NESSI

Software and Key Digital Technologies

Executive summary

This position paper is an analysis by NESSI[1] about the role of software technology in Key Digital Technologies (KDT), one of the proposed institutionalised European Partnerships under Horizon Europe (2021-2027).

Software already plays an important role in the value chain of electronic components and systems, which is the focus of the ECSEL partnership, a predecessor of KDT. Embedded software, compilers and frameworks are important parts of electronic components, and it is the development and operation of software applications that eventually brings electronic systems to the end users. Built on ECSEL, the KDT partnership aims to "reinforce Europe's innovation potential by contributing robust electronic components and systems, software technologies, sub-assemblies, and systems of systems (SoS) to complete value chains/networks, and providing secure and trusted technologies tailored to the needs of user industries"[2]. To achieve this, it plans to extend the scope to cover more research and innovation actions on the software aspects of electronic components and systems. NESSI fully supports this extension, as software technologies have proven to be a game changer for digitalisation in different industry sectors. At the same time, the way we develop and operate software is evolving significantly, powered by multiple key enabling technologies emerging recently, such as AI and ubiquitous connectivity. However, advances in software engineering are mainly within the original software industry, and are adopted only with significant delay in other industry sectors. In this paper, we analyse the important challenges that hinder the improvement of software development in the KDT context. To address the challenges and to support the area of KDT with advanced software technologies, NESSI has identified and recommend the following research and innovation topics:

- DevOps for electronic components and systems, involving all players along the KDT value chain.
- Platformization of KDT applications to support software-driven integration and to build ecosystems.
- Digital trust in systems with large number of components in a distributed and context-aware way.
- Management of complexity in the presence of dynamicity and uncertainty arising from, and addressed by, AI-powered self-adaptation.
- Spatial Computing that utilizes advanced human-machine interactions enabled by novel electronic components.
- Software support for sustainable and efficient computing.

Importance of Software Technology as a Game Changer

The world is facing another deep crisis: the COVID-19 pandemic has had a dramatic impact on many business sectors, drastically modified the lives of billions of people around the world, and forced individuals, enterprises and administrations to rely more than ever on digital services to help fight the disease and keep the economy running. Transformation will be an important factor in enabling businesses to survive. This transformation, however, must also take into account two emergent market trends that are demanded by the European civil society: sustainability (climate and the circular economy), and human-centric industrial technology (as opposed to automation technologies that minimize human involvement). These trends, already embedded within the recently published EU Green Deal strategy and the New Industrial Strategy for

---

[1] NESSI (Networked European Software and Services Initiative), the European Technology Platform (ETP) dedicated to software, data and services; http://www.nessi.eu/

[2] Joint statement on preparations for a partnership on key digital technologies; https://aeneas-office.org/wp-content/uploads/2019/11/Joint-Statement-EFECS-2019-v6.pdf

Europe, share an important point in common: the significant importance and role that digital technologies play to enable the green transition and the human-centric industrial automation transition (emerging as Industry 5.0). Software technology is a scalable process-changer and innovation-supporter that can leverage the adoption of new business models, services and products aligned with these new market trends.

The Digital Transformation, or Digitalisation[3], is demonstrably a key enabler in the global economy: it has been responsible for major market disruptions whenever digital technologies have been adopted as a core part of business and progressively embraced by different industrial sectors[4]. Between 2001 and 2011, digitalisation accounted for 30% of GDP growth in the EU, although far from the 55% of the US[5]. However, the Digital Single Market strategy of the EU has contributed to an increase in the degree of digital transformation of the EU companies, to 40% on average in the EU in 2019[6].

One key component of digitalisation has been software technology. Studies have documented a systematic shift in the nature of innovation in information technology towards increasing dependence on software[7]. Software technology is demonstrably a game changer, since it has enabled change in the way that products and services are designed, produced, and sold. Software has supported the integration of digital innovations in products (e.g. smart products), the transformation of service delivery processes (e.g. resource management, productivity, logistics, automation), and even the creation of new business models with digital at their core (e.g. XaaS, 3D printing, maintenance).

Within the ICT sector, services account for the vast majority of activity, being more than ten times as large as manufacturing when measured in value added terms (3.28% of GDP compared with 0.31% of GDP in 2017[8]). Within these subsectors, “computing programming, consultancy and other services” (i.e. the software sector) and “manufacture of electronic components and boards” are the most dominant activities. Overall, the software sector accounts for 3.4% of the total value added generated within the non-financial business economy across the EU7. Moreover, since software enables and supports the digital transformation of other sectors of the economy, its indirect contribution to GDP is even higher. This can be seen in the fact that the ICT sector increased 20% on average[2] in the post-2008 crisis period (from 2012-2017), which shows a high demand for ICT services and products. The software sector includes those companies that develop software traded under licenses or offered as services. It is a very dynamic economic sector in Europe, with fast growing revenues, numbers of jobs and enterprises, and levels of innovation. In Europe, one of the strongest points is the high engineering capacity of software businesses, leading to a high level of technological and innovation potential, corresponding with the highly competitive European industry sector in general[9].

It is very important to consider that software engineering is not confined to the software industry, but it is also widely conducted in many companies from other ICT and non-ICT sectors. A 2019 survey of 90,000

---

[3] Digitalisation is defined as the use of digital technologies to change the ways in which firms do business and interact with their customers and suppliers.

[4] European Central Bank: Digitalisation and its impact on the economy: insights from a survey of large companies. ECB Economic Bulletin, Issue 7/2018; https://www.ecb.europa.eu/pub/economic-bulletin/focus/2018/html/ecb.ebbox201807_04.en.html

[5] Van Welsum, D. et al., Unlocking the ICT Growth Potential in Europe: Enabling People and Businesses, The Conference Board for the European Commission, 2013.

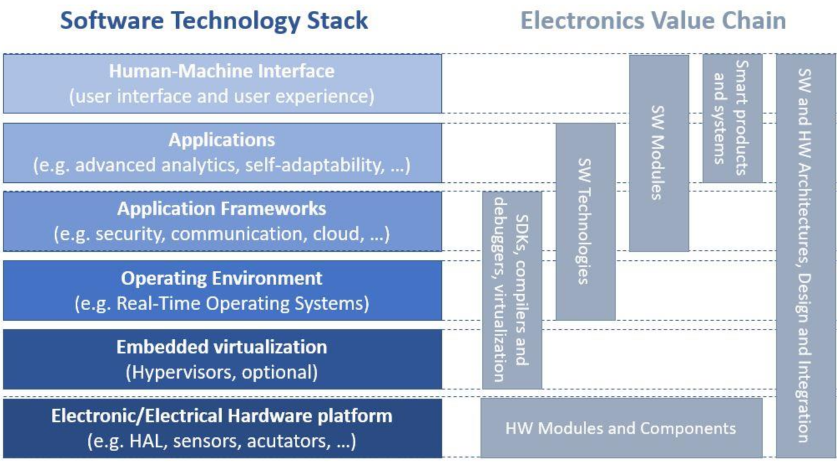
[6] DESI 2019: Digital Economy and Society Index Report – Integration of Digital Technology

[7] Ashish Arora, Lee G. Branstetter and Matej Drev. Going Soft: How the Rise of Software-Based Innovation Led to the Decline of Japan's IT Industry and the Resurgence of Silicon Valley. Harvard College and MIT Review of Economics and Statistics. Volume 95, Issue 3, July 2013. p.757-775; https://doi.org/10.1162/REST_a_00286

[8] Eurostat, ICT sector - value added, employment and R&D; https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_sector_-_value_added,_employment_and_R%26D#The_size_of_the_ICT_sector_as_measured_by_value_added

[9] Beckert, B., Schleife, K, Dupuis, D., Wydra, S., Niemann, F. The economic and social impact of software & services on competitiveness and innovation. European Commission, DG Connect, March 2017; https://op.europa.eu/en/publication-detail/-/publication/480eff53-0495-11e7-8a35-01aa75ed71a1

software developers[10] showed that only 30% of them work inside the ICT sector, while the others work in a diverse range of sectors including healthcare, transportation, critical infrastructure and manufacturing. In these vertical sectors, software developers work at different levels across the software technology stack (see figure below). At the hardware level, developers design and develop the embedded software or the device drivers that control physical sensors, actuators or other peripherals. Above this, virtualization layers enable the developers to decouple from concrete physical systems, and support the deployment of heterogeneous Operating Systems (e.g. Real-Time OS for critical applications) and increase the security (through the isolation of less-critical applications from more critical ones). The next layer, the application framework layer, provides higher-level services to the applications, covering security, communication, data management, etc. Here, software developers integrate, combine and adapt these services to address the non-functional requirements of the smart product or system being built. Finally, in the application and user interface layers, developers design, produce, validate and deploy the software which ensures that the product/system behaves and interacts with the end user as expected. This software stack reflects the entire value chain of electronic components and systems, including the manufacturing of hardware components, the development of low-level software components such device drivers, virtualization drivers and compilers, the development of software modules exploiting the hardware capabilities or providing advanced features such as analytics or intelligence, up to the development of smart products and systems for the final customer. The higher the level in the value chain, the more important is the role that software plays, and the more value it creates for the vertical sectors.



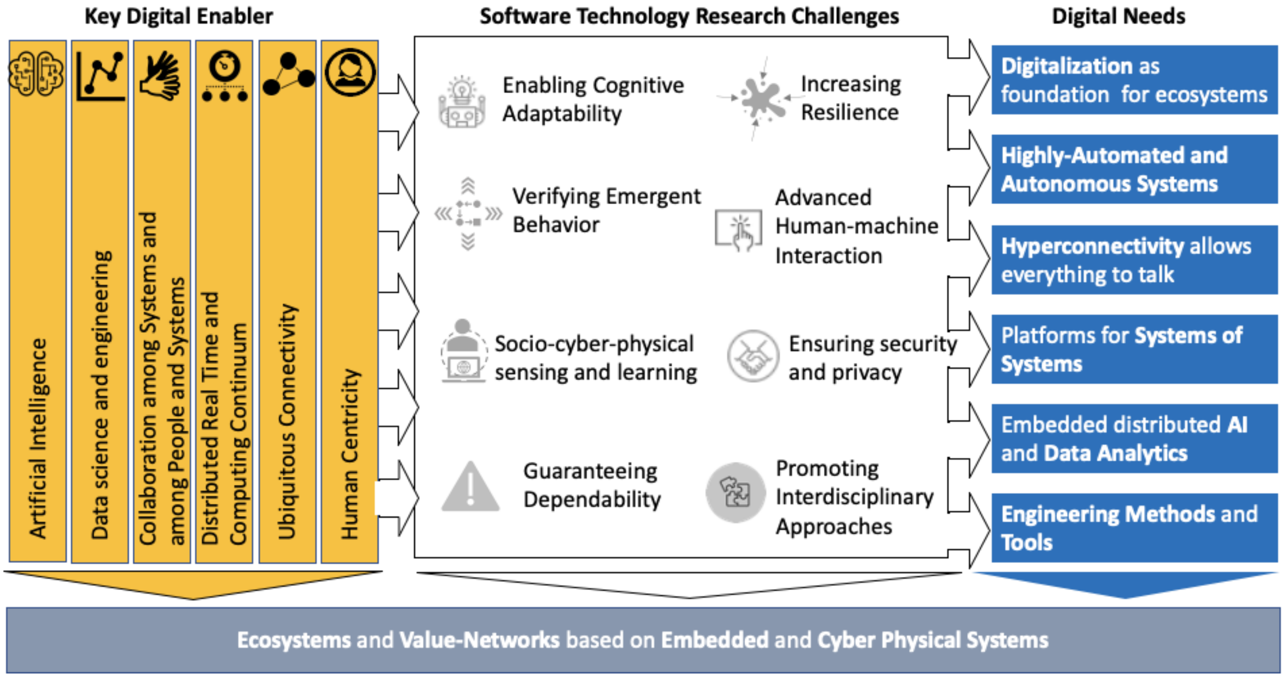
The software industry has evolved significantly in recent years, resulting in completely new ways to develop, manage and maintain software. With continuous engineering and the DevSecOps methods, software companies are embracing changes in an extreme way, by releasing new features or patching for security vulnerabilities in hours or even minutes. The software business model is turning from being license-oriented to service-oriented. Customers and end users are more and more involved in development, and new techniques such as AI are successfully used in many software applications. Such breakthroughs in software development are the result of multiple key enablers emerging in recent years, such as AI and ubiquitous connectivity. However, such advanced improvements in software engineering are still mainly within the software industry itself, and are adopted only with significant delay in other industry sectors. This is due to several challenges in applying those key enablers to the software development of electronic components and systems, as discussed below. To address these challenges, we need research and innovation actions addressing several important topics in software technologies.

In the rest of this paper, we first give a brief analysis of the current trends in software technology, elaborating the key enablers behind the advanced improvement of software technology, and the future challenges to apply these enablers in the context of software development for KDT systems in order to meet the digital needs. After that, we identify and recommend several software research and innovation topics to address these future challenges.

[10] Stackoverflow, Developer Survey Results 2019; https://insights.stackoverflow.com/survey/2019#company-type

Trends and Future Challenges in Software Technology

In a previous paper "Next Generation Software Technologies Empowering the Digital Transformation of Europe - Recommendations on Software Technology Research for Horizon Europe"[11], NESSI considered the fundamental role that software plays in the digitalisation of society and economy. Even though software by itself (i.e. without a hardware execution environment) is not functional and has no behaviour, the software part of complex technical systems becomes the game changer in the international competition. Five key emerging software technologies were identified (key digital enablers), which will have a fundamental impact on the extent and quality of how embedded systems and cyber-physical systems will master the challenges of digitalisation and ecology in the future (see figure below).



Research is required to deliver the next generation of software technologies that exploit the following emerging digital enablers:

- Artificial Intelligence (AI) delivers unprecedented machine-based analysis, cognition and decision making based on advanced machine learning technologies and big data assets. Thereby, AI enables novel, cognitive software systems, as well as new means to support software development.
- Data Science and Engineering advancement, together with Artificial Intelligence, have paved the way for profiting from data as never before. Cyber-physical and IoT based systems operations yield a huge amount of data that thanks to current Data Science and Engineering can be turned into knowledge, used for business benefit, and as a basis to enhance user experience.
- Collaboration among systems and among people and systems refers to the ability of systems and humans to interact to achieve an overall goal that the individual systems or people themselves cannot achieve. This also includes the combination of AI technology with the ability of human thinking. This is an essential prerequisite for the realisation of almost all future scenarios in the field of digitalisation and the responsible handling of natural resources and of our habitat.[12]

[11] NESSI - Software Empowering the Digital Transformation of Europe; http://www.nessi.eu/Files/Private/NESSI%20-%20Software%20Empowering%20the%20Digital%20Transformation%20of%20Europe%20-%20final%20version%2009-2018%20v1.pdf

[12] Malone, T. W., Superminds - The Surprising Power of People and Computers Thinking Together. Little, Brown Spark, 2018.

- Distributed real-time software and the Computing Continuum ensure that distributed systems of different types of electronic components communicating via networks are able to perform their tasks while ensuring compliance with hard real-time conditions, in particular if technologies like cloud, fog, and edge computing have to be combined. Software-defined networks play a fundamental role in providing on-demand dynamic network infrastructures.
- Ubiquitous connectivity, driven by the new generation of mobile networks and novel electronic devices, allows large-scale, high-bandwidth and almost instantaneous interactions between computers and devices, among users, communities and autonomous agents, as well as between data, media and knowledge actors.
- Human centricity means that software is intuitive and trustworthy, facilitating transparent and fluent interaction between users and digital systems. Human centricity empowers more and more people to use, interact and adapt software systems to their needs and situation.

These enablers will empower Europe to meet the following fundamental needs of digital society and the Green Deal:

- Digitalisation as the foundation for ecosystems spanning full value chains and relying on the idea of integrating different players focusing from the physical world to cyber space in order to create value for the whole of society, and to optimize the overall resource consumption.
- Highly automated and autonomous systems to relieve people wherever and whenever possible and to act reliably wherever it is possible or necessary to react quickly in an automated way.
- Hyperconnectivity allows very different technical devices, notably embedded and cyber-physical systems, to communicate and share large amounts of information with each other, including a seamless integration of human users into communication processes.
- Platforms for systems-of-systems represent the distributed software infrastructure that allows the realization of ecosystems and of the related value network as an underlying element of the ecosystem itself, enabling full monitoring and control of its parts, operations management and full life-cycle support, and ensuring trust.
- Embedded distributed Artificial Intelligence and Data Analytics are essential means for enabling systems to operate to a large extent autonomously in unforeseen, unknown, or uncertain environments, or within novel forms of collaborating system groups.
- Engineering methods and tools that are innovative in the sense that they support novel development paradigms such as continuous development in a practical way.

Research is required to deliver new software technology that exploits the key digital enablers in building embedded and cyber-physical systems, to address the digital needs. NESSI envisions the following software technology research challenges to achieve such new software technology:

- Enabling cognitive adaptability to ensure that systems or groups of collaborating systems can flexibly adapt to changing environments (e.g. by AI-based reconfiguration of a system groups), even beyond constellations of the environments that could already be foreseen at design time.
- Verifying emergent behaviour to prove the presence of desired behaviour as well as the absence of unwanted behaviour (in particular, hazardous behaviour) resulting from the interplay of systems during operation.
- Socio-cyber-physical sensing and learning capability of cyber-physical systems, interacting with people and physical environments in context (including e.g. industrial production machinery), to autonomously learn patterns or behaviours which can be exploited, e.g. for exploitation of tacit human knowledge in industrial work for automation of tasks via autonomous learning agents, or for optimization of industrial production and work processes.

- Guaranteeing dependability by helping implement smart autonomic loops to control the behaviour of critical digital systems, including guaranteeing reliability, safety, security and availability, as well as compliance to normative regulations and ethical principles.
- Increasing resilience, ensuring that individual systems and groups of collaborating systems continue to maintain functionality in face of unexpected alteration in the environment, including the capability of groups of collaborating systems to reconfigure to increase resilience.
- Advanced human-machine interaction, by enabling the adaptation of systems to human behaviours and improving the machine understanding of human intent, including the explainability of AI-induced behaviour, and transfer of control to human users (e.g. avoiding mode confusion).
- Ensuring security and privacy, important for the data and the systems, as well as the perceived reputation of the companies responsible for the development or operation of the systems.
- Promoting interdisciplinary approaches, considering all relevant perspectives, including systems and software engineering, automation science, cognitive sciences, behavioural psychology, data analysis and machine learning, as well as technology assessment and ethics.

Software Engineering Research & Innovation Topics

NESSI envisions that research and innovation actions around the following topics will be important to address the future challenges of software in the context of Key Digital Technology, and eventually bring the new trends of software technologies to the development of embedded and cyber-physical systems. The table below summarizes how these topics are related to the eight challenges we presented in the previous section.

	Cognitive adaptability	Verifying emergent behaviour	Sensing and learning	Garanteeing dependability	Increasing resilience	Advanced H-M interaction	Ensuring security and privacy	Interdiscipline approaches
DevOps for electronic components and systems	✓				✓	✓		
Software-driven system integration		✓					✓	✓
Trustworthiness of software and development	✓	✓		✓	✓		✓	
Managing complexity dynamicity and uncertainty	✓	✓		✓		✓		
Software for Spatial Computing			✓			✓		✓
Software for sustainability and energy efficiency	✓		✓					✓

Advanced software development methods for electronic components and systems

Modern software development teams are turning quickly towards DevOps, which offers great agility for introducing new and enhanced functions required to stay competitive. Some market-leading tech companies are releasing tens of new versions of their software every day, deployed directly to their customers[13]. However, the current automatic DevOps tools assume homogeneous and centralized resources provided by cloud, which is not the case in KDT systems that are composed of heterogeneous and distributed components. Research and innovations are required on all the stages along the development lifecycle. As the

[13] How Etsy Deploys More Than 50 Times a Day; https://www.infoq.com/news/2014/03/etsy-deploy-50-times-a-day/

scale and complexity of electronic systems can easily exceed the capacity of manual management[14], AI will play an important role, raising automation to a new level.

The development of KDT applications is an integrated process with both software development and hardware design. New methods and tools must ensure the involvement of multiple stakeholders, such as device manufacturers, technicians, and end users. Considering the flexibility of hardware choices, higher-level languages that can be compiled onto multiple platforms are also a promising direction. Due to the heterogeneity of electronic components, the source code may need to be compiled separately towards different hardware architectures or packaged with different libraries. The advancement of virtualization and container technologies will help, but needs to be extended to cover specialized processors (e.g. neuromorphic chips, tensor processing units and even FPGAs). Automatic testing after each DevOps iteration is critical to guarantee security and dependability, but effective auto-testing of software on KDT systems relies heavily on the proper simulation of electronic components and even the physical world behind them, including their emerging behaviours, using new techniques such as digital twins. Chaos engineering is widely used in cloud computing to test the resilience of running systems, but it requires novel ways to inject artificial failures into the hardware layer.

Automatic deployment powered by Infrastructure as Code (IaC) techniques is a main driving force behind DevOps. In electronic systems each node should be deployed individually with the specific software to maximize the fit to its unique context. In a large set of devices, fully automatic deployment can be only achieved via cognitive adaptability powered by AI techniques, e.g. learning from historical and simulated deployment experiences to plan for the best deployment arrangement. Linked to this is the need to support automatic upgrades when new versions of the code (e.g. security patches) must be deployed along the whole computing continuum, supporting the cohabitation of different versions without interrupting the service. All the automation tools need to be coordinated by a CI/CD (Continuous Integration/Continuous Delivery) pipeline, which needs to cover the activities that are not typically software development, such as the manufacture and delivery of devices, the enrolment of devices, hardware maintenance and upgrade, etc., based on advanced human-machine interaction between stakeholders and the pipeline. Unlike in cloud computing, during the long lifecycle of electronic components and systems the pipelines will not always drive every node at the same pace. Some nodes may keep running an older version of software to maintain global resilience, and some nodes may be chosen to preview an immature version, similar to the A/B testing of web applications.

Software-driven integration of KDT applications

Without interaction with each other, KDT applications are information silos, which become an obstacle for potential business value creation. Software-driven integration, i.e. developing new software to integrate existing applications, is a trend in the software industry. Mainstream applications are undergoing platformization, e.g. Facebook is now a platform that allows third parties to develop value-added services to connect the social network with other applications. For enterprise computing, companies usually spend ten times more on customizing and integrating the enterprise applications than the licence fees of these applications[15]. New research is needed for the platformization of KDT applications, which requires the applications to provide powerful APIs to external components and systems. It also requires novel software architecture within the applications to achieve the flexibility for deep customization. Supporting integration is challenging because custom code will share the already constrained resource of electronic components and may also bring vulnerability to applications that are security- or safety-critical. New design methods are needed, with performance and risk assessment considering resource and hardware aspects, together with

[14] As observed by Gartner: “the cost of deploying and managing an edge computing environment can easily exceed the project’s financial benefits"; https://www.gartner.com/en/newsroom/press-releases/2018-12-03-gartner-says-the-future-of-it-infrastructure-is-always-on-always-available-everywhere

[15] Gartner: The Rise of the Postmodern ERP and Enterprise Applications World; https://www.gartner.com/en/documents/2633315

novel isolation mechanisms on low-level electronic components, potentially supported by virtualization techniques. KDT applications also need to be integrated with traditional enterprise and consumer software applications, since the latter are currently managing the data and business processes. Such integration introduces electronic components into the traditional human-data interaction, and thus calls for novel software-hardware co-design in an agile and continuous way, in order to bridge the social-cyber-physical sensing powered by new electronic components with the business data and process controlled by the traditional enterprise software systems.

Software integrators play an important role in the software industry and in the software value chain. New abstractions, orchestration languages and integration methodologies are needed to facilitate the interdisciplinary thinking of integrators. Future research and innovation actions in this direction require close collaboration among software engineering researchers, application providers, software integrators and the integration platform providers, resulting in the extension of mainstream iPaaS (integration Platform as a Service) solutions to reach the lower-end devices. Platformization and integration are also important for KDT application providers to develop ecosystems involving component providers, software vendors and integrators. Research and innovation is needed to investigate business models, legal issues, data sharing strategies, etc., in order to integrate scattered businesses into prosperous European KDT ecosystems.

Trustworthiness of software and software development for KDT

Following ISO/IEC 25000, trust is the "degree to which a user or other stakeholder has confidence that a product or system will behave as intended", and trustworthiness of software includes characteristics such as security, privacy, accountability, reliability and resilience. Trust in KDT systems is particularly complicated, because: (i) quality characteristics of electronic components are interdependent; (ii) electronic components with their software interfaces are close to the physical world, which means that security or reliability risks may lead to crucial or even fatal consequences; (iii) the inclusion of the social dimension into KDT systems results in software-interoperated systems-of-systems, where human factors play an important role in the trust of systems; and (iv) many of the functions of the KDT systems are achieved thanks to system and component interoperability (that is, each of the systems, and components that make up the KDT systems can work with many of the rest of the KDT systems and components, making an extensive use of the functions that each system or component implements).

Trustworthiness needs to be considered as an objective throughout the entire development lifecycle of these systems. Agile development and DevOps enable prompt reaction to emerging vulnerabilities, which is helpful in increasing the trustworthiness of the development process, but ensuring trustworthiness of the product itself along fast release cycles and earlier involvement of users is still an open research topic. The classical testing and verification approaches are still important for the trustworthiness of software for KDT, but as systems becomes more and more complex new approaches are required, such as systematic risk and threat analysis during the development lifecycle, based on evidence collected from not only the running system but also the development processes, as well as the novel security by design approach which emphasizes the fast reaction to emerging vulnerabilities. Every system is developed considering a context, and the context for every system will not be the same; this is particularly challenging in the case of systems of systems. Including hardware characteristics into risk analysis and security-by-design is not an easy job.

Trustworthiness of complex systems comprising distributed and heterogeneous components is challenging, requiring novel ways of trust management. For example, it is not always feasible for a car to assess the trustworthiness of other cars by relying only on either its own knowledge or a central referee. More attention is needed to distributed and decentralized trust mechanisms, combining new technologies such as distributed ledger. Such mechanisms need to be able to verify behaviours emerging from both the unpredictable external environment and the complex interaction among the internal components. Considering the social aspect, new human-machine interaction methods are also needed to build trust between systems and their stakeholders in a reliable and user-friendly way. The trust management mechanisms need to be self-adaptive against environment changes, system failures, newly discovered vulnerabilities, or attacks. As KDT systems usually comprise many components from different sources, cognitive adaptation across multiple security domains will be challenging.

Managing complexity, dynamics, and uncertainty of KDT applications

Digital systems and groups of collaborating systems, as well as the environments in which these systems operate, are becoming more complex, show highly dynamic behaviour, and increasingly face uncertainty during operation. The trend toward digitalization is accompanied by a significant increase in complexity, dynamics and uncertainty. The increasing complexity can be seen in individual KDT systems, groups of collaborating systems, and the environments in which KDT systems operate. Managing the dynamics of systems, collaborating groups and their environment poses great challenges for the engineering and operation of embedded and cyber-physical systems in future. Furthermore, embedded and cyber-physical systems increasingly face uncertainty during runtime, i.e. systems have incomplete or ambiguous information about the environment in which they operate. This is especially the case when systems are operating in open contexts where the relevant properties of the environment cannot be completely anticipated at design time, and therefore cannot be fully handled by predefined adaptations. In many future scenarios, such as autonomous driving or smart factories, systems must be able to meet their goals even on the basis of incomplete or contradictory information about the environment, e.g. the intentions of other systems or humans, or the preferences and skills of human users. In order to address these challenges, software research has to provide answers to questions such as how to guarantee appropriate and safe cognitive adaptability in complex, highly dynamic and uncertain environments, and how to verify - under hard real-time constraints and during runtime - emergent system behaviour resulting from the interactions of subsystems and the ambiguity of environments. A special challenge will be the modelling and implementation of human-machine interactions under those conditions, for example the efficient and effective transfer of control between systems and human users to avoid effects of mode confusion in autonomous driving. Developing software for complex and dynamic systems, able to deal with uncertainties, will require engineering processes involving multiple disciplines such as cognitive science and sociology.

Promising approaches to deal with the increasing complexity, dynamics and uncertainty must consider both design-time and run-time. Such approaches will be based on innovative combinations and improvements of software technologies in the following categories: innovative technical solutions in the area of software technology including design-time and run-time techniques for collaborative information fusion, collaborative runtime verification (including digital twin technology), environmental perception with shared models of the environment, prediction of future behaviour and cognitive adaptability of individual systems, collaborative groups and the environment. Innovative process-related solutions in software technology should include approaches from design science, agile methods, and the creation of appropriate team culture. The core of design science and application of digital technologies is to produce solutions, artefacts and knowledge which are relevant for human purposes and achieving goals (organization and individual goals). In design science digital solutions are produced and studied in an operational application context (real world use context), where the maturity, quality and value of the solutions can be evaluated. The solutions often involve combinations of evolving operational processes, software, hardware and ICT systems governed by multiple organizations, which creates a complex context for design and deployment of new digital solutions. At design-time, the related engineering activities, including software engineering, need to be well aligned with the overall solution goals and requirements, and hardware and network requirements and limitations, as well as business requirements, need to be considered in parallel with the software engineering process. At run-time, the resulting technical system (or system-of-systems) may be widely distributed across different embedded, networked and cloud computing nodes governed by multiple different organizations. Accordingly, the role of governance boundaries and multi-organization collaboration, both at design-time and run-time, deserve further research in software engineering, to discover ways to manage the complexities involved in design and deployment of new digital solutions.

Software for Spatial Computing

Spatial Computing is the emerging interaction mechanism for digital content in a converged cyber-physical world. Advances in devices and user interfaces (e.g. mixed reality glasses, gesture recognition, haptic feedback, interfaces built from new materials) and their integration into a spatial computing system will allow for more adaptable, responsive and immersive interactions with the digital world. The services offered by the spatial computing system will provide new ways of augmenting user experience and will allow us to 'feel' with all senses the virtual environment around us. Just like the real word, spatial computing offers a rich environment for multi-user interaction, and empowers a human-centric approach for future digitalisation. Taking industrial automation as an example, the introduction of spatial computing will help put human needs and interests in a central role, focusing not only on how to automate and optimize the production process, but also on how to get workers involved in the process.

Spatial computing systems will be complex constellations of software and content components operated by a multitude of ecosystem participants. The intelligence required for smart interaction mechanisms will depend on collecting and analysing massive amounts of social-cyber-physical sensing data across all stakeholders in the ecosystems into digital super twins. The required computing power will not be limited by the capabilities of the devices involved but provided by the cloud or at the edge. Software engineering approaches for designing and developing spatial computing systems will need to cope with the challenging environment of diversity in devices and application domains, with intelligent deployment and adaptation capabilities. New programming models, languages and methodologies will emerge in the spatial computing era, and research and innovation actions will help to create break-through progresses. This will require interdisciplinary research integrating advances in media technologies (new coding technologies for digital content), cognitive psychology (new human-machine interaction for better attention and perception from the users), social science (new methods and processes for better human collaboration), etc.

Software for sustainability and energy efficiency

Although digital technologies and software may provide very powerful tools to optimize the energy efficiency in vertical domains, their absolute and relative energy and resource consumptions continue to increase, even if hardware itself improves (notably for embedded systems as a by-product of autonomy optimisation or to reduce the cost of big data centres). The increasing functional scope of software and applications, the introduction of data intensive algorithms and systematic logging of events, the use of complex middleware stacks (hypervisors, virtual machines, containers, languages runtimes, bloated framework) all contribute to degrade the environmental impact of software-based systems, often sacrificing frugality for the sake of ease of development and time-to-market. The sustainability concern needs to be natively addressed in the development and execution phase of all digital systems (embedded, personal, large-scale, communication equipment, etc.)[16]. New tools and models are needed to optimize the intimate interactions between hardware and lower software layers, to adapt to runtime context, and continuously minimize energy and resource consumption, based on monitoring not only the internal behaviour of software systems but also the external physical environments through advanced sensing and learning. Sustainability also calls for simplified and efficient architectural patterns, along with the appropriate education of key actors such as developers, software architects, system integrators and data centre management teams. Interdisciplinary research will provide novel solutions towards sustainable digital systems, e.g. the use of energy harvesting from the environment (wind, solar, pressure, etc) or other energy sources (body heat, foot strikes, etc.) to power lower-end devices, making them battery-free[17]. This also calls for new programming models, software architectures and self-adaptation approaches to cope with novel and potentially unstable power sources.

[16] Lean ICT: towards digital efficiency, The Shift Project; https://theshiftproject.org/en/article/lean-ict-our-new-report

[17] Garg, N and Garg, R, Energy harvesting in IoT devices: a survey, 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, 2017, pp. 127-131, doi: 10.1109/ISS1.2017.8389371.

Recommendations for Horizon Europe and KDT

To address the challenges and to support the area of KDT with advanced software technologies, NESSI has identified and recommend the following research and innovation topics:

- DevOps for electronic components and systems, involving all players along the KDT value chain.
- Platformization of KDT applications to support software-driven integration and to build ecosystems.
- Digital trust in systems with large number of components in a distributed and context-aware way.
- Management of complexity in the presence of dynamicity and uncertainty arising from, and addressed by, AI-powered self-adaptation.
- Spatial Computing that utilizes advanced human-machine interactions enabled by novel electronic components.
- Software support for sustainable and efficient computing.