2.3

Cross-Sectional Technologies

ARCHITECTURE AND DESIGN: METHODS AND TOOLS

2.3. ARCHITECTURE AND DESIGN: METHODS AND TOOLS

2.3.1. SCOPE

To strengthen European industry's potential to transform new concepts and ideas cost- and effort-effectively into high- value and high-quality electronic components and systems (ECS)-based innovations and applications, two assets are essential: effective architectures and platforms at all levels of the design hierarchy; and structured and well-adapted design methods and development approaches supported by efficient engineering tools, design libraries and frameworks. These assets are key enablers to produce ECS-based innovations that are: (i) beneficial for society; (ii) accepted and trusted by end-users; and thus (iii) successful in the market.

Future ECS-based systems will be intelligent (using intelligence embedded in components), highly automated up to fully autonomous, and evolvable (meaning their implementation and behaviour will change over their lifetime), cf. Part 3. Such systems will be connected to, and communicate with, each other and the cloud, often as part of an integration platform or a system-of-system (SoS, cf. Chapter 1.4). Their functionality will largely be realised in software (cf. Chapter 1.3) running on high-performance specialised or general-purpose hardware modules and components (cf. Chapter 1.2), utilising novel semiconductor devices and technologies (cf. Chapter 1.1). This Chapter describes needed innovations, advancements and extensions in architectures, design processes and methods, and in corresponding tools and frameworks, that are enabling engineers to design and build such future ECS-based applications with the desired quality properties (i.e. safety, reliability, cybersecurity and trustworthiness, see also Chapter 2.4, in which these quality requirements are handled from a design hierarchy point of view, whereas here a process oriented view is taken). The technologies presented here are therefore essential for creating innovations in all application domains (cf. Part 3); they cover all levels of the technology stack (cf. Part 1), and enable efficient usage of all cross-cutting technologies (cf. Part 2).

Traditionally, there is a huge variety of design processes and methods used in industry, such as processes based on the V-Model in systems and software design, based on Gajsky and Kuhn's diagram (Y-chart) in hardware design, based on the waterfall model or any other kind of (semi-) formal process definition (see Figure 1).

DRAFT

SIMPLIFIED EXAMPLES OF APPLIED "TRADITIONAL" DESIGN PROCESSES

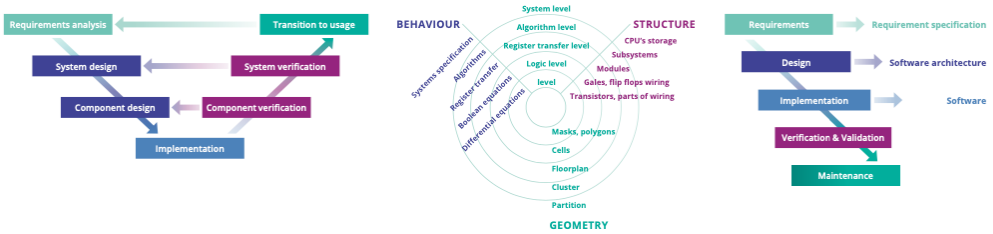

Figure 1 Simplified examples of applied "traditional" design processes: V-Model, Gajsky–Kuhn diagram (Y-chart) and the waterfall model. These are heavily in use, but not sufficient to handle future ECS-based systems and products.

Adding to the variety of design processes in use, the practical instantiation of these processes differs between companies, and sometimes even between different engineering teams within the same company. Nonetheless, most of these processes and their variants have common properties. They comprise several steps that divide the numerous design, implementation, analysis, and validation/verification tasks into smaller parts, which are then processed sequentially and with iterations and loops for optimisation. These steps include: activities and decisions on requirements elicitation and management; technologies used; system architecture; system decomposition into subsystems, components and modules; hardware/software partitioning and mapping; implementation and integration; and validation and testing on all levels of the design hierarchy.

DRAFT

CONTINOUS DEVELOPMENT AND INTEGRATION (DevOps)

Figure 2 Simplified examples for continuous development processes (DevOps processes), Such processes are essential for building future ECS-based systems and products since they enable data collected during the operation phase to be used in iterative (continuous) development for updates of existing products.

Due to the sheer size and complexity of current and future ECS-based products, the amount of functionality they perform, and the number and diversity of subsystems, modules and components they comprise, managing complexity and diversity have always been crucial in these processes. The trend of further growing complexity and diversity in future ECS-based applications increases the corresponding challenges, especially in employing model-based and model driven design approaches, and in divide-and-conquer based approaches, both on a technical level – where modular, hierarchical designs need to be integrated into reference architectures and platforms –, and also on an organisational level – i.e. by employing open

source solutions like e.g., RISC V (cf. XXX) or the various open-source integration platforms (cf. Chapter 1.3 and 1.4), to increase interoperability and thus cooperation.

A further commonality in the different design processes in use today is that almost all of them end after the complete system has been fully tested and validated (and, in some domains, been homologated/certified). Although feedback from production/manufacturing has sometimes been used to increase production quality (e.g. with run-to-run control in semiconductor fabrication), data collected during the lifetime of the system (i.e. from maintenance, or even from normal operations) is rarely taken into account. If such data is collected at all, it is typically used only for developing the next versions of the system. Again, for future ECS-based applications this will no longer be sufficient. Instead, it is vitally important to extend these processes to cover the complete lifecycle of products. This includes collecting data from system's operation, and to use this data within the process to: (i) enable continuous updates and upgrades of products; (ii) enable in-the-field tests of properties that cannot be assessed at design-, development- or testing-time; and (iii) increase the effectiveness of validation and test steps by virtual validation methods based on this data (see also Major Challenge 2 and 3 in Chapter 1.3 Embedded Software and Beyond). Apart from the technical challenges in collecting and analysing this data and/or using it for maintenance purposes, non-technical challenges include compliance to the appropriate data protection regulations and privacy concerns of system's owners (Intellectual Property) and users (privacy data). The resulting agile "continuous development processes" will ease quality properties assurance by providing design guidelines, design constraints and practical architectural patterns (e.g. for security, safety, testing), while giving engineers the flexibility and time to deliver the features that those development methodologies support ("quality-by-design solutions"). Last, but not least, the topic of virtual validation is of central importance in these continuous development processes, since both, the complexity of the system under test and the complexity and openness of the environment in which these systems are supposed to operate, are prohibitive for validation based solely upon physical tests. Although considerable advances have been made recently in scenario based testing approaches, including scenario generation, criticality measures, ODD (Operational Design Domain) definitions and coverage metrics, simulation platforms and testing methodologies, and various other topics, further significant research is needed to provide complete assurance cases as needed for certification, which combine evidences gained in virtual validation and verification with evidences generated in physical field testing to achieve the high confidence levels required for safety assurance of highly automated systems.

2.3.2. TECHNOLOGY-ENABLED SOCIETAL BENEFITS

The technologies described in this Chapter (methods and tools for developing and testing applications and their architectures) are the key enabler for European engineers to build future ECS with the desired quality properties (safety, security, reliability, trustworthiness, etc.) with an affordable effort and at affordable cost. As such, these technologies are necessary preconditions for all the achievements and societal benefits enabled by such applications.

DRAFT

ECS-based applications are becoming increasingly ubiquitous, penetrating every aspect of our lives. At the same time, they provide greater functionality, more connectivity and more autonomy. Thus, our dependency on these systems is continuously growing. It is therefore vitally important that these systems are trustworthy – i.e. that they are guaranteed to possess various quality properties (cf. Chapter 1.4). They need to be safe, so that their operation never harms humans or causes damage to human possessions or the environment; even in the case of a system malfunction, safety must be guaranteed. They also need to be secure: on the one hand, data they might collect and compute must be protected from unintended access; on the other hand, they must be able to protect the system and its functionality from access by malicious forces, which could potentially endanger safety. In addition, they must be reliable, resilient, dependable, scalable and interoperable, as well as posess many other quality properties. Most of all, these systems must be trustworthy – i.e. users, and society in general, must be enabled to trust that these systems possess all these quality properties under all possible circumstances.

Trustworthiness of ECS-based applications can only be achieved by implementing all of the following actions.

- Establishing architectures, methods and tools that enable "quality by design" approaches for future ECS-based systems (this is the objective of this Chapter). This action comprises:
    - Providing structured design processes, comprising development, integration and test methods, covering the whole system lifecycle and involving agile methods, thus easing validation and enabling engineers to sustainably build these high-quality systems.
    - Implementing these processes and methods within engineering frameworks, consisting of interoperable and seamless toolchains providing engineers the means to handle the complexity and diversity of future ECS-based systems.
    - Providing reference architectures and platforms that ensure interoperability, enable European Industries to re-use existing solutions and, most importantly, integrate solutions from different vendors into platform economies.
- Providing methodology, modelling and tool support to ensure that all relevant quality aspects (e.g. safety, security, dependability) are designed to a high level (end-to-end trustworthiness). This also involves enabling balancing trade-offs with those quality aspects within ECS parts and for the complete ECS, and ensuring their tool-supported verification and validation (V&V) at the ECS level.
- Providing methodology, modelling and tool support to enable assurance cases for quality aspects – especially safety – for AI-based systems, e.g. for systems in which some functionality is implemented using methods from Arteficial Intelligence. Although various approaches to test and validate AI-based functionality are already in place, today these typically fall short of achieving the high level of confidence required for certification of ECS. Approaches to overcome this challenge include, amongst others
    - Adding quality introspection interfaces to systems to enable engineers, authorities and end-users to inspect and understand why systems behave in a certain way in a certain situation (see "trustworthy and explainable AI" in Chapters 2.1 and 2.4), thus making AI-based and/or highly complex system

DRAFT

behaviour accessible for quality analysis to further increase user's trust in their correctness.

- Adding quality introspection techniques to AI-based algorithms – i.e., to Deep-Neural Networks (DNN) – and/or on-line evaluation of 'distance metrics' of input data with respect to test data, to enable computation of confidence levels of the output of the AI algorithm
- Extending Systems Engineering methods – i.e., assurance case generation and argumentation lines – that leverage the added introspection techniques to establish an overall safety case for the system.

The technologies described in this Chapter are thus essential to build high-quality future ECS-based systems that society trusts in. They are therefore key enablers for ECS and all the applications described in Part 3. In addition, these technologies also strengthen the competitiveness of European industry, thus sustaining and increasing jobs and wealth in Europe.

2.3.3. STRATEGIC ADVANTAGE FOR THE EU

DRAFT

Traditionally, Europe is strong in developing high-quality products. European engineers are highly skilled in systems engineering, including integration, validation and testing, thus ensuring system qualities such as safety, security, reliability, etc, for their products. Nevertheless, even in Europe industrial and academic roadmaps are delaying the advent of fully autonomous driving or explainable AI, for instance. After the initial hype, many highly ambitious objectives have had to be realigned towards more achievable goals and/or are predicting availability with significant delay. The main obstacle, and thus the reason for this technical slowdown, is that quality assurance methods for these kinds of systems are mostly not available at all, while available methods are not able to cover all the complexity of future systems. Worldwide, even in regions and countries that traditionally have taken a more hands-on approach to safety and other system qualities – e.g. a "learning-by-doing" type approach – a market introduction of such systems has failed, mainly due to non-acceptance by users after a series of accidents, with timing goals for market introduction being extended accordingly.

It has to be noted that currently there is a skill gap - a lack of skilled engineers who can drive the innovation process within the European electronics industry. Engineers having the correct skills and in sufficient numbers are crucial for Europe to compete with other regions and exploit the sector's true potential for the European economy. Universities have a vital role in the supply of graduate engineers, and it is essential that graduates have access to industry relevant design tools, leading edge technologies and training. Programs such EUROPRACTICE are essential in providing this access in an affordable manner and ensure that sufficient trained engineers are entering the European industry with relevant skills and experience.

The technologies described in this Chapter will substantially contribute to enabling European Industries to build systems with guaranteed quality properties, thus extending Europe's strength in dependable systems to trustworthy, high-quality system design ("made in

Europe" quality), contributing to European strategic autonomy. This in turn will enable Europe to sustain existing jobs and create new ones, as well as to initiate and drive corresponding standards, thus increasing competitiveness

Design frameworks, reference architectures and integration platforms developed with the technologies described in this Chapter will facilitate cooperation between many European companies, leading to new design ecosystems building on these artefacts. Integration platforms, in particular, will provide the opportunity to leverage a high number of small and medium-sized enterprises (SMEs) and larger businesses into a platform-based economy mirroring the existing highly successful platforms of, for example, Google, Apple, Amazon, etc.

In itself, the market for design, development, validation and test tools is already of considerable size, with good growth potential. The DECISION study[1], for example, has put the global market for materials and tools at €141 billion in 2018 (EU share: €24 billion), while Advancy considers the global market for equipment and tools for building ECS-based products at €110 billion in 2016 (EU share: 25%), with an estimated growth to €200 billion by 2025[2]. In addition to this existing and potential market, tools and frameworks are also key enabling technologies facilitating access to the application markets (cf., Part 3), since without them products cannot be built with the required qualities. Furthermore, the existence of cost-efficient processes implemented and supported by innovative development tools and frameworks that guarantee high-quality products typically reduces development time and costs by 20–50% (as shown by previous projects such as ENABLE S3, Arrowhead Tools, and many more). Thus, these technologies contribute substantially to European competitiveness and market access; cost-effectiveness also leads to lower pricing and therefore substantially contributes to making societal-beneficial technologies and applications accessible to everyone.

Last, but not least, the technologies described in this Chapter will contribute significantly to additional strategic goals such as the European Green Deal, while extending design processes to cover the whole lifecycle of products also enables recycling, re-using and a more circular economy.

2.3.4. MAJOR CHALLENGES

We identified four **Major Challenges** within the transversal topic "Architecture and Design: Methods and Tools". Together, these four challenges answer the need for Software tools and frameworks for engineering support for future ECS covering the whole lifecycle:

- **Major Challenge 1**: extending development processes and frameworks to handle connected, intelligent, autonomous and evolvable systems seamlessly vertically – from semiconductor-level to system-of-system-level – and horizontally – from initial requirement analysis via design, test, production, operation, maintenance, and

[1] DECISION Etudes & Conseil (Eds). "Emerging Technologies in Electronic Components and Systems (ECS) – Opportunities Ahead". A study conducted for DG-CONNECT, 2019.
[2] Advancy, 2019: Embedded Intelligence: Trends and Challenges, A study by Advancy, commissioned by ARTEMIS Industry Association. March 2019.

DRAFT

evolution (updates) to end-of-lifetime. This challenge covers necessary changes in the processes needed to develop, operate, maintain and evolve future ECS- based systems, especially their extensions to cover the whole lifecycle.

- **Major Challenge 2:** managing new functionality in safe, secure and trustworthy systems. This challenge covers methods and the corresponding tool support to ensure high-quality ECS-based systems, especially with respect to the new capabilities/functions these systems will exploit.
- **Major Challenge 3**: managing complexity. This challenge deals with methods to handle the ever-increasing complexity of ECS-based systems.
- **Major Challenge 4**: managing diversity. Handling diversity in all aspects of developing ECS-based systems is a key objective.

2.3.4.1. Major Challenge 1: extending development processes and frameworks (to handle connected, intelligent, autonomous, evolvable systems)

1.3.4.1.2 State of the art

There is currently a strict separation between the development and the operation of ECS-based systems. Data collected in any of these phases rarely "crosses the border" into the other phase (cf. section 2.3.1).

Future ECS-based systems need to be connected, intelligent, highly automated, and even autonomous and evolvable (cf. section 2.3.1). This implies a huge variety of challenges, including how to validate autonomous systems (considering that a full specification of the desired behaviour is inherently impossible), how to test them (critical situations are rare events, and the number of test cases implied by the open-world-assumption is prohibitively large), and how to ensure safety and other system quality properties (security, reliability, availability, trustworthiness, etc.) for updates and upgrades.

It is therefore necessary to overcome the "data separation barrier" and to "close the loop" by enabling systems to collect relevant data during the operation phase (Design for Monitoring) and by feeding this data back into the design phase to be used for continuous development within lifecycle-aware holistic design flows. In addition, engineering processes for future ECS-based systems should be extended to shift as much engineering effort as possible from physical to virtual engineering, and include advanced methods for systems and components design as well as new V&V methods enabling safety cases and security measures for future ECS-based systems.

2.3.4.1.2 Vision and expected outcome

The vision is to enable European engineers to extend design processes and methods to a point where they allow handling of future ECS-based systems with all their new functionalities and capabilities for the whole lifecycle. Such extended processes must retain the qualities of the existing processes: (i) efficiency, in terms of effort and costs; (ii) enable the design of trustworthy systems, meaning systems that provably possess the desired quality properties of safety, security, dependability, etc; and (iii) be transparent to engineers, who must be able to completely comprehend each process step to perform optimisations and error correction.

DRAFT

Such extended processes will cover the complete lifecycle of products, including production, maintenance, decommissioning, and recycling, thereby allowing continuous upgrades and updates of future ECS- based systems that also address the sustainability and environmental challenges (i.e. contribute to the objectives of the Green Deal). As can be derived from the timelines at the end of this Chapter, we expect supply chain-aware digital design flows enabling design for optimised manufacturing and operation (i.e. the "from design phase to operation phase direction" of the continuous design flow) for fail-aware cyber-physical systems (CPS), where selected data from operations is analysed and used in the creation of updates, by 2027. This will be completed through seamless and continuous development processes, including automated digital data flow based on digital twins and AI-based data analysis – online in the system, edge or cloud (run-time digital twin) and offline at the system's designer – as well as data collection at run-time in fail-operational CPS (i.e. "from operation to design phase" direction), by online validation, and by safe and secure deployment, by 2030.

These extended processes also require efficient and consistent methods in each of their phases to handle the new capabilities of future ECS-based systems (cf. Major Challenge 2), as well as their complexity (cf. Major Challenge 3) and diversity (cf. Major Challenge 4).

DRAFT

3.3.4.1.2 Key focus areas

There are four key focus areas in this challenge.

Virtual engineering of ECS

Design processes for ECS must be expanded to enable virtual engineering on all hierarchy levels (i.e. from transistor level "deep down", up to complete systems and even System of Systems, cf. "Efficient engineering of embedded software" in Chapter 1.3 and "Systems of Embedded and Cyber-Physical Systems - SoECPS - engineering" in Chapter 1.4 for more details of this software-focused challenge, especially with respect to SoS).

Central to this approach are "digital twins", which capture all necessary behavioural, logical and physical properties of the system under design in a way that can be analysed and tested (i.e. by formal, AI-based or simulation based methods cf. Chapter 2.1). This allows for optimisation and automatic synthesis (see also Major Challenge 1 and 2 in Chapter 2.4, and the key focus area "Modelling" in Major Challenge 2 of this Chapter) – for example, of AI-supported, data-driven methods to derive (model) digital twins.

Supporting methods include techniques to visualise V&V and test efforts (including their progress), as well as sensitivity analysis and robustness test methods for different parameters and configurations of the ECS under design. Test management within such virtual engineering processes must be extended to cover all layers of the design hierarchy, and be able to combine virtual (i.e. digital twin and simulation-based) and physical testing (for final integration tests, as well as for testing simulation accuracy).

To substantially reduce design effort and costs, a second set of supporting methods deals with the automated generation of design artefacts such as identification and synthesis of

design models, automatic scenario, use-case and test vector generation, generative design techniques, design space exploration, etc. Typically, these build upon AI-supported analysis of field data.

Last, but not least, virtual validation and testing methods must be enhanced considerably in order to achieve a level of realism and accuracy (i.e., conformity to the physical world) that enables their use in safety assurance cases and thus fully enables the shift from physical testing to virtual testing that is needed to cope with the number of test scenarios required. This includes overcoming limitations in realism of models (c.f. Challenge 2) and simulation accuracy, as caused for example by sensor phenomenology, vehicle imperfections like worn components, localisation and unlimited diversity in traffic interactions.

System and component *design* (methods and *tools*)

To fully enable virtual engineering, design processes have to switch completely to model-based processes (including support for legacy components, i.e. 'black box modelling'), where those models may be constructed using data-driven methods. Models are needed for the system and all its components on every level of the design hierarchy, especially for sensors and actuators, as well as the environment of the ECS under design, including humans and their behaviour when interacting with the system. Model-based design will also enable: (i) modular and updateable designs that can be analysed, tested and validated both virtually (by formal methods, simulation, etc) and physically; and (ii) consistent integration of all components on all levels of the design hierarchy to allow application-aware HW/SW co-design.

Such processes must be implemented by seamless design and development frameworks comprising interoperable, highly automated yet comprehensible tools for design, implementation, validation and test on all levels of the design hierarchy, including support for design space exploration, variability, analysis, formal methods and simulation.

Lifecycle-aware holistic *design* flows

"Closing the loop" – i.e. collecting relevant data in the operation phase, analysing it (using AI-based or other methods) and feeding it back into the development phase (using digital twins, for example) – is the focus of this research topic. It is closely related to the major challenges "Continuous integration and deployment" and "Lifecycle management" in Chapter 1.3, which examines the software part of ECS, and Major Challenges 1 and 2 in Chapter 2.4.

Closing the loop includes data collected during operation of the system on all levels of the hierarchy, from new forms of misuse and cyber-attacks or previously unknown use cases and scenarios at the system level, to malfunctions or erroneous behaviour of individual components or modules. Analysing this data leads to design optimisations and development of updates, eliminating such errors or implementing extended functionality to cover "unknowns" and "incidents".

Data on physical aspects of the ECS must also be collected and analysed. This includes design for optimised manufacturing and deployment, awareness of physical effects and

DRAFT

interferences, consideration of end-of- life (EOL) of a product and recycling options within a circular economy.

All of these aspects must be supported by new approaches for multi-level modelling, analysis, verification and formalisation of ECS's operational reliability and service life (c.f. previous challenges), including a consequent usage of open (and inner) source in HW and SW for the complete product lifecycle. As non- (or partly-) technical Challenges, all data collection activities described in this Chapter also need to comply to privacy regulations (e.g., the General Data Protection Regulations GDPR of the EU) as well as in a way that protects the Intellectual Property (IP) of the producers of the systems and their components.

Integration of new *V&V* methods

The required changes of current design processes identified above, as well as the need to handle the new systems capabilities, also imply an extension of current V&V and test methods. First, safety cases for autonomous systems need to rely on an operational design domain (ODD) definition – i.e. characterisation of the use cases in which the system should be operated, as well as a set of scenarios (specific situations that the system might encounter during operation) against which the system has actually been tested. It is inherently impossible for an ODD to cover everything that might happen in the real world; similarly, it is extremely difficult to show that a set of scenarios cover an ODD completely. Autonomous systems must be able to detect during operation whether they are still working within their ODDs, and within scenarios equivalent to the tested ones. V&V methods have to be expanded to show correctness of this detection. Unknown or new scenarios must be reported by the system as part of the data collection needed for continuous development. The same reasoning holds for security V&V: attacks – regardless of whether they are successful or not – need to be detected, mitigated, and reported on. cf. Chapter 1.4 and Chapter 2.4)

Second, the need to update and upgrade future ECS-based systems implies the need to be able to validate and test those updates for systems that are already in the field. Again, corresponding safety cases have to rely on V&V methods that will be applied partly at design-time and partly at run-time, thereby including these techniques into continuous development processes and frameworks. For both of these challenges, energy- and resource-efficient test and monitoring procedures will be required to be implemented.

Third, V&V methods must be enhanced in order to cope with AI-based ECS (i.e., systems and components, in which part of the functionality is based upon Artificial Intelligence methods). This includes, amongst others, adding quality introspection techniques to AI-based algorithms – i.e., to Deep-Neural Networks (DNN) – and/or on-line evaluation of 'distance metrics' of input data with respect to test data, to enable computation of confidence levels of the output of the AI algorithm (c.f. Chapter 2.1) as well as extending Systems Engineering methods – i.e., assurance case generation and argumentation lines – that leverages the added introspection techniques to establish an overall safety case for the system.

DRAFT

2.3.4.2. Major Challenge 2: managing new functionality in safe, secure and trustworthy systems

1.3.4.2.2 State of the art

Models are abstractions that support technical processes in various forms – for instance, they help systems engineers to accelerate and improve the development process. Specific models represent different aspects of the system under development, and allow different predictions, such as on performance characteristics, temporal behaviour, costs, environmental friendliness or similar. Ideally, models should cover all of these aspects in various details, representing the best trade-off between level of detail, completeness and the limitations listed below:

- Computational complexity and execution performance: models have different levels of complexity, and therefore the computational effort for the simulation sometimes varies considerably. For system considerations, very simple models are sometimes sufficient; for detailed technical simulations, extremely complex multi-physics 3D models are often required. Assessing the needed model fidelity (abstraction resp. granularity level) for each validation task, and balancing it against the needed performance requirements, is therefore essential. To achieve the necessary performance even for high fidelity models, one solution can be the parallel processing of several simulations in the cloud on the other hand super-fast embedded computing and/or edge computing can be good solutions, too. Therefore, cloud and edge computing play an important role for this topic. (cf. Chapter 2.1)
- Effort involved in creating models: very complex physically-based 3D models require considerable effort for their creation. For behavioural models, the necessary parameters are sometimes difficult to obtain or not available at all. For models based on data, including AI-based and ML-based models, extensive data collection and analysis tasks have to be carried out. Further research is urgently needed to reduce the effort for data gathering, model creation and parameterisation.
- Interfaces and integration: often, different models from different sources are needed simultaneously in a simulation. However, these models are frequently created on different platforms, and must therefore be linked or integrated. The interfaces between the models must be further standardised, e.g. FMU, FMI, extensions thereof, or similar. Interoperable models and (open source) integration platforms are needed here; they will also require further cooperation between manufacturers and suppliers. Cloud based simulation platforms require different solutions, which must be uninterrupted and with low latencies.
- Models for software testing, simulation, verification, and for sensors: another very complex field of activity concerns the model-based testing of software in a virtual environment (including virtual hardware platforms and open source solutions like RISC V (cf. Appendix XXX)). This implies that sensors for the perception of the environment must also be modelled, resulting in further distortion of reality. The challenge here is to reproduce reality and the associated sensors as accurately as possible, including real-time simulation capacity. Standards for model creation as well as standardized metrics for quality/completeness of models with respect to validation and verification are essential.

DRAFT

For each model it is important to validate that they conform to real world aspects of the system under development, i.e., that they model reality with a sufficient, guaranteed fidelity.

2.3.4.2.2 Vision and expected outcome

Efforts supporting the generation of realistic models for the entire lifecycle of a complex cyber-physical product remain very high, as the requirements for simulation accuracy, the number of influencing parameters of interest and the depth of detail are constantly increasing over time. On the other hand, the application of the highest fidelity models throughout the development process and lifecycle of products with cyber-physical components and software in turn creates numerous opportunities to save development, operating and maintenance costs. These opportunities arise in cyber-physical components or products such as vehicles, medical devices, semiconductor components, ultra-low-power ECS or any other elements in such complex technical systems. Therefore, research on advanced model-based design, development, and V&V methods and tools for the successful creation of safe, secure and trustworthy products in Europe is of utmost importance, and should be the highest priority of the research agenda. The vision is to derive efficient and consistent methods for modelling, designing, and validating future ECS-based systems, supporting the different steps in the continuous development processes derived in Major Challenge 1 by 2026 (resp. by 2029).

3.3.4.2.2 Key focus areas

This challenge comprises the following three key focus areas.

Modelling techniques for new functionalities

Model generation includes different methods (e.g. data-driven techniques, physics- or rules-based abstraction techniques) for describing (modelling) the behaviour of safety-critical, mixed, physical and logical components on different, hierarchical system levels. Model generation finally results in model libraries that are suitable for different purposes (analysis techniques, simulation, etc). There are different aspects of the modelled artefact (of the system, component, environment, etc), such as their physical properties, their (timed) behaviour, and their functional and non-functional properties, which often are modelled with different modelling approaches using different modelling tools. In addition, due to distributed architectures (edge, cloud, IoT, multi-processor architectures, etc.), future systems will become increasingly complex in their interaction and new communication and connection technologies will emerge, which must also be modelled and simulated realistically.

For the design of ECS-based systems, models are required on all levels of the design hierarchy and with different levels of fidelity (cf. Sub-section 2.3.4.2.1 in this chapter as well as Part 1 and Appendix XXX). These should range from physically-based 3D models of individual components via simplified models for testing component interaction (c.f. 'compact models' in Chapter 2.4) to specific models for sensors and the environment, also taking into account statistical scattering from production and system changes during the service life. Latest advances in AI and ML also enable novel data-based modelling techniques that can often, and especially, deliver excellent results in combination with known physical methods (cf. Chapter

DRAFT

2.1). Multi-Scale Modelling is of highest importance here: not only do we need different abstraction levels for the same system, each captured by a model suitable for efficient simulation, analysis and test on that particular abstraction level, we also need formal relations (i.e., refinement relations) between the different abstraction layers that also allow to transfer validation and test results between layers.

Furthermore, it is important to create reusable, validated and standardised models and model libraries for system behaviour, system environment, system structure with functional and non-functional properties, SoS configurations, communication and time-based behaviour, as well as for the human being (operator, user, participant) (cf. Chapter 1.4).

Most importantly, model-based design methods, including advanced modelling and specification capabilities, supported by corresponding modelling and specification tools, are essential. The models must be applicable and executable in different simulation environments and platforms, including desktop applications, real-time applications, multi-processor architectures, edge computing- and hardware-in-the-loop (HiL) platforms as well as cloud (fog) based simulation with heterogeneous access management that includes uninterrupted, wireless and cellular connectivity with low latency.

Design and *V&V* methods for ECS evolving during lifetime (including *AI*-enabled systems)

The more complex the architecture of modern ECS systems becomes, the more difficult it is to model its components, their relevant properties and their interactions to enable the optimal design of systems. Classical system theory and modelling often reaches its limits because the effort is no longer economically feasible. AI-supported modelling can be used effectively when large amounts of data from the past or from corresponding experiments are available. Such data-driven modelling methods can be very successful when the exact behaviour of the artefact to be modelled is unknown and/or very irregular. However, the question of determining model accuracy resp. model fidelity is largely unsolved for these methods.

When AI-based functions are used in components and systems, V&V methods that assure quality properties of the system are extremely important, especially when the AI-based function is safety critical. Experience-based AI systems (including deep learning-based systems) easily reach their limits when the current operating range is outside the range of the training database. There can also be stochastic, empty areas within the defined data space, for which AI is not good at interpolating. Design methods for AI-enabled ECS must therefore take into account the entire operational domain of the system, compensate for the uncertainty of the AI method and provide additional safety mechanisms supervising the AI component (i.e. mechanisms to enable fail-aware and fail-operational behaviour). Advances in V&V methods have to be accompanied by advances in AI-Algorithms, especially those that enable a higher level of introspection and increase the analysability of these algorithms (cf. Chapter 2.1).

A further source of uncertainty results from variabilities (production tolerances, ageing effects or physical processes that cannot be described with infinite accuracy) resulting from human interaction with the system and from other effects. For determining quality properties such as safety and reliability, these effects must be taken into account throughout

DRAFT

the designs' V&V. The communication channels in distributed architectures (either in cloud/fog or multi-processor architectures) also fall within the scope of these uncertainties, which can, for example, exhibit certain delays or contain certain disturbances. These effects must also be simulated on the one hand and verified accordingly in the overall simulation and system test.

There are also structured (i.e. foreseeable at design-time) variabilities in technical systems in the form of configurable changes during their lifetime, whether through software updates, user interventions or other updates. For secure systems with structured variability, suitable SW and HW architectures, components and design methods, as well as tools for adaptive, extensible systems, are crucial. This includes (self-)monitoring, diagnosis, update mechanisms, strategies for maintaining functional and data security, and lifecycle management (including End-of-life management, sustainability and possible recycling), as well as adaptive security and certification concepts (c.f. Prognostic Health Management in Chapter 2.4)

The verification and validation of ECS-based systems can also be carried out with the help of AI-based test methods (cf. Chapter 2.1). This approach allows to benefit from already performed V&V activities and developed methods and to further enhance them substantially. At the same time, the development and application of completely new test methods is also possible, as long as there is sufficient training data available for this task.

The V&V of safety-critical systems does not end with the deployment of the system. Rather, for such systems, the continuous monitoring and safeguarding of adaptive and/or dynamic changes in the system or evolving threads is of utmost importance (cf. Chapter 1.4). Further release cycles might be triggered by problems occurring in operation and the DevOps cycles must be iterated again (e.g. via reinforcement learning).

DRAFT

Ultra-low power design methods

The potential application area for ultra-low power electronic systems is very high due to the rapidly advancing miniaturisation of electronics and semiconductors, as well as the ever-increasing connectivity enabled by it. This ranges from biological implants, home automation, the condition-monitoring of materials to location-tracking of food, goods or technical devices and machines. Digital products such as radio frequency/radio frequency identification (RF/RFID) chips, nanowires, high-frequency (HF) architectures, SW architectures or ultra-low power computers with extremely low power consumption support these trends very well. Such systems must be functional for extended periods of time with a limited amount of energy.

The ultra-low-power design methods comprise the areas of efficiency modelling and low-power optimisation with given performance profiles, as well as the design of energy-optimised computer architectures, energy-optimised software structures or special low-temperature electronics (c.f. Chapters 1.1, 1.2 and 1.3). Helpful here are system-level automatic DSE (design space exploration) approaches able to fully consider energy/power issues (e.g., dark silicon, energy/power/performance trade-offs) and techniques. The design must consider the application-specific requirements, such as the functional requirements, power demand, necessary safety level, existing communication channels, desired fault

tolerance, targeted quality level and the given energy demand and energy supply profiles, energy harvesting gains and, last but not least, the system's lifetime.

Exact modelling of the system behaviour of ultra-low power systems and components enables simulations to compare and analyse energy consumption with the application-specific requirements so that a global optimisation of the overall system is possible. Energy harvesting and the occurrence of parasitic effects, must also be taken into account.

2.3.4.3. Major Challenge 3: managing complexity

1.3.4.3.2 State of the art

The new system capabilities (intelligence, autonomy, evolvability), as well as the required system properties (safety, security, reliability, trustworthiness), each considerably increase complexity (c.f. Part 3 and Sub-section 2.3.1 above). Increasingly complex environments in which these systems are expected to operate, and the increasingly complex tasks (functionalities) that these systems need to perform in this environment, are further sources of soaring system complexity. Rising complexity leads to a dramatic upsurge in the effort of designing and testing, especially for safety-critical applications where certification is usually required. Therefore, an increased time to market and increased costs are expected, and competitiveness in engineering ECS is endangered. New and improved methods and tools are needed to handle this new complexity, to enable the development and design of complex systems to fulfil all functional and non-functional requirements, and to obtain cost-effective solutions from high productivity. Three complexity-related action areas will help to master this change:

- methods and tools to increase design efficiency.
- complexity reduction methods and tools for V&V and testing.
- methods and tools for advanced architectures.

2.3.4.3.2 Vision and expected outcome

The connection of electronics systems and the fact that these systems change in functionality over their lifetime continuously drives complexity. In the design phase of new connected highly autonomous and evolvable ECS, this complexity must be handled and analysed automatically to support engineers in generating best-in-class designs with respect to design productivity, efficiency and cost reduction. New methods and tools are needed to handle this new complexity during the design, manufacturing and operations phases. These methods and tools, handling also safety related non functional requirements, should work either automatically or be recommender-based for engineers to have the complexity under control (see also the corresponding challenges in Chapter 1.3. Embedded Software and Beyond)

Complexity increases the effort required, especially in the field of V&V of connected autonomous electronics systems, which depend on each other and alter over their lifetime (cf. Chapter 3.1). The innumerable combinations and variety of ECS must be handled and validated. To that end, new tools and methods are required to help test engineers in creating test cases automatically, analysing testability and test coverage on the fly while optimising

DRAFT

the complete test flow regarding test efficiency and cost. This should be achieved by identifying the smallest possible set of test cases sufficient to test all possible system behaviours. It is important to increase design efficiency and implement methods that speed up the design process of ECS. Methods and tools for X-in-the-loop simulation and testing must be developed, where X represents hardware, software, models, systems, or a combination of these. A key result of this major challenge will be the inclusion of complexity-reduction methods for future ECS-based systems into the design flows derived in Major Challenge 1, including seamless tool support, as well as modular architectures that support advanced computation methods (AI, advanced control), system improvements (updates), replacement and recycling by 2026. Building on these, modular and evolvable/extendable reference architectures and (hierarchical, open source-based) chips (i.e., RISC V, see Appendix XXX), modules, components, systems and platforms that support continuous system improvement, self- awareness, health and environment monitoring, and safe and secure deployment of updates, will be realised by 2029.

3.3.4.3.2 Key focus areas

Methods and tools to increase design and V&V efficiency

Design efficiency is a key factor for keeping and strengthening engineering competitiveness. Design and engineering in the virtual world using simulation techniques require increasingly efficient modelling methods of complex systems and components. Virtual design methodology will be boosted by X-in-the-loop, where X (HW, SW, models, systems) are included in the simulation process, which helps to increase accuracy and speed up multi-discipline co-simulation. This starts at the architecture and design evaluation, where real tests are implemented in a closed loop such as in the exploration process.

Complexity reduction methods and tools for V&V and testing

A second way to manage complexity is the complexity-related reduction of effort during the engineering process. Complexity generates most effort in test, and V&V, ensuring compatibility and proper behaviour in networking ECS. Consistent hierarchical design and architectures, and tool-based methods to design those architectures automatically, are needed. Advanced test methods with intelligent algorithms for test termination, as well as automated metrics for testability and diagnosis (including diagnosis during run-time), must be developed and installed. Recommender-based guidance supports where no automated processes can be used. Model-based V&V and test supported by AI techniques can help to minimise the efforts driven by complexity. Models and digital twins of ECS can also be used to calculate the test coverage and extract test cases automatically.

Methods and tools for advanced *architectures*

Complexity, and also future complexity, is mainly influenced by the architecture. Future architectures must support complex, highly connected ECS that use advanced computational methods and AI, as well as machine learning, which lead to a change of ECS over lifetime. Especially for AI (cf. Chapter 2.1), this includes support for V&V of the AI method, for shielding mechanisms and other forms of fault/uncertainty detention resp. for prevention of

DRAFT

fault propagations, and for advanced monitoring concepts, that allow deep introspection of components and modules as well as hierarchical 'flagging', merging and handling of monitoring results and detected anomalies. For this, reference architectures and platform architectures are required on all levels of the design hierarchy (for the system and SOS levels, see also the challenges "architectures for SoS", "SoS interoperability" and related challenges in Chapter 1.4 on System of Systems).

An additional focus of architecture exploration and optimisation must be architectures that ease the necessary efforts for analysis, test, V&V and certification of applications. Hierarchical, modular architectures that support a divide-and-conquer approach for the design and integration of constituent modules with respect to subsystems have the potential to reduce the demand for analysis and V&V ("correct by design" approach). As integration platforms, they have to ensure interoperability of constituent ECS. For the architecture exploration and optimisation itself, AI-based methods are needed to achieve a global optimum. Overall, holistic design approaches and tools for architectures of multi-level/multi-domain systems are the goal.

Apart from the benefits that reference architectures and platforms have at a technological level, they are also important economically. As integration platforms for solutions of different vendors, they serve as a focal point for value chain-based ecosystems. Once these ecosystems reach a certain size and market impact, the platforms can serve as the basis for corresponding "platform economies" (cf. Major Challenge "open SoCPS and SoES platforms" in Chapter 1.4).

DRAFT

2.3.4.4. Major Challenge 4: managing diversity

1.3.4.4.2 State of the art

In the ECS context, diversity is everywhere – between polarities such as analogue and digital, continuous and discrete, and virtual and physical. With the growing diversity of today's heterogeneous systems, the integration of analogue-mixed signals, sensors, micro-electromechanical systems (MEMS), actuators and power devices, transducers and storage devices is essential. Additionally, domains of physics such as mechanical, photonic and fluidic aspects have to be considered at the system level, and for embedded and distributed software. The resulting design diversity is enormous. It requires multi-objective optimisation of systems (and SoS), components and products based on heterogeneous modelling and simulation tools, which in turn drives the growing need for heterogeneous model management and analytics. Last, but not least, a multi-layered connection between the digital and physical world is needed (for real-time as well as scenario investigations). Thus, the ability to handle this diversity on any level of the design hierarchy, and anywhere it occurs, is paramount, and a wide range of applications has to be supported.

2.3.4.4.2 Vision and expected outcome

The management of diversity has been one of Europe's strengths for many years. This is not only due to European expertise in driving More-than-Moore issues, but also because of the

diversity of Europe's industrial base. Managing diversity is therefore a key competence. Research, development and innovation (R&D&I) activities in this area aim at the development of design technologies to enable the development of complex, smart and, especially, diverse systems and services. All these have to incorporate the growing heterogeneity of devices and functions, including its V&V across mixed disciplines (electrical, mechanical, thermal, magnetic, chemical and/ or optical, etc). New methods and tools are needed to handle this growing diversity during the phases of design, manufacturing and operation in an automated way. As in complexity, it is important to increase design efficiency on diversity issues in the design process of ECS. A major consequence of this challenge will be the inclusion of methods to cope with all diversity issues in future ECS-based systems, which have been introduced into the design flows derived in Major Challenge 1, including seamless tool support for engineers.

3.3.4.4.2 Key focus areas

The main R&D&I activities for this fourth major challenge are grouped into the following four key focus areas.

Multi-objective *design* and *optimisation* of components and systems

The area of multi-objective optimisation of components, systems and software running on SoS comprises integrated development processes for application-wide product engineering along the value chain (cf. Part 1 and Appendix XXX on RISC-V). It also concerns modelling, constraint management, multi-criteria, cross-domain optimisation and standardised interfaces. This includes consistent and complete co-design and the integrated simulation of integrated circuits, package and board in the application context. As well it is also about methods and tools to support multi-domain designs (electronic/electric and hydraulic, etc) and multi-paradigms (different vendors, modelling languages, etc) as well as HW/SW co-design Furthermore, it deals with advanced design space exploration and iterative design techniques, the modular design of 2.5 and 3D integrated systems and flexible substrates, and the trade-offs between performance, cost, space, power and reliability.

Modelling, *analysis*, design and test methods for *heterogeneous systems* considering properties, physical effects and constr*aint*s

The area of modelling, analysis, design, integration and testing for heterogeneous systems considering properties, physical effects and constraints comprises methods and tools for the design, modelling and integration of heterogeneous systems, as well as hierarchical methods for HW/SW hybrid modeling and co-simulation, and co- development of heterogeneous systems (including multi-scale and multi-rate modelling and simulation). Furthermore, it deals with modelling methods to consider operating conditions, statistical scattering and system changes, as well as hierarchical modelling and the early assessment of critical physical effects and properties from SoC up to the system level (cf. Chapter 1.1 and 1.2). Finally, there is a need for analysis techniques for new circuit concepts (regarding new technologies up to the system level), and special operating conditions (voltage domain check, especially for start-ups, floating node analysis, etc).

DRAFT

Automation of analogue and *integration of analogue and digital design methods*

The area of integration of analogue and digital design methods comprises metrics for testability and diagnostic efficiency, especially for analogue/mixed signal (AMS) designs, harmonisation of methodological approaches and tooling environments for analogue, RF and digital design and automation of analogue and RF design – i.e. high-level description, synthesis acceleration and physical design, modularisation and the use of standardised components (cf. Chapter 1.2).

Connecting the virtual and physical world of mixed domains in real environments

The main task in the area of connecting the virtual and physical worlds of mixed domains in real environments is an advanced analysis that considers the bi-directional connectivity of the virtual and physical world of ECS and its environment (including environmental modelling, multimodal simulation, simulation of (digital) functional and physical effects, emulation and coupling with real, potentially heterogenous, hardware, and integration of all of these into a continuous design and validation flow for heterogeneous systems, cf. Major Challenge 1 and 2 above). Furthermore, the key focus area comprises novel more-than-Moore design methods and tools, as well as models and model libraries for chemical and biological systems.

DRAFT

2.3.5. TIMELINE

These timelines can (and should) be taken from the excel file: 2022_10_03_Chapter_2.3_Timelines_2023_draft.xlsx

Major Challenge 2: Managing new functionality in safe, secure, and trustworthy systems

	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
Modelling Techniques for new functionalities										
Model creation/elicitation, modelling techniques, modelling tools, model libraries										
Techniques and tools to model behaviour, timing, functional and non-functional properties of (a) components, (b) systems, (c) environment / real world, (d) test-cases / scenarios										
Multi-scale modelling. Detailed and slow fine-grained models feed key parameters to design development models. These parameters are again used in real-time models as part of the system.										
Executable models of sensors (incl. accuracy, confidence,...)										
Design and V&V methods for ECS evolving during lifetime (incl. AI-enabled systems)										
Design methods for AI enabled ECS										
V&V of AI enabled ECS, trustable AI (incl. quality attributes like safety, security, reliability, etc, but also un-biased decisions, explainability, etc.)										
Methods and tools for online risk assessment										
Methods and tools for handling cooperation (with other CPS, with humans), incl. recognizing and acting on the perceived intent of cooperation partners										
Incremental V&V for all system qualities (safety, security, reliability, trustworthiness,...)										
V&V for safe & secure systems with structural variability										
Design methods and V&V for handling of uncertainty (in perception, in communication, in prediction, in trustworthiness of data sources, etc.										
Lifetime monitoring; secure and GDPR-compliant data collection from device to multiple stakeholders (incl. system manufacturer); support for analysis & issue identification										
Ultra-low power design methods										
Advanced design methods for ultra-low-power design, focusing on component-level as well as on system-level (most potential in system architecture and system operation)										
Design methods for (autonomous) ultra-low-power systems, taking into account application-specific requirements										
Method for comprehensive assessment and optimization of power management and power consumption										

- **Start research activities aiming at** Technology Readiness Level (**TRL**) **2-4** (applied research - validation in laboratory environment) **or higher**

- **Start research activities aiming at TRL 4-6** (validation in laboratory ennvironment - demonstration in relevant environment) **or higher**

- **Start research activities aiming at TRL 6-8** (demonstration in relevant environment - prototyping in an operational environment qualified)

Major Challenge 3: Managing Complexity

	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
Methods and tools to increase design efficiency										
XIL-testing (X-in-the-loop, with X=model, system, software, hardware,.. incl. mixed-modes										
XIL simulation techniques and tools, speed up of simulation, accuracy of simulation, multi-domain co-simulation										
Efficient modelling, test and analysis for reliable, complex systems on different abstraction levels										
Evaluation of architecture and design of the ECS SW/HW with real tests										
Complexity reduction methods and tools for V&V and Test										
Recommender-based guidance in V&V process for complex ECS systems										
Automated generation of testcases from models/digital twins of ECS systems										
Test coverage calculation by means of models and testcases (coverage-driven V&V)										
Minimizing effort for V&V based on models, AI techniques										
Advanced test methods, intelligent concepts for test termination, automated metrics/tools for testability, diagnosis, and extraction of diagnostic information										
Methods and tools for consistent, hierachical design, V&V and test										
Energy and resource efficient test procedures and equipment										
Methods and tools for advanced architectures										
Architecture exploration and optimization, including multi-aspect optimization (e.g. safety, security, comfort, functionality,...), also (incl. AI based optimization methods)										
Architectures supporting advanced computation methods (AI, advanced control,...)										
Architectures and tools for non von-Neumann and neuromorphic computing										
Architectures supporting self-awareness, health and environment monitoring on all levels of the design hierarchy										
Platform and middleware architectures, also for extremely distributed, multi-layered SoS and IoT applications										
Reference architectures for continuous system improvement, i.e. across evolving system generations										
Architectures for V&Architectures for V&V and certification, including automatic evaluation of computation and deployment decisions (i.e., on chip, edge, fog, cloud).										
Modular and evolvable/extendable architectures (supporting traceability of evolution, also supporting modular updates, replacement and recycling for a circular economy										
(SW-HW) architecture mapping (incl. resource mapping and tracing (communication, scheduling, ...), incl. requirement matching and tracing										

- **Start research activities aiming at** Technology Readiness Level (**TRL**) **2-4** (applied research - validation in laboratory environment) **or higher**

- **Start research activities aiming at TRL 4-6** (validation in laboratory ennvironment - demonstration in relevant environment) **or higher**

- **Start research activities aiming at TRL 6-8** (demonstration in relevant environment - prototyping in an operational environment qualified)

Major Challenge 4: Managing Diversity

	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
Multi-objective design & optimisation of components and systems										
Consistent & complete Co-Design & integrated simulation of IC, package and board in the application context										
Methods and Tools to support multi-domain designs (i.e., electronic/electric and hydraulic, ...) and multi paradigm designs (different vendors, modelling languages, ...)										
Advanced Design Space Exploration and iterative Design techniques, inl. Multi-aspect optimization (Performance vs. cost vs. space vs. power vs. reliability)										
Modular design of 2.5 and 3D integrated systems										
Modelling, analysis, design and test methods for heterogeneous systems considering properties, physical effects and constraints										
Methods and tools for design, modelling and integration of heterogeneous systems										
Hierarchical methods for hardware/software co-simulation and co-development of heterogeneous systems (multi-scale, multi-rate modelling and simulation)										
Modelling methods to take account of operating conditions, statistical scattering and system changes										
Hierarchical modelling and early assessment of critical physical effects and properties from SoC up to system level										
Analysis techniques for new circuit concepts and special operating conditions (voltage domain check, especially for start-up, floating node analysis ...)										
Automation of analogue and Integration of analogue and digital design methods										
Metrics for AMS testability and diagnostic efficiency (including V&V & test)										
Harmonization of methods and tooling environments for analogue, RF and digital design										
Automation of analogue and RF design (high-level description, synthesis acceleration and physical design, modularization, use of standardized components										
Connecting the virtual and physical world of mixed domains in real environments										
Advanced analysis considering the connection of virtual and physical world and its environment										
Novel More than Moore design methods and tools										
Models and model libraries for chemical and biological systems										

- **Start research activities aiming at** Technology Readiness Level (**TRL**) **2-4** (applied research - validation in laboratory environment) **or higher**

- **Start research activities aiming at TRL 4-6** (validation in laboratory ennvironment - demonstration in relevant environment) **or higher**

- **Start research activities aiming at TRL 6-8** (demonstration in relevant environment - prototyping in an operational environment qualified)

2.3.6. SYNERGY WITH OTHER THEMES

The processes, methods and tools addressed in this Chapter relate to all other Chapters of the ECS-SRIA. They enable the successful development, implementation, integration and testing of all applications described in Part 3 of the ECS-SRIA, cover all levels of the technology stack (as indicated in Part 1) and enable the sufficient usage of all transversal technologies described in Part 2. Thus, there is a high synergy potential to carry out joint research on these topics. This holds especially true for topics in Chapter 2.4: qualities such as safety and security described there are a driver for the technologies in this Chapter, where we describe processes, methods and tools that enable engineers to design systems guaranteed to possess the required qualities in a cost- and time-efficient way. Additionally, strong ties exist to the 'engineering challenges' both in Chapter 1.3 (on embedded software) and in Chapter 1.4 (on systems of systems). Finally, especially for our challenge 4 on managing diversity, there is a overlap – and thus an opportunity to synergistically merge key topics from this challenge – with topics from Chapter 1.2 (on components and modules).

There is also a high synergy potential with additional activities outside of the pure funded projects work: reference architectures, platforms, frameworks, interoperable toolchains and corresponding standards are excellent nuclei around which innovation ecosystems can be organised. Such ecosystems comprise large industries, SMEs, research organisations and other stakeholders. They are focused on a particular strategic value chain, certain technology or any other asset for which sustainability and continuous improvement must be ascertained. The main activities of such innovation ecosystems are, first, to bring together the respective communities, implement knowledge exchange and establish pre-competitive cooperation between all members of the respective value chains. Second, they should promote the technology around which they are centred, i.e. by refining and extending the platform, providing reference implementations and making them available to the community, provide integration support, establish the standard, etc. Third, they should ensure greater education and knowledge-sharing. Fourth, they should develop those parts of the Strategic Research and Innovation Agenda and other roadmaps that are related to the respective technology, monitor the implementation of the roadmaps, and incubate new project proposals in this area.

Last, but not least, the technologies described in this Chapter are essential and necessary, but they are also to a large extent domain-agnostic, and can thus also serve as a connection point with activities in other funding programmes (for example Xecs, PENTA, ITEA and other EUREKA clusters).

DRAFT