2.3

Cross-Sectional Technologies

ARCHITECTURE AND DESIGN: METHODS AND TOOLS

2.3 Architecture and Design: Methods And Tools

2.3.1 Scope

Two assets are essential to strengthen European industry's potential to transform new concepts and ideas cost- and effort-effectively into high- value and high-quality innovations and applications based on electronic components and systems (ECS): 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. Next to these technologically induced advancements and benefits, the methods in this chapter also further the European goal of supporting sustainability (c.f. Chapter 0), both by advancing the creation of sustainable components and systems as well as supporting their creation in a sustainable way.

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).

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 when designing, implementing and testing these products. In addition, many of these systems need to fulfill high quality requirements, i.e. their performance, their usability, their dependability and in many cases also their functional safety and security need to conform to the highest standards. The trend of further growing functionality, complexity and diversity in future ECS-based applications combined with the advancement of new technologies (for example Artificial Intelligence) and system architectures (for example cloud- or edge-based architectures) further increases the corresponding challenges. Thus, engineers need to be able to create ECS-based products of increasing complexity fulfilling increasingly demanding challenges on the one hand, while still working cost- and effort-effective. In order to enable engineers to do so, design processes are in constant need to be adapted and to incorporate new methods and tools as well as completely new approaches to design and validation of ECS.

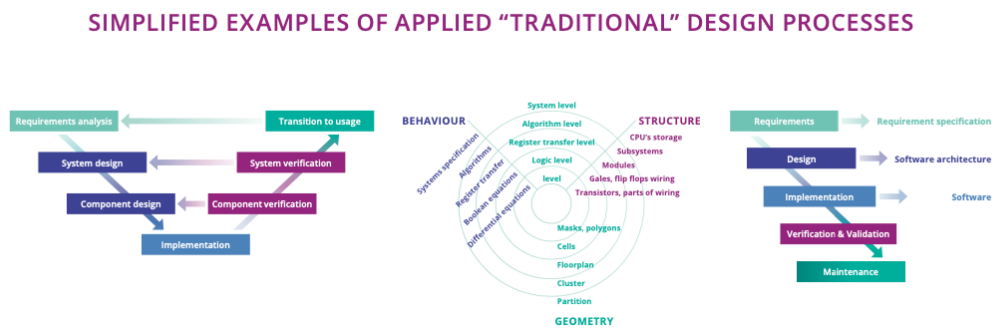

Figure 2.3.1 Simplified examples of applied "traditional" design processes: V-Model, Gajsky–Kuhn diagram (Y-chart) and the waterfall model. These are heavily in use, but not sufficient to handle future ECS-based systems and products.

One major change in design processes in recent years is, that the essentially linear traditional processes that typically end at market introduction (see Figure 1) are extended to so called continuous design processes that (a) span the whole lifecycle of a product and (b) use data collected from production, operation, and maintenance to improve on the original design. Data collected is used 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).

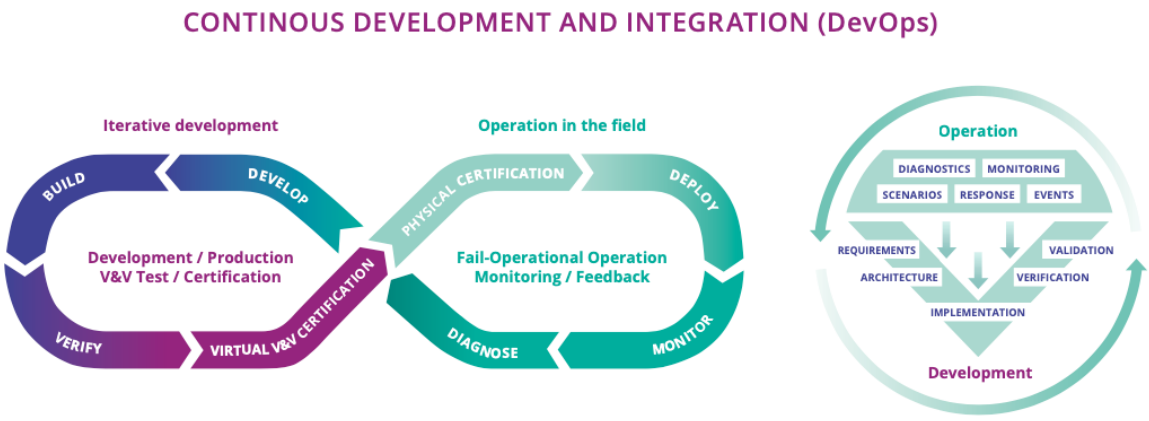

Figure 2.3.2 Simplified examples for continuous development processes (DevOps processes). Such processes are essential for building future ECS-based systems and products since they enable data collected during the operation phase to be used in iterative (continuous) development for updates of existing products.

Second, while model-based design techniques have already for some time been a major instrument to decrease both design complexity and corresponding validation effort, they now are a main enabler also for the shift to virtual engineering and virtual validation. This shift, in turn, is essential for cost- and effort-effective design and is also mirrored in current regulations for e.g. type certification. With virtual engineering, much of the design effort can be done 'virtually', i.e. by creating and using digital twins and formal models and/or simulation based test and validation methods. Additional supporting techniques are 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 organizational level – i.e. by employing open source solutions like e.g. RISC-V (cf. appendix A) or the various open-source integration platforms (cf. chapter 1.3 and 1.4), to increase interoperability and thus cooperation.

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.

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) like safety, security, dependability, reliability and similar. 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 Artificial 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 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.

2.3.2 Major Challenges

We identified four **Major Challenges** within the transversal topic "Architecture and Design: Methods and Tools".

- **Major Challenge 1**: Enabling cost- and effort-efficient Design and Validation Frameworks for High Quality ECS. The ever-increasing functionality of ECS, usage and integration of new technologies to enable these functions and the high demands for validation and testing to ensure their quality drive the need for efficient, framework- and tool-supported design and validation processes and frameworks.
- **Major Challenge 2:** Enabling Sustainable Design for Sustainability. Methods and tools to support the design and validation of sustainable ECS as well as supporting a sustainable design and validation process.
- **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 the key objective of this challenge.

Together, these four challenges answer the need for Software tools and frameworks for engineering support for sustainable high-quality ECS covering the whole lifecycle.These four challenges are highly interconnected, and can hardly be seen separately: First, advancements in one challenge often also

implies advancements in other challenges. For example, techniques to reduce the complexity of the ECS-based system from Major Challenge 3 often also result in a decrease of cost and effort for validation and test within the design and validation process (and corresponding framework), as described in Major Challenge 1. Second, solutions in one Major Challenge often also require at least partial solutions in other Major Challenges to be applicable.

2.3.2.1 Major Challenge 1: Enabling cost- and effort-efficient Design and Validation Frameworks for High-Quality ECS.

2.3.2.1.1 State of the art

Future ECS-based systems need to be connected, intelligent, highly automated, and even autonomous and evolvable. 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.

With the increased functionality of ECS and with the increasing demands on testing and validating them, engineers need tool support to enable them to design, build, test and validate, produce and maintain such complex system; these tools need to be integrated into seamless design flows (frameworks) such that the effort and the cost associated with these tasks ideally stays constant even though functionality and validation demands increase. New functionality of ECS, that cannot be handled by currently available tools, requires new methods which, when successful, lead to additional design and validation tools.

Both technology push that enables new functionalities and – even more – market pull, i.e. the demand for new functions in products – are extremely high in ECS. This is why design methods often lag behind current development methods. This holds even more for validation tools, where there a prominent examples – like showing safety for systems with functions based on AI – for which even the methods used are not complete, yet, let alone their implementation in appropriate tools. This leads to the absurd situation, that although the technologies needed for specific functions are available and even though the market demands these functions, Engineers cannot deliver them at all, or not in the required quality (i.e., safety) or not at an affordable price (because the effort, and thus the cost, for the manufacturer to build and validate these Systems is too high.

2.3.2.1.2 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 over 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.

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, see also Major Challenge 2). The main objectives to be reached here are the creation and extension of Lifecycle-aware holistic design flows that especially allow continuous development and updates of systems by collecting data from operations and maintenance (and possibly other phases of the process), feeding it back into the design phase and allowing for continuous improvements and updates. Equally important is the switch to virtual engineering of ECS, where many of the test and validation efforts required are done by simulation and analysis methods, thereby considerably increasing the efficiency of this. Advances in System Design Methods, especially in Model-Based Design, as well as in V&V methods complete the key research areas that are most important for this challenge.

2.3.2.1.3 Key focus areas

This Major Challenge comprises the following key focus areas:

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, 2 and 4 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". On the one hand, this continuous development allows for shared learning, where in a system family (like a fleet of highly automated cars) all systems can be optimized based the experience of each single member of the family. Combined with a strong monitoring concept – i.e., where unknown scenarios and unknown use cases are detected before they actually arise and where corresponding safety measures like minimum risk maneuvers or evasive actions can be triggered – this 'learning in the field' concept can achieve significant advancements in safety assurance cases of such systems (online validation and verification).

All of these aspects must be supported within holistic design flows (frameworks) that also must support:

- Supply-chain-Awareness: From requirements to optimized system architecture considering supply chain leveraging seamless digital twin from component to design to manufacturing to operation

- Complete traceability of products and processes in virtual engineering, supporting sensitivity analysis and robustness investigation, included in the optimization process and the system monitoring process
- Design for optimized manufacturing and operation; awareness of physical effects and interferences; awareness of complete lifecycle, incl. energy, resource, CO2-footprint, recycling, circular economy
- Consistent methods and new approaches for (multi-level, multi-paradigm) modeling , analysis, verification and formalization of ECS's operational reliability and service life
- Open (and inner) source in HW and SW for 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.

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 "SoS integration along the lifecycle" in chapter 1.4 for more details of this software-focused challenge, especially with respect to SoS). This requires model-based design methods including advanced modelling and specification capabilities, supported by corresponding modelling and specification tools. 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 and the following key focus area "*Advancing System and component design (methods and tools)*).

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). This allows for optimisation and automatic synthesis (see also Major Challenge 1 and 2 in chapter 2.4, 'virtual prototypes' in appendix A, – for example, of AI- supported, data-driven methods to derive (model) digital twins.

Supporting methods include techniques to visualize 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. This includes overcoming limitations in realism of models and simulation accuracy, as caused for example by sensor phenomenology, vehicle imperfections like worn components, localization and unlimited diversity in traffic interactions.

Advancing System and component design (methods and tools)

System and component design, validation and test methods have to be continuously advanced and extended in order to keep up with the new functionalities of and new technologies used for ECS. Currently, advancement in the following topics is needed urgently to realize the potential that these new functionalities and technologies can bring:

- Model based design technologies
    - Model creation/elicitation, modelling techniques, modelling tools, model libraries, using explicit models as well as data-driven models
    - (AI-based) Model identification, synthesis, improvement and parameterization with measurement data
    - 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 (physical rule based as well as data-driven).
    - Executable models of sensors (incl. accuracy, confidence,…)
- Test-Management on all hierarchy layers
    - Automatic generation of test cases on all hierarchy layers (from physical sensor input via bitvectors up to concrete test scenarios for systems)
    - Means to efficiently process & analyse dynamic test results (traces, observations , loggings,…) to derive tangible knowledge for design improvements
- Augmented and virtual reality in design, development, manufacturing and maintenance processes
- Monitoring Techniques
    - V&V extended by life-time monitoring of security and reliability aspects
    - Methods and tools for (automatically generated) monitoring of systems (based on their digital twins), including monitoring for anomaly detection (for both security and safety)

In addition, there are two research topics requiring special attention:

First is the handling of uncertainty. The higher level of automatism – up to autonomy – of systems like cars and other transportation systems, medical machinery, production plants and many others, implies that the level of uncertainty that the system has to cope with also increases and has to be handled. This is mainly the case in perception, where the system ‘observes’ its surroundings with sensors. Despite long known and used techniques like sensor fusion and similar, there is always an inherent level of uncertainty as to the accuracy of the sensor data, which leads to ‘ghost artefacts’, i.e. detection of objects that are nor there in reality, and non-detection of objects that are. The other source of uncertainty is typically in prediction, where the system needs to predict how its surroundings will evolve over time in order to adjust its own behaviour accordingly. A key research topic here is therefore the advancement

of design methods and V&V techniques for handling these uncertainties, where recent approaches implementing and guaranteeing bounds of these uncertainties and have shown first successes.

Second, the usage of Artificial Intelligence in Design and Validation promises a high potential for cost- and effort-efficiency, in a similar way as the use of AI in the systems themselves promises and increase in functionality of these systems.

- Integration of AI (including generative AI) and AI-based tools into engineering and development processes on all levels of the design hierarchy, to shorten development time, incl. metrics for quantification of covered design space, etc.
- Usage of AI and AI-based tools for V&V and development task.
- AI support for Model identification, synthesis, improvement and parameterization with measurement data (c.f. Model based design technologies above)
- Design and V&V for new technologies (i.e., flexible electronics, textile electronics…)

Usage of AI techniques in Design and Validation imposes the additional challenge that these have to be either supervised (i.e., they need to be able to explain to their user, why and how they achieved the results that they produce) or have to be validated themselves ('Who validates the validator?').

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 (compare to 'Explainable AI' in 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.

2.3.2.2 Major Challenge 2: Enabling Sustainable Design for Sustainability

2.3.2.2.1 State of the art

Sustainability is a major goal of the European ECS industry; as described in Chapter 0, there are two sustainability goals:

On the one hand, we need to produce *sustainable ECS*, i.e. products that contribute to sustainability by being highly energy efficient, by producing less or even no emissions; products, which have a long lifetime and which are built from materials that can be recycled or used in other contexts afterwards (second life).

On the other hand, we need to *produce ECS in a sustainable way*, i.e., by using less energy for design, validation and production, and by increasing the useful lifetime of systems and/or their components.

Together, these actions contribute to eight of the seventeen sustainability goals of the United Nations and the EU, namely SDG 3 Good Health and Wellbeing, SDG 6 Clean Water (and sanitation), SDG 7 Affordable and clean Energy, SDG 8 Decent work and economic growth, SDG 9 Industry, Innovation, and Infrastructure, SDG 11 Sustainable Cities and communities, SDG 12 Responsible consumption and production, and SDG 13 Climate Action.

This Major Challenge groups the most prominent Key Focus Areas contributing to these goals. It should be noted, however, that these two goals have been a driving factor for many of the advancements done in ECS Design and Validation during the last decades. Building engines that are better – stronger, more reliable, etc. – while using less fuel or other resources, heavily contributing to the Smart Grid, usage of new materials and many other activities during recent years aim to achieve sustainable ECS based products as well as a sustainable way of designing and testing them. Thus, one kind find sustainability goals behind many of the focus areas described in Major Challenges 1, 3 and 4, although there sustainability might not be the main driver, but a highly advantageous second goal of that topic area.

2.3.2.2.2 Vision and expected outcome

There is a strong technology push combined with a need raised by the increased functionality of ECS based products to switch to lifecycle aware design processes and continuous development of ECS-based products, including data collection from systems under production and from operation, as described Major Challenge 1. This switch perfectly matches the Lifecycle aware Design Optimizations described in the following section. Together, these will enable the design of ECS based products that are produced using less and less energy and other resources, which are more easily repairable and which are recyclable

or reusable at the end of their initial lifetime. Continuous Design Flows also drive the need for Updates and updatable systems, which perfectly matches the fact that Updates are also a need to extend useful lifetime of a product, thus increasing its sustainability. The two focus areas of 'Energy and resource efficient test procedures and equipment' and 'Low power design' directly contribute to the sustainability of the design and test process as well as the energy consumption of the system.

2.3.2.2.3 Key focus areas

Lifecycle aware Design Optimizations

Lifecycle aware design optimizations is a collective term for procedures and methods that aim at designing sustainable ECS. It comprises

- 'Design for producability' aiming at designing ECS that can be produced resource efficiently (i.e. with low consumption of energy and other resources'), Notice, that this topic comprises a lot more than only thinking about production materials and changing them in a sustainable way. For example, in the Electric-/Electronic Architecture of cars, the recent change from using point to point communication lines between different computational units to installing more and more communication busses (i.e. common communication lines used by many computational units) resulted not only in using less copper wires, but also in a fundamental shift in communication patterns and protocols that influenced the complete design of the application software.
- 'Design for recycling/reuse', which is mostly concerned with the materials, and
- 'Design for reparability', which comprises both, architectural methods that enable or ease reparability of a system as well as design and management of spare parts.

All of them essentially face the same challenges, namely

- Specification techniques for the corresponding requirements (of producibility, recycling/reuse and reparability)
- Requirement capturing
- Implementation techniques for these requirements
- Validation/Test methods for these requirements
- Conflict resolution techniques for requirement (i.e., how to handle a situation in which e.g., a safety requirement conflicts with a producibility requirement).

Especially for requirement capturing and validation methods, these techniques profit from data collection during production and operation, as described in Major Challenge 1 under "Lifecycle aware holistic design flows", which enables a continuous design flow also for these aspects of the system.

Updates

Updates resp. the property to be 'updateable' has the potential of significantly increasing the lifetime of a product and thus increase its sustainability. An 'Update' of an ECS is a change in the functionality

and/or in the realization (implementation) of some functionality of an ECS. Updates can be done by exchanging components or modules of a (sub-)system and by exchanging (parts) of the software running on it. The former requires physical access to the product, which can be done during regular or additional maintenance services, the later can also be done 'over the air' only. There are three reasons for updates: The first is error correction, i.e., when an hitherto unknown functional error or a security hazard is detected, it can be corrected via an update. The second is performance or usability optimization and variation. If a function in an ECS can be implemented in different ways – for example engine control in a car can be done 'smooth' or 'sportive' or in many other ways – then a change in the implementation can be realized by updates. Updates initiated by one of these two reasons already have a highly positive impact on sustainability. However, the main benefit for sustainability stems from the third reason for updates, which is adding functionality to an ECS already in use, thus avoiding replacement of the product in favour of 'newer, better versions'. The main challenges for updates are first, ensuring quality of the updated system. Taking safety as the major quality of an ECS again, it is difficult to analyse and guarantee all the existing safety properties of a system for the new, updated system, seeing that the updated system is already in operation and thus not available for physical tests anymore. The second challenge lies in the fact that there are typically many – sometimes even thousands – of variants of a product in operation, and the update has to fit all of them and quality has to be assured for all of them. The third challenges mostly concerns over-the-air software updates, for which safe and secure deployment has to be guaranteed.

Energy and resource efficient test procedures and equipment

Testing and validation of ECS used to be a very energy consuming task, with hundreds and thousands of physical tests taking place not only with the final product (or a prototype thereof), but also with all subsystems, components and modules of the system. The setup, maintenance and operation of the various test environments were also highly resource intensive. With the shift to virtual engineering including the use of digital twins and virtual testing (c.f. Major Challenge 1), many physical tests can be avoided. However, first, the number of physical tests needed is still fairly large, and reducing their energy and resource consumption is still an important topic. In addition, virtual engineering itself requires energy for all the computers, servers, and simulation suites that are needed for it. Any technique that lowers power consumption for computers, servers, and clouds is therefore also beneficial here, as are methods that reduce the number of test cases needed for showing quality attributes.

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 (see also appendix A on RISC-V). 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.

2.3.2.3 Major Challenge 3: Managing complexity

2.3.2.3.1 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 to enable efficient Safety Assurance Cases
- methods and tools to increase design efficiency.
- complexity reduction methods and tools for V&V and testing.
- methods and tools for advanced architectures.

2.3.2.3.2 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 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 A), 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.

2.3.2.3.3 Key focus areas

Assurance Cases

Safety Assurance Cases are a commonly and successfully used method to show that a certain product possesses certain qualities, e.g., safety. Corresponding arguments for other qualities (like security, dependability, etc.) are being used as well.

Safety Assurance cases consist of a set of Safety Cases or Safety Arguments. ISO 26262 states that “the purpose of a safety case is to provide a clear, comprehensive and defensible argument, supported by evidence, that an item is free from unreasonable risk when operated in an intended context’. According to the CAE principle, Safety Arguments comprise Claims, i.e. that the system fulfills a certain safety related requirement, and Arguments, i.e., an explanation – or proof -- why a claim (or sub claim) is valid. Arguments, in turn, comprise sub-claims – for which in turn an Argument is needed -- or evidence, i.e., an analysis or test results or other findings from the implemented system that support the (sub-claim).

Safety Assurance cases are still a valid means for showing Safety (or other qualities) of an ECS-based product. However, the following non-trivial extensions to the method are needed in order to be able to handle modern and future ECS-based products:

- ODD, behavior competencies: To construct safety cases, both the operational context as well as the functionality of the system needs to be specified. While there has been considerable advancement in describing operational context by ODDs (Operational Design Domains), including first standardization activities, there still is a lot of room for further improvement. Behavior Competencies, which are used to formally describe the systems functionality, are still further behind. Together these two concepts need to be extended by means to describe reduced functional behavior and minimal risk maneuvers.
- Handling of unknowns: Both the definition of the operational context as well as the functionality of the system will contain uncertainties and unknowns (c.f. Handling of Uncertainties in Major Challenge 1). To include these in the safety argument, statistical reasoning and/or three-valued logics need to be used.
- Quality metrics and Guarantees: For highly automated (up to autonomous) systems, which may have part of their functionality implemented by Artificial Intelligence, it is sometimes infeasible to collect the evidence needed to support a certain (sub-)claim. For example, in the automotive sector for autonomous driving, test driving the number of miles required is infeasible. Sometimes, on the other hand, we do not even know what quality to measure or what guaranteed performance would be evidence to support a certain (sub-)claim. Again as an example, for AI-based object detection, we cannot interfere how much training and how many tests are required to guarantee this property. The challenge here is to find quality Metrics and Guarantees that (a) are sufficiently strong to support the safety argument and (b) can be measured or analyzed with the needed efficiency.

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 the research topics:

- 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 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. This includes the following research topics:

- 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, hierarchical design, V&V and test
- Energy and resource efficient test procedures and equipment

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 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 "SoS Architecture and open integration platforms", "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 "SoS Architecture and open integration platforms" in chapter 1.4). Thus, the following research topics turn out:

- Architecture exploration and optimization, including multi-aspect optimization (e.g. safety, security, comfort, functionality,…) and 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&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

All of the above topics need to be examined and developed in conjunction with corresponding changes in design and validation tools ('Design for Target Architectures', 'Design for platforms') in order to leverage on the complexity reduction that they bring.

2.3.2.4 Major Challenge 4: Managing diversity

2.3.2.4.1 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), chiplets, 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.

2.3.2.4.2 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.

2.3.2.4.3 Key focus areas

The main R&D&I activities for this fourth major challenge are grouped into the following 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 A on RISC-V). It also concerns modelling, constraint management, multi-criteria, cross-domain optimisation and standardised interfaces. This includes the following research topics

- 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, …) and HW/SW co-design
- 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 and chiplets and flexible substrates

Modelling, analysis, design and test methods for heterogeneous systems considering properties, physical effects and constraints

The area of modelling, analysis, design, integration and testing for heterogeneous systems considering properties, physical effects and constraints comprises the following methods and tools which all need to consider chiplet technology aspects: .

- Methods and tools for design, modelling and integration of heterogeneous systems (incl. chiplet technology)
- 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

The area of automation of analogue and integration of analogue and digital design methods comprises the following research topics:

- Metrics for analogue/mixed signal (AMS) testability and diagnostic efficiency (including V&V & test)
- Harmonisation of methods and tooling environments for analogue, RF and digital design
- Automation of analogue and RF design – i.e. high-level description, synthesis acceleration and physical design, modularisation and the use of standardised components

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

The area of connecting the virtual and physical worlds of mixed domains in real environments is about the following research topics:

- Advanced analysis considering 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).
- Novel more-than-Moore design methods and tools,
- Models and model libraries for chemical and biological systems

2.3.3 Timeline

Major Challenge	Topic / Key focus area	Short Term	Medium Term	Long Term
		2025-2029	2030-2034	2035 and beyond
MC 1: Enabling cost- and effort-efficient Design and Validation Frameworks for High Quality ECS	**Topic 1.1:** Lifecycle-aware holistic design flows	Supply-chain aware design frameworks covering the complete lifecycle of ECS, including data feedback from production, operations, and maintenance supporting multi-level, multi-paradigm continuous modelling, integration, and analysis; interoperable tool chains	Supply-chain aware design frameworks allowing 'forward optimization' (Design for optimized manufacturing / operations / recycling) as well as 'backwards optimization' (continuous design, integration and test)	
	Topic 1.2: Virtual engineering of ECS	Advanced Model-Based Design and Specification Tools for the System under Development as well as for the Environment; semi-automatic generation of design artefacts; aspect-specific Digital Twins; advanced simulation and validation tools	Domain specific Digital Twins. Accurate simulation environments enhanced by automatic validation tools; automatic generation of design artefacts; accurate simulation of environment and ECS behavior	Full Digital Twins
	Topic 1.3 Advancing System and	Executable models of sensors; model based design technologies,	Uncertainty – aware, residual risk based design and analysis	Full Explainable AI support for all phases of

	component design (methods and tools)	switch to probabilistic analysis of behaviour to handle uncertainties; AI supported modelling, design space exploration, and data analysis	of perception chain and prediction engines; Explainable AI-based modelling, design space exploration and data analysis	the development, integration and test process.
	Topic 1.4 Integration of new V&V methods	Safety Cases based on ODD and system capabilities, ODD monitoring within fail-aware ECS, scenario and test case coverage of ODD; V&V for modular Updates	Full V&V for Updates based on Digital Twins. V&V for systems with shielded or guarded AI based components	V&V for systems with AI-based components; certification at run-time (for known environments and restricted updates)
MC 2: Enabling Sustainable Design for Sustainability	**Topic 2.1:** Lifecycle aware Design Optimizations	Requirement engineering for 'Design for X', specification, implementation and testing techniques; health monitoring of system components. AI supported analysis of health data	Resolution techniques for conflicting requirements; AI based analysis of health data	
	Topic 2.2: Updates	Development and secure deployment of safe updates based on selected data from the field; Variant Management for Updates, V&V for modular updates	Full V&V for variant aware Updates based on Domain-specific Digital Twins within virtual engineering frameworks	
	Topic 2.3: Energy and resource efficient test procedures and equipment	c.f. Topic 1.2 'Virtual Engineering for ECS' above as well as Major Challenge 1 of Chapter 2.1		

	Topic 2.4: Ultra-low power design methods	System level DSE tools supporting energy aware design decisions; energy aware modelling of system behaviour, energy aware simulation of Ultra-low-power systems	Increased accuracy of Modelling and simulation techniques considering fault tolerances, targeted quality level, etc.	
MC 3: Managing complexity	**Topic 3.1:** Assurance Cases	Safety Assurance Cases based on ODD and system capabilities/behaviour competencies, ODD monitoring within fail-aware ECS, scenario and test case coverage of ODD; Quality metrics, Quality guaranties	Handling of unknown scenarios and system behaviour uncertainties beyond Minimum Risk Manoeuvres; Efficient validation of quality metrics and guarantees based on statistical analysis.	Assurance Cases for fully autonomous systems
	Topic 3.2: Methods and tools to increase design and V&V efficiency	Fully hierarchical XIL testing with clearly defined relations between abstraction levels, multi-domain, multi-objective simulation; validation of simulation results by physical testing; AI Support (c.f. Topic 1.3)	Accurate and fast simulation of system behaviour and environment; minimization of the need for physical tests; AI Support (c.f. Topic 1.3)	Fully simulation based V&V using Digital Twins; AI Support (c.f. Topic 1.3)
	Topic 3.3: Complexity reduction methods and tools for V&V and testing	Automated generation of test cases from models/digital twins of ECS systems; coverage-driven V&V, AI supported V&V and testing	Recommender-based guidance in V&V process for complex ECS systems; Explainable AI-based V&V	Complexity reduction and V&V techniques fully based on explainable AI

	Topic 3.4: Methods and tools for advanced architectures	Hierarchical, modular architectures for V&V and certification supporting AI and advanced control, multi-aspect design space exploration and multi-objective optimizations for Architecture Design, Reference Architectures supporting continuous system improvement	Modular and evolvable/extendable Reference Architectures and Platforms, design tools for architectures for multi-level, multi-domain systems	Architectures and Tools for new technologies, holistic design approaches and tools for architectures for multi-level, multi-domain systems
MC 4: Managing diversity	**Topic 4.1:** Multi-objective design and optimisation of components and systems	Methods and Tools to support multi-domain and multi paradigm designs, as well as HW/SW co-design; Modular design of 2.5 and 3D integrated systems and chiplets and flexible substrates	Consistent & complete Co-Design & integrated simulation of IC, package and board in the application context; Advanced Design Space Exploration and iterative Design techniques, incl. Multi-aspect optimization	
	Topic 4.2: 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 (incl. chiplet technology); Hierarchical methods for HW/SW co-simulation and co-development of heterogeneous systems	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	
	Topic 4.3: Automation of analogue and integration of analogue and digital design methods	Harmonisation of methods and tooling environments for analogue, RF and digital design; Automation of analogue and RF design – i.e. high-level description, synthesis acceleration and physical design, modularisation and the use of standardised components	Metrics for analogue/mixed signal (AMS) testability and diagnostic efficiency (including V&V & test)	
	Topic 4.4: Connecting the virtual and physical world of mixed domains in real environments	Novel more-than-Moore design methods and tools	Advanced analysis considering the bi-directional connectivity of the virtual and physical world of ECS and its environment (cf. Major Challenge 1 and 2 above); Models and model libraries for chemical and biological systems	