering

- Tools for the management of models and relationships between models that allow to keep track of past, current and planned system configurations at the architectural level and provide linkage with the associated models.
- System-wide simulation techniques that allow to assess the properties of the system prior to effectuating of evolution steps (in case these can be controlled). These should also aid in detecting emerging behavior.
- Methods to detect significant evolutions in the cyber-physical system of systems in order to react timely with adapted control.

Bibliography

[1] S. Engell, “Cyber-physical Systems of Systems – Definition and core research and development areas,” CPSoS, 2014.

[2] R. Paulen, S. Engell, W. Fokkink, S. Klessova, H. Thompson and D. Marron, “D1.2 Report about the first meeting of the Working Groups,” CPSoS, 2014.

[3] T. Henzinger, “The Theory of Hybrid Automata,” in *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*, 1996.

[4] R. David and H. Alla, “On Hybrid Petri Nets,” *Discrete Event DYnamic Systems: Theory and Applications,* pp. 9-40, 2001.

[5] P. Cuijpers and M. Reniers, “Hybid Process Algebra,” *Journal of Logic and Algebraic Programming,* pp. 191-245, 2005.

[6] D. van Beek, K. Man, M. Reniers, J. Rooda and R. Schiffelers, “Syntax and consistent equation semantics of hybrid Chi,” *Journal of Logic and Algebraic Programming,* pp. 129-210, 2006.

[7] D. Hristu-Varsakelis and S. Levine, Handbook of Networked and Embedded Control Systems, 2005.

[8] R. Gupta and M.-Y. Chow, “Networked Control System: Overview and Research Trends,” *IEEE Transactions on Industrial Electronics ,* vol. 57, no. 7, pp. 2527-2535, 2010.

[9] FeedNetback, “Project Final Report,” 2010.

[10] M. Branicky and S. Mattsson, “Simulation of Hybrid Systems,” 1997.

[11] P. Fishwick, Handbook of Dynamic System Modeling, Chapman and Hall /CRC Taylor and Francis Group, 2007.

[12] C. Sonntag, “Modeling, simulation and optimization envoronments,” in *Handbook of Hybrid Systems Control - Theory, Tools, Applications*, 2009, pp. 328-362.

[13] Y. Dessouky and C. Roberts, “A review and classification of combined simulation,” *Computers and Industrial Engineering,* vol. 32, no. 2, pp. 251-264, 1997.

[14] P. Mosterman, “An overiew of hybrid simulation phenomena and their support by simulation packages,” in *HSCC*, 1999.

[15] F. Breitenecker, N. Popper, G. Zauner, M. Landsiedl, M. Roessler, B. Heinzl and A. Koerner, “Simulators for physical modelling - classification and comparison of features (revision 2010),” in *EUROSIM Congress*, 2010.

[16] D. Goldsman, R. Nance and J. Wilson, “A brief history of simulation revisited,” in *Winter Simulation Conference*, 2010.

[17] C. Sonntag, D. Hendriks and D. van Beek, “D6.2.1 Conceptual Outline of the CIF Co-simulation Integration and Definition of the Standardized Interface,” HYCON2, 2012.

[18] Modelica, “A Unified Object-Oriented Language for Physical Systems Modeling - Language Specification V3.2,” 2010.

[19] M. Oh and C. Pantelidis, “A modelling and simulation language for combined lumped and distributed parameter systems,” *Computers and Chemical Engineering,* vol. 20, no. 6-7, pp. 611-633, 1996.

[20] E. Internacional, “Ecosimpro user manual,” 2008. [Online]. Available: http://www.ecosimpro.com.

[21] B. Zeigler, T. Kim and H. Praehofer , Theory of Modeling and Simulation, Academic Press, 2000.

[22] E. Huang, R. Ramamurthy and L. McGinnis, “System and Simulation Modeling using SysML,” in *Winter Simulation Conference*, 2007.

[23] K. Berkenkoetter, S. Bisanz, U. Hannemann and J. Peleska, “Executable hybrid UML and its application to train control systems,” in *Integration of Software Specification Techniques for Applications in Engineering*, 2004.

[24] G. Kappel, E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger, W. Schwinger and M. Wimmer, “On models and ontologies - A layered approach for model-based tool integration,” in *Modellierung*, 2006.

[25] J. Larsson , “A framework for implementation0-independent simulation models,” *Simulation,* vol. 82, pp. 563-579, 2006.

[26] A. Siemers, P. Fritzson and D. Fritzson, “Encapsulation in object-oriented modeling for mechanical systems simulation - comparica of Modelica and Beast,” in *Vienna International Conference on Mathematical Modelling*, 2009.

[27] D. van Beek, M. Reniers, R. Schiffelers and J. Rooda, “Foundations of a compositional interchange format for hybrid systems,” in *HSCC*, 2007.

[28] A. Al-Hammouri, M. Branicky and V. Liberatore, “Co-simulation tools for networked control systems,” in *HSCC*, 2008.

[29] A. Cervin, M. Ohlin and D. Henriksson, “Simulation of networked control systems using TrueTime,” in *International Workshop on Networked Control Systems: Tolerant to Faults*.

[30] D. Henriksson and H. Elmqvist, “Cyber-physical systems modeling and simulation with Modelica,” in *International Modelica Conference*, 2011.

[31] P. Baldwin, S. Kohli, E. Lee, X. Liu and Y. Zhao, “Modeling of sensor nets in Ptolemy II,” in *IPSN*, 2004.

[32] M. Branicky, V. Liberatore and S. Phillips, “Networked control system co-simulation for co-design,” in *American Control Conference*, 2003.

[33] J. Nutaro, P. Kuruganti, L. Miller, S. Mullen and M. Shankar, “Integrated hybrid-simulation of electric power and communications systems,” in *IEEE Power Engineering Society General Meeting*, 2007.

[34] T. Kohtamäki, M. Pohjola, J. Brand and L. Eriksson, “Piccsim toolchain -- design, simulation and automatic implementation of wireless networked control systems,” in *IEEE Conference on Networking*, 2009.

[35] T. Blochwitz, T. Neidhold, M. Otter, M. Arnold, C. Bausch, M. Monteiro, C. Clauss, S. Wolf, H. Elmqvist, J. Mauss, D. Neumerkel, J. Peetz, H. Olsson and A. Junghanns, “The functional mockup interface for tool independent exchange of simulation models,” in *Modelica Conference*, 2011.

[36] MODELISAR, “Functional mock-up interface for model exchange and co-simulation,” 2012.

[37] W. Chen, M. Huhn and P. Fritzon, “A generic FMU inteface for Modelica,” in *Workshop on Equation-Based Object-Oriented Modelign Languages and Tools*, 2011.

[38] J. Dabney and T. Harman, Mastering Simulink, Prentice Hall, 2003.

[39] IEEE Computer Society, “IEEE Standard 1516 for modeling and simulation - high level architecture (hla),” 2010.

[40] R. Alur, “Can we verify Cyber-Physical Systems?,” *Communications of the ACM,* p. 96, 2013.

[41] M. Althoff, A. Rajhans, B. Krogh, S. Yaldiz, X. Li and L. Pileggi, “Formal Verification of Phase-Locked Loops Using Reachability Analysis and Continuization,” *Communications of the ACM,* pp. 97-104, 2013.

[42] A. Platzer and J.-D. Quesel, “KeYmaera: A Hybrid Theorem Prover for Hybrid Systems (System Description),”

in *IJCAR*, 2008.

[43] A. Platzer, “Differential Dynamic Logic for Hybrid Systems,” *Journal of Automated Reasoning 41(2),* pp. 143-189, 2008.

[44] Z. Biron and P. Pisu, *Real-time supervisory controller for cyber-physical systems,* 2014.

[45] S. Engell, C. Sonntag and R. Paulen, “Roadmap: From distributed and coordinated control to the management of cyber-physical systems of systems. Deliverale D3.4.1.,” HYCON2: Highly-complex and networked control systems, 2013.

[46] S. Riverso, M. Farina and G. Ferrari-Trecate, “Plug-and-play decentralized model predictive control for hybrid systems,” *IEEE Transactions on Automatic Control 58(10),* pp. 2608-2614, 2013.

[47] S. Lucia, T. Finkler, D. Basak and S. Engell, “A new robust NMPC scheme and its application to a semi-batch reactor example,” in *IFAC Symposium ADCHEM*, 2012.

[48] V. Barot, M. Henshaw, C. Siemieniuch, M. Sinclair, S. Lim, S. Henson, J. Jamshidi and D. DeLaurentis, “SoA Report,” T-AREA-SoS, 2013.

[49] H. Kopetz, “Towards an Understanding of Emergence in a System of Systems,” 2014.

[50] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger and K. Larsen, “Contracts for Systems Design,” INRIA, 2012.

[51] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone and C. Sofronis, “Multiple viewpoint contract-based specification and design,” in *FMCO*, 2008.

[52] W. Damm, E. Thaden, I. Stierand, T. Peikenkamp and H. Hungar, “Using contract-bases component specifications for virtual integration and architecture design,” in *DATE*, 2011.

[53] P. Bhaduri and I. Stierand, “A proposal for real-time interfaces in speeds,” in *DATE*, 2010.

[54] B. Delahaye, B. Caillaud and A. Legay, “Probabilistic contracts: a compositional reasoning methodology for the design of systems with stochastic and/or non-deterministic aspects,” *Formal Methods in System Design,* 2011.

[55] A. Gomes, T. Batista, A. Joolia and G. Coulson, “Architecting Dynamic Reconfiguration in Dependable Systems,” in *LNCS 4615*, 2007.

[56] J. Almeida, M. Wegdam, L. Pires and M. van Sinderen, “An approach to dynamic reconfiguration of distributed systems based on object-middleware”.

[57] AMADEOS.

[58] J. Osmundson, T. Huynh and G. Langford, “Emergent Behavior in Systems of Systems,” 2008.

[59] E. Bartocci, “Adaptive Runtime Verification,” in *RV 2012*, 2012.

[60] H. Breivold, I. Crnkovic and M. Larsson, “A systematic review of software architecture evolution research,” *Information and Software Technology 54,* pp. 16-40, 2012.

[61] T. Bureš, I. Gerostathopoulos and R. Al Ali, “DEECo: Software Engineering for Smart CPS,” *ERCIM News 97,* pp. 17-18, 2014.

[62] N. McKeown, “OpenFlow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Reviews 38(2),* pp. 69-74, 2008.

[63] M. Leucker and C. Schallhart, “A brief account of runtime verification,” *Journal of Logic and Algebraic Programming 78(5),* pp. 293-303, 2009.

[64] D. Nickovic and O. Maler, “AMT: a property-based monitoring tool for analog systems,” in *FORMATS*, 2007.

[65] A. Prasad Sistla, M. Zefran and Z. Feng, “Runtime monitoring of stochastic cuber-physical systems with hybrid state,” in *RV*, 2012.

[66] B. Schätz, “The Role of Models in Engineering of Cyber-Physical Systems - Challenges and Possibilities,” in *CPS20*, 2014.

This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

[67] P. Fritzson, Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, Wiley-IEEE Press, 2004.

[68] T. Johnson, C. Paredis and R. Burkhart, “Integrating models and simulations of continuous dynamics into SysML,” in *Modlica Conference*, 2008.

[69] OMG, “UML Profile for MARTE, v 1.0,” 2009.

[70] H. Vangheluwe and J. de Lara, “XML-based modeling and simulation: meta-models are models too,” in *Winter Simulation Conference*, 2002.

[71] J. Sprinkle, B. Rumpe, H. Vangheluwe and G. Karsai, “Metamodelling - State of the art and research challenges,” in *MBEERTS*, 2010.

[72] A. Bhave, B. Krogh, D. Garlan and B. Schmerl, *Multi-domain modeling of cyber-physical systems using architectural views,* 2011.

[73] S. Neema and J. Sztipanovits, *Multi-model language suite for cyber physical systems,* 2013.

[74] C. Ncube, “On the engineering of systems of systems: key challenges for the requirements engineering community,” in *Requirements Engineering for Systems, Services and Systems-of-Systems*, 2011.

[75] B. Penzenstadler and J. Eckhardt, “A Requirements Engineering Concept Model for Cyber-Physical Systems,” in *RESS*, 2012.

[76] A. Nhlabatsi, B. Nuseibeh and Y. Yu, “Security Requirements Engineering for Evolving Software Systems: A Survey,” vol. 1, no. 1, p. International Journal of Secure Software Engineering.

[77] G. Lewis, E. Morris, P. Place, S. Simanta and D. Smith, “Requirements engineering for systems of systems,” in *IEEE International Systems Conference*, 2009.

[78] Y. Zheng and R. Taylor, “A classification and rationalization of model-based software development,” *Software & Systems Modeling,* vol. 12, no. 4, pp. 669-678, 2013.

[79] J. Sauceda and S. Kothari, “Modern revision control and configuration management of Simulink models,” 2010.

[80] D. Merschen, J. Pott and S. Kowalewski, “Integration and Analysis of Design Artefacts in Embedded Software Development,” in *IEEE International Workshop on Tools in Embedded Systems Design Process*, 2012.

[81] W. Brown, P. Feiler and K. Wallnau, “Past and future models of CASE integration,” in *International Workshop on Computer-Aided Software Engineering*, 1992.

[82] E. Clarke, B. Krogh, A. Platzer and R. Rajkumar, “Analysis and Verification Challenges for Cyber-Physical Transportation Systems”.

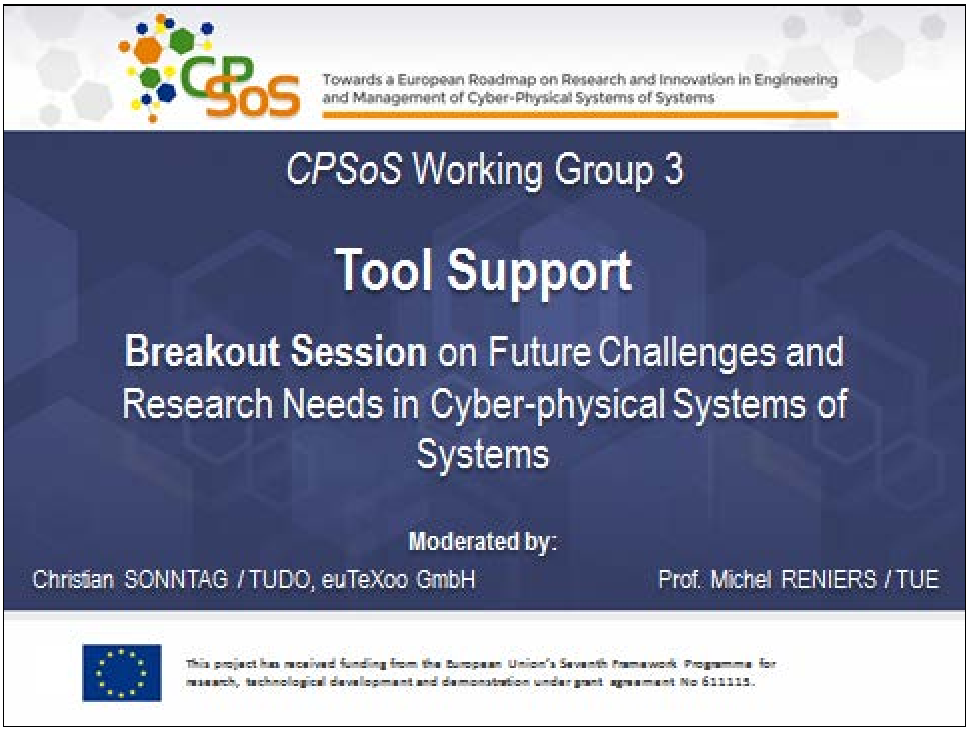
[83] J. Fitzgerald, P. Gorm Larsen and M. Verhoef, “Linking Design Disciplines in Cyber-Physical System Development: The DESTECS/Crescendo Technolog,” *ERCIM News,* vol. 97, pp. 23-24, 2014.

[84] J. Fitzgerald, P. Gorm Larsen and M. Verhoef, Collaborative Design for Embedded Systems – Co-modelling and Co-simulation., 2014.

[85] E. B. e. al., “Toward real-time simulation of cardiac dynamics,” in *CMSB'11*, 2011.

[86] J. Osmundson, N. Irvine, G. Schacher, J. Jensen, G. Langford, T. Huynh and R. Kimmel, “Application of systems of systems engineering methodology to study of joint military systems interoperability,” in *System of Systems Engineering Conference*, 2006.

4.7 Presentation for Breakout Session in Zürich



Towards a European Roadmap on Research and Innovation in Engineering and Management of Cyber-Physical Systems of Systems

CPSoS Working Group 3

Tool Support

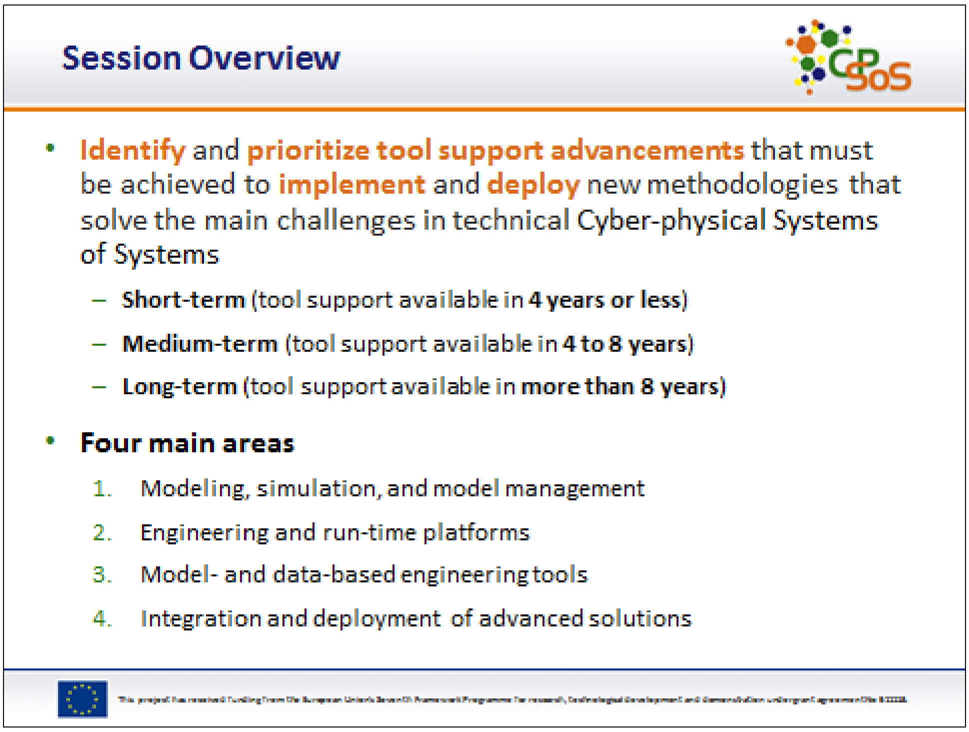
Breakout Session on Future Challenges and Research Needs in Cyber-physical Systems of Systems

Moderated by:

Christian SONNTAG / TUDO, euTeXoo GmbH

Prof. Michel RENIERS / TUE

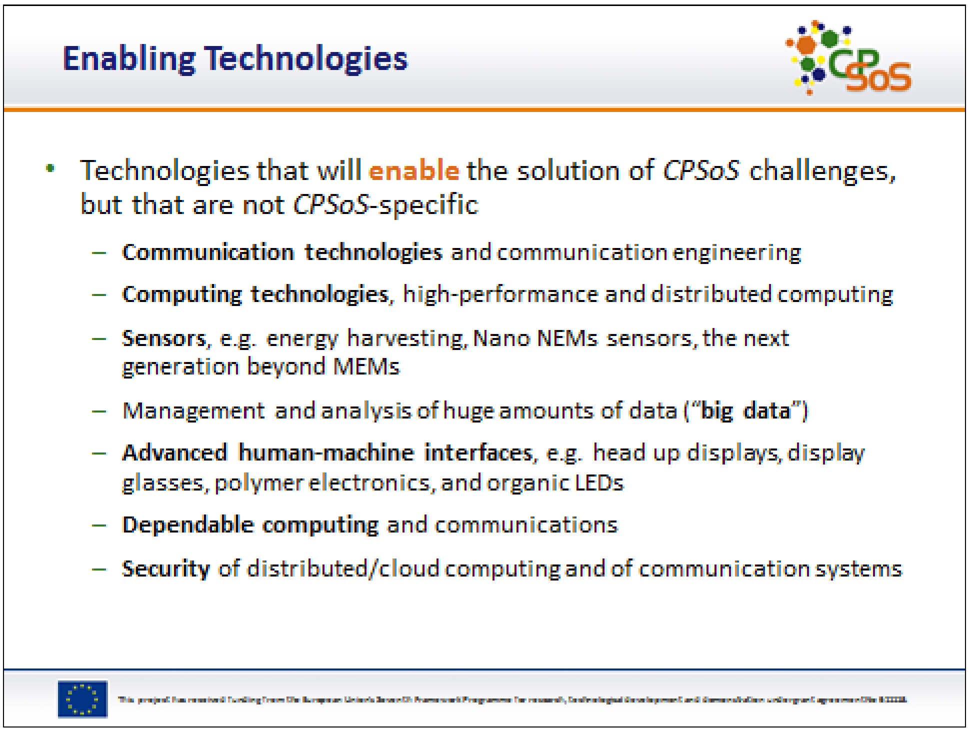
This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



Session Overview

- **Identify** and **prioritize tool support advancements** that must be achieved to **implement** and **deploy** new methodologies that solve the main challenges in technical Cyber-physical Systems of Systems
  - **Short-term** (tool support available in **4 years or less**)
  - **Medium-term** (tool support available in **4 to 8 years**)
  - **Long-term** (tool support available in **more than 8 years**)
- **Four main areas**
  1. Modeling, simulation, and model management
  2. Engineering and run-time platforms
  3. Model- and data-based engineering tools
  4. Integration and deployment of advanced solutions

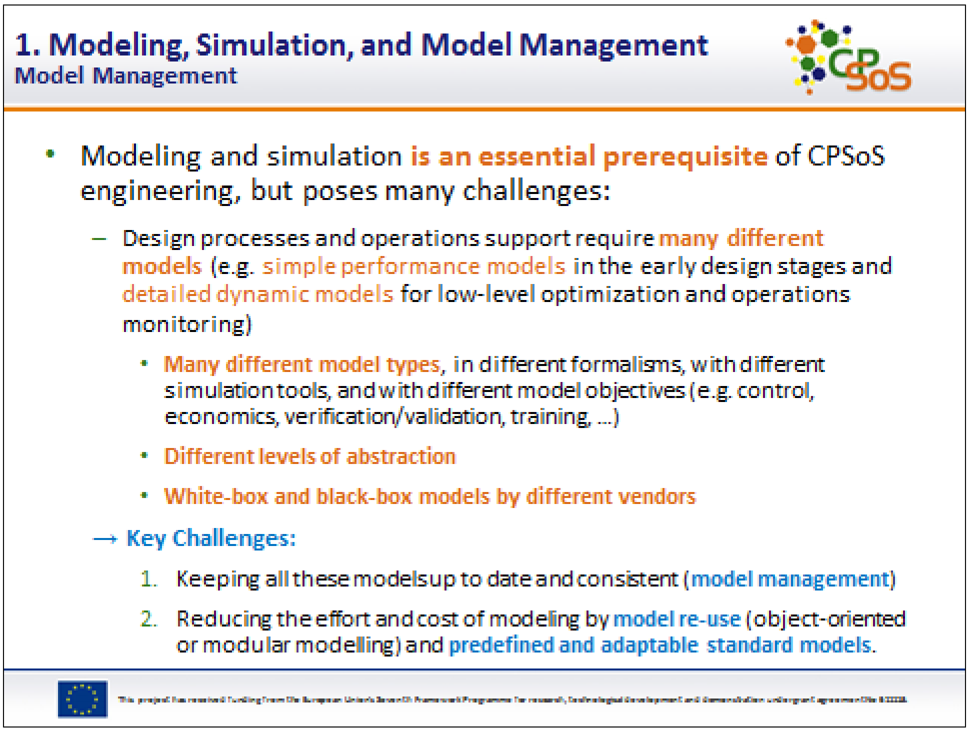
This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



Enabling Technologies

- Technologies that will **enable** the solution of *CPSoS* challenges, but that are not *CPSoS*-specific
  - **Communication technologies** and communication engineering
  - **Computing technologies**, high-performance and distributed computing
  - **Sensors**, e.g. energy harvesting, Nano NEMs sensors, the next generation beyond MEMs
  - Management and analysis of huge amounts of data (“**big data**”)
  - **Advanced human-machine interfaces**, e.g. head up displays, display glasses, polymer electronics, and organic LEDs
  - **Dependable computing** and communications
  - **Security** of distributed/cloud computing and of communication systems

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



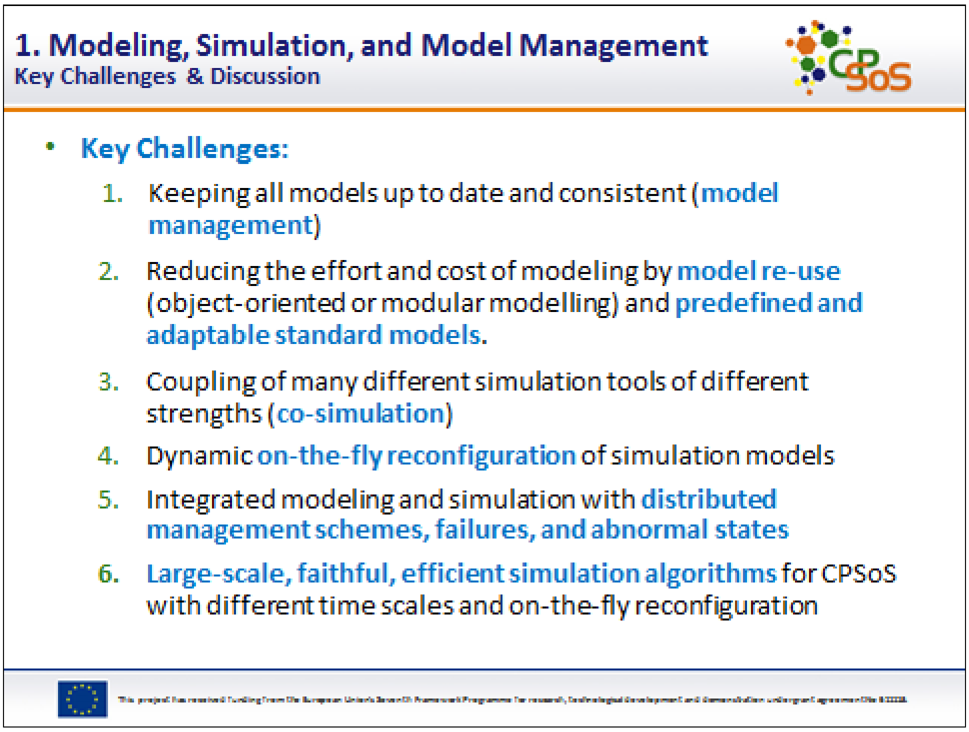
1. Modeling, Simulation, and Model Management
Model Management

- Modeling and simulation **is an essential prerequisite** of CPSoS engineering, but poses many challenges:
  - Design processes and operations support require **many different models** (e.g. simple performance models in the early design stages and detailed dynamic models for low-level optimization and operations monitoring)
    - **Many different model types**, in different formalisms, with different simulation tools, and with different model objectives (e.g. control, economics, verification/validation, training, …)
    - **Different levels of abstraction**
    - **White-box and black-box models by different vendors**

→ **Key Challenges:**

1. Keeping all these models up to date and consistent (**model management**)
2. Reducing the effort and cost of modeling by **model re-use** (object-oriented or modular modelling) and **predefined and adaptable standard models**.

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

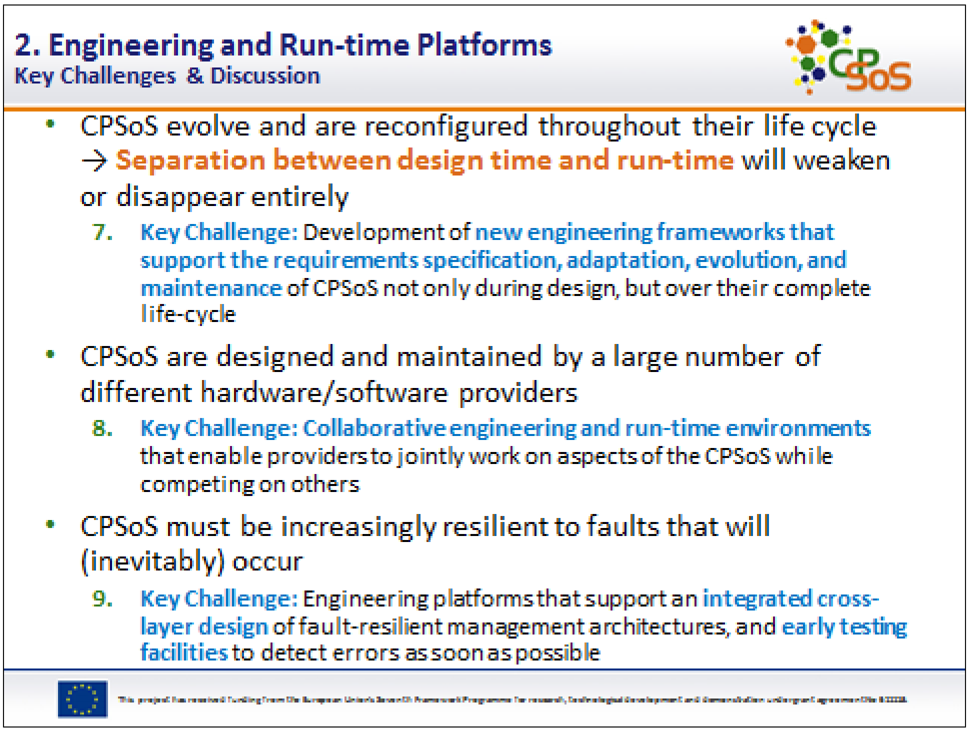


1. Modeling, Simulation, and Model Management

Key Challenges & Discussion

- **Key Challenges:**
  1. Keeping all models up to date and consistent (**model management**)
  2. Reducing the effort and cost of modeling by **model re-use** (object-oriented or modular modelling) and **predefined and adaptable standard models**.
  3. Coupling of many different simulation tools of different strengths (**co-simulation**)
  4. Dynamic **on-the-fly reconfiguration** of simulation models
  5. Integrated modeling and simulation with **distributed management schemes, failures, and abnormal states**
  6. **Large-scale, faithful, efficient simulation algorithms** for CPSoS with different time scales and on-the-fly reconfiguration

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

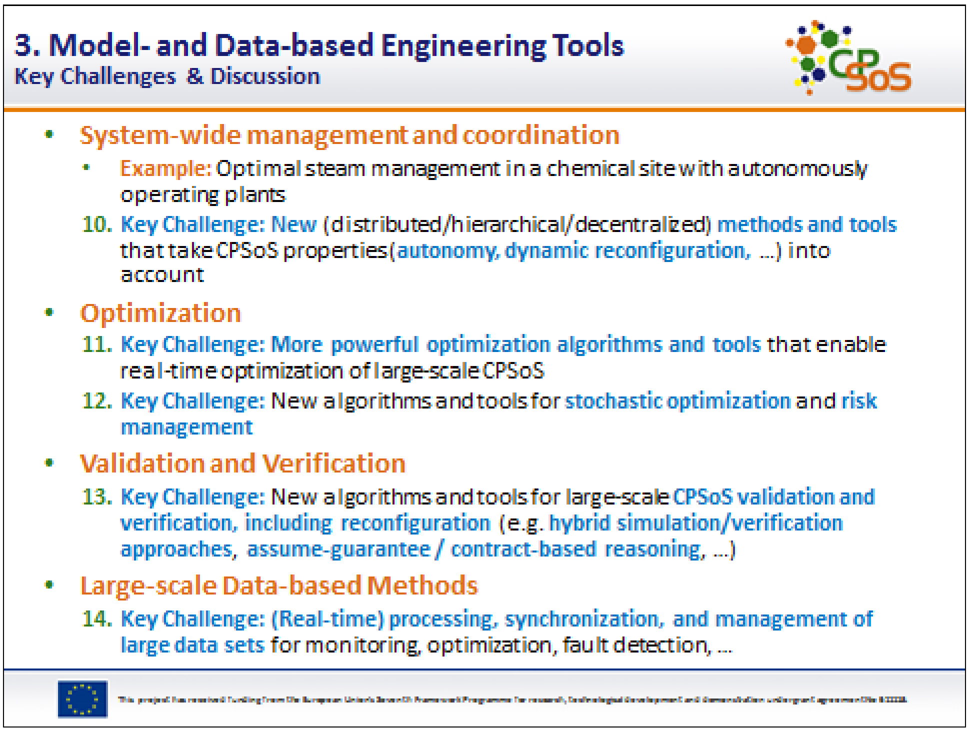


2. Engineering and Run-time Platforms

Key Challenges & Discussion

- CPSoS evolve and are reconfigured throughout their life cycle → **Separation between design time and run-time** will weaken or disappear entirely
  7. **Key Challenge:** Development of **new engineering frameworks that support the requirements specification, adaptation, evolution, and maintenance** of CPSoS not only during design, but over their complete life-cycle
- CPSoS are designed and maintained by a large number of different hardware/software providers
  8. **Key Challenge: Collaborative engineering and run-time environments** that enable providers to jointly work on aspects of the CPSoS while competing on others
- CPSoS must be increasingly resilient to faults that will (inevitably) occur
  9. **Key Challenge:** Engineering platforms that support an **integrated cross-layer design** of fault-resilient management architectures, and **early testing facilities** to detect errors as soon as possible

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



3. Model- and Data-based Engineering Tools

Key Challenges & Discussion

- System-wide management and coordination
  - Example: Optimal steam management in a chemical site with autonomously operating plants

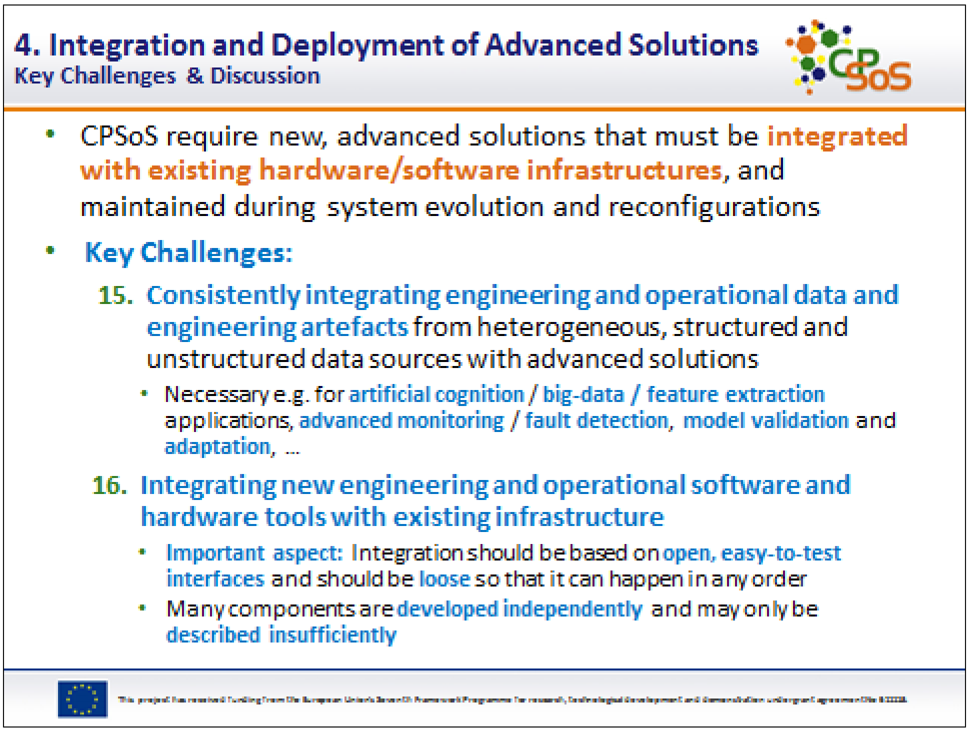
  10. Key Challenge: New (distributed/hierarchical/decentralized) methods and tools that take CPSoS properties (autonomy, dynamic reconfiguration, …) into account
- Optimization

  11. Key Challenge: More powerful optimization algorithms and tools that enable real-time optimization of large-scale CPSoS
  12. Key Challenge: New algorithms and tools for stochastic optimization and risk management
- Validation and Verification

  13. Key Challenge: New algorithms and tools for large-scale CPSoS validation and verification, including reconfiguration (e.g. hybrid simulation/verification approaches, assume-guarantee / contract-based reasoning, …)
- Large-scale Data-based Methods

  14. Key Challenge: (Real-time) processing, synchronization, and management of large data sets for monitoring, optimization, fault detection, …

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



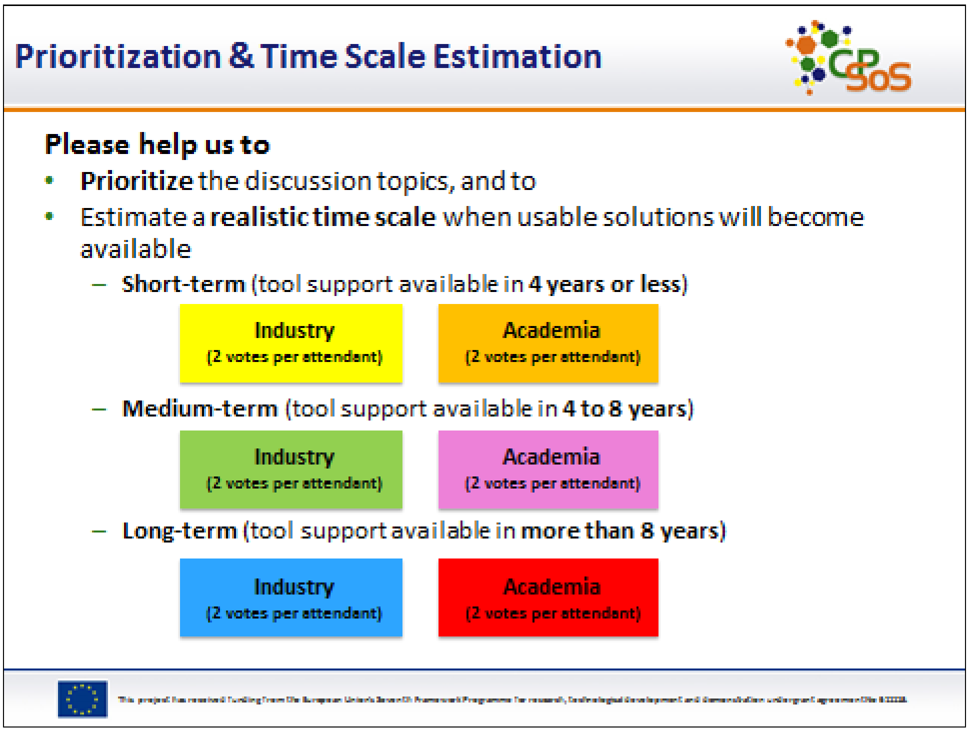
4. Integration and Deployment of Advanced Solutions

Key Challenges & Discussion

- CPSoS require new, advanced solutions that must be integrated with existing hardware/software infrastructures, and maintained during system evolution and reconfigurations
- Key Challenges:

  15. Consistently integrating engineering and operational data and engineering artefacts from heterogeneous, structured and unstructured data sources with advanced solutions
      - Necessary e.g. for artificial cognition / big-data / feature extraction applications, advanced monitoring / fault detection, model validation and adaptation, …
  16. Integrating new engineering and operational software and hardware tools with existing infrastructure
      - Important aspect: Integration should be based on open, easy-to-test interfaces and should be loose so that it can happen in any order
      - Many components are developed independently and may only be described insufficiently

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.



Prioritization & Time Scale Estimation

Please help us to

- **Prioritize** the discussion topics, and to
- Estimate a **realistic time scale** when usable solutions will become available
  - **Short-term** (tool support available in **4 years or less**)

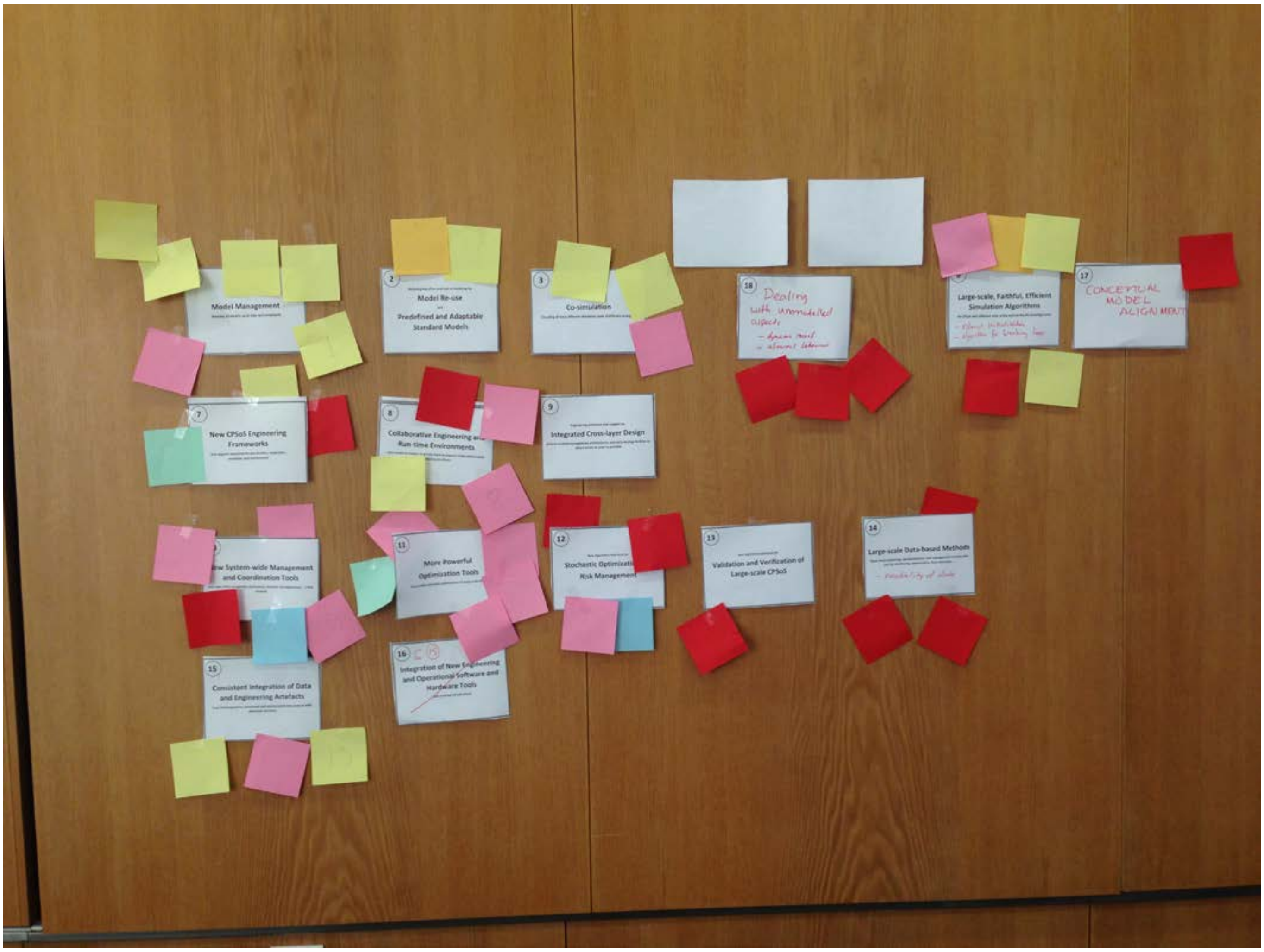
    Industry (2 votes per attendant) | Academia (2 votes per attendant)
  - **Medium-term** (tool support available in **4 to 8 years**)

    Industry (2 votes per attendant) | Academia (2 votes per attendant)
  - **Long-term** (tool support available in **more than 8 years**)

    Industry (2 votes per attendant) | Academia (2 votes per attendant)

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.

4.8 Picture of the Prioritization Results


Model Management
2
Model Re-use
Predefined and Adaptable Standard Models
3
Co-simulation
18
Dealing with unmodelled aspects
Large-scale, Faithful, Efficient Simulation Algorithms
17
CONCEPTUAL MODEL ALIGNMENT
7
New CPSoS Engineering Frameworks
8
Collaborative Engineering and Run-time Environments
9
Integrated Cross-layer Design
New System-wide Management and Coordination Tools
11
More Powerful Optimization Tools
12
Stochastic Optimization Risk Management
13
Validation and Verification of Large-scale CPSoS
14
Large-scale Data-based Methods
15
Consistent Integration of Data and Engineering Artefacts
16
Integration of New Engineering and Operational Software and Hardware Tools

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No 611115.