Managing Complexity

Cyber-physical systems

ESI 20 YEARS

ESI

An initiative of industry, academia and TNO


TNO
ESI















Managing Complexity

in cyber-physical systems

ESI 20 YEARS

ESI

Powered by industry,
academia and TNO

Text
Nieke Roos, Alexander Pil (Techwatch)

Editorial committee
Joris van den Aker, Frans Beenker, Wouter Leibbrandt, Judith Tesser, Jacco Wesselius, Ad Westveer (ESI)

Graphic design and layout
Justin López (Techwatch), Camiel Lintsen (Kade05)

Pictures
Bart van Overbeeke (BVOF), Bram Saeys, Jeroen Verreijt, Studio John Geven

Printing
New Energy

ISBN
978-90-5986-512-9

Acknowledgment
Thanks to all who contributed to the contents of this booklet, especially our ESI colleagues and ESI partners: ASML, Canon Production Printing, ITEC, Philips, Thales, Thermo Fisher Scientific, Vanderlande, University of Amsterdam, Delft University of Technology, Eindhoven University of Technology, Radboud University Nijmegen, University of Twente and TNO.

Contact
ESI (TNO)
High Tech Campus 25
5656 AE Eindhoven
The Netherlands
T +31 88 866 54 20
E esi-office@tno.nl
esi.nl

Contents

Preface

This year, 2022, we're celebrating the 20th anniversary of ESI. To mark this moment, we've compiled this booklet. It contains an overview of our history, ongoing research activities and results, and a glimpse into the future of ESI.

At the start in 2002, "embedded systems" was selected by the Ministry of Economic Affairs with the support of industry and academia as one of the key areas for Dutch competitiveness. Twenty years later, we can say with some conviction that systems engineering has become a strategic spearhead worldwide. ESI's research program for the coming years is full of demanding, topical challenges such as the need for guaranteed performance and reliability of high-tech equipment ("towards zero unscheduled downtime"), the wide diversity (customer and application-specific) with which high-tech equipment is produced, used and maintained, the integration of high-tech equipment in larger systems, the application of artificial intelligence in high-tech equipment (opportunities and challenges), the (market) need to continuously and frequently update safety/mission-critical systems, the shortage of R&D experts who can oversee the complexity of the systems, the effective introduction of model-based methodologies in the industry, and the list goes on.

Per July 2022, the ESI management team changed from Frans Beenker and Wouter Leibbrandt (above) to Jacco Wesselius and Wouter Leibbrandt (below).

The knowledge ESI has been able to develop over the years together with partner companies from the high-tech industry, university partners, international partners and TNO is widely disseminated. Methodologies are freely available and shared through the ESI academy, industrial and academic partners, the international network and with the help of implementation partners.

Our "industry-as-a-lab" approach is unique: research is conducted on-site at our industrial partners. This booklet contains a selection of stories illustrating the results of this approach. Each of these stories was previously published in Bits&Chips. Our thanks go to Nieke Roos, who was able to compile all the stories based on interviews into a coherent whole. And to the designer Camiel Lintsen, who was able to transform our usually invisible work into appealing images.

2022 is also the year in which we bid farewell to Frans Beenker, board member from the very beginning. His contribution to ESI has been invaluable. His role will be taken over by Jacco Wesselius, who together with Wouter Leibbrandt will work on strengthening the position of ESI in the coming years.

We hope to meet you again in the near future to continue discussing the challenges and opportunities of the high-tech industry or similar challenges and opportunities in other areas.

ESI

When you visit ESI today, you enter a large open office space, surrounded by meeting rooms and a few closed-off spaces for conducting calls and remote sessions. The place is geared towards connecting colleagues and exchanging results, questions and insights. ESI is all about collaboration.

Each Monday, all colleagues are at the office, to connect across projects, meet in expertise teams, join the weeklies and have brainstorms. You see groups of research fellows drawing on the wall-covering whiteboards to exchange views and sharpen their thoughts. And regularly, the open office space changes into a central gathering area where general ESI matters and news are shared.

From Tuesday to Friday, our offices are much quieter. The research fellows will typically be working on one of our partner locations as part of our successful industry-as-a-lab concept. Since the introduction of hybrid working following the Covid-19 pandemic, they may also be working from home one or two days per week.

Industry-as-a-lab

ESI's 35 research fellows are a mix of young PhD graduates and experienced practitioners, including seven part-time university professors. We have 5 project managers guiding the collaborations with our partners. A pool of about 10 scientific programmers on temporary assignment brings in invaluable software engineering skills. Including our competence development managers and our supporting staff and general management, ESI currently employs approximately 60 people from 17 nationalities.

Our researchers work multiple days per week at the location of our industrial partners on actual industrial systems. This industry-as-a-lab concept is an essential part of the ESI way of working. It's based on the fundamental belief that studying methodologies that solve complex industrial challenges can't be successful without being confronted with the daily practices of the industry. The best laboratory facility for our people to work is the industry itself and we're fortunate that industry offers us that opportunity.

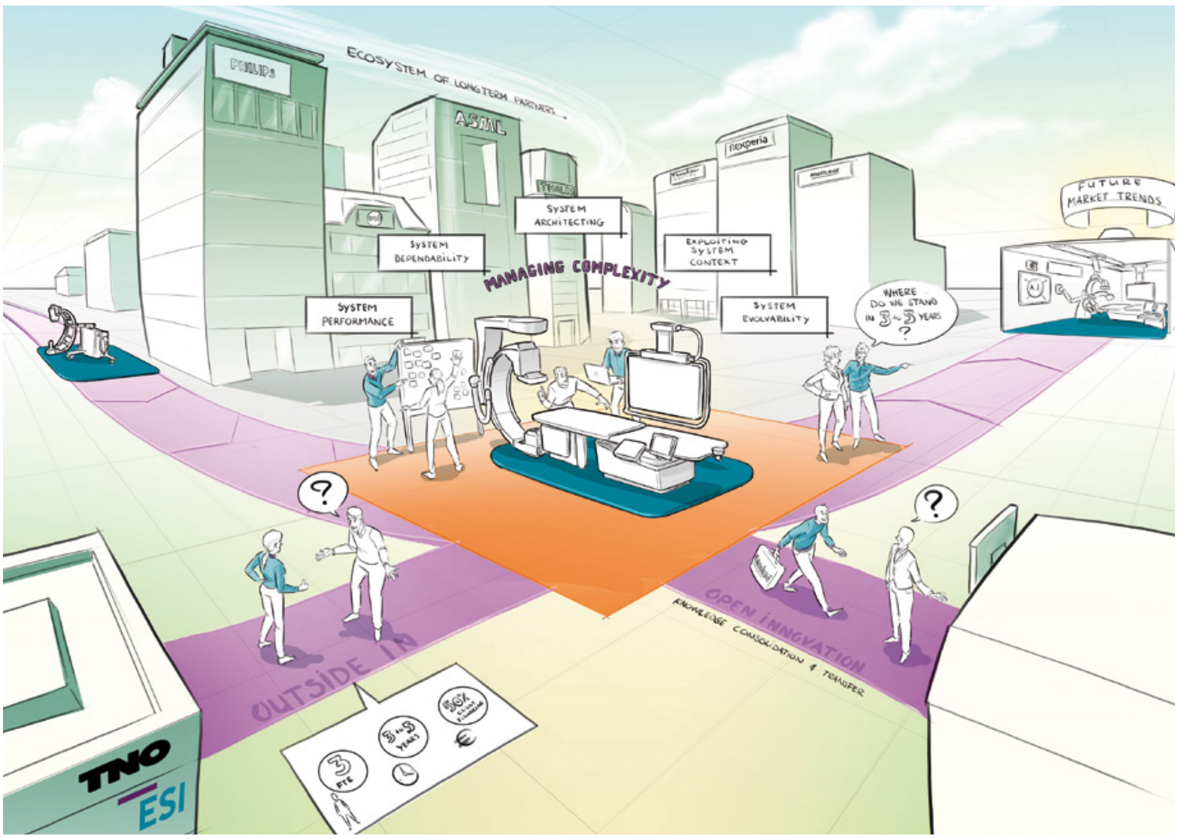
For ESI, industry-as-a-lab leads to much better insight, understanding and appreciation of industrial problems. Conversely, industry partners build an early appreciation of our results and methodologies. Our researchers have access to sensitive information, allowing their findings to be directly validated to realistic industrial cases. Valorization is thus pre-built into the process, with the industry location as the research laboratory. Furthermore, industry-as-a-lab provides us and our academic partners with valuable new insights and drivers for future research.

Bridge between industry and academia

ESI also collaborates closely with these academic partners. Our part-time professors work one or more days a week at our partner universities. This gives us direct and free access to the latest relevant scientific results, which we can bring to higher technology readiness levels through our applied research program. Conversely, through our part-time professors or by responding to or even defining calls, we feed the academic community with relevant research questions and challenges that need to be addressed to satisfy the future needs of the high-tech equipment industry.

Universities collaborate with ESI for a variety of reasons. The partnership provides them with a route to valorize their own work. It also opens the door to valuable industrial connections. For their students, it's a rich source for interesting and relevant PhD projects.

ESI's research ties into both national and international innovation policy. We're involved in the Systems Engineering roadmap of the Dutch Top Sector High-Tech Systems & Materials and European frameworks like the Electronic Components and Systems Strategic Research and Innovation Agenda. Through projects and workshops with peer institutes such as the DLR Institute of Systems Engineering for Future Mobility and Fraunhofer IESE from Germany, the US-based Systems Engineering Research Center (SERC) and the Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at KTH Royal Institute of Technology in Stockholm, we identify cross-border industry problems and needs and we exchange insights and results.

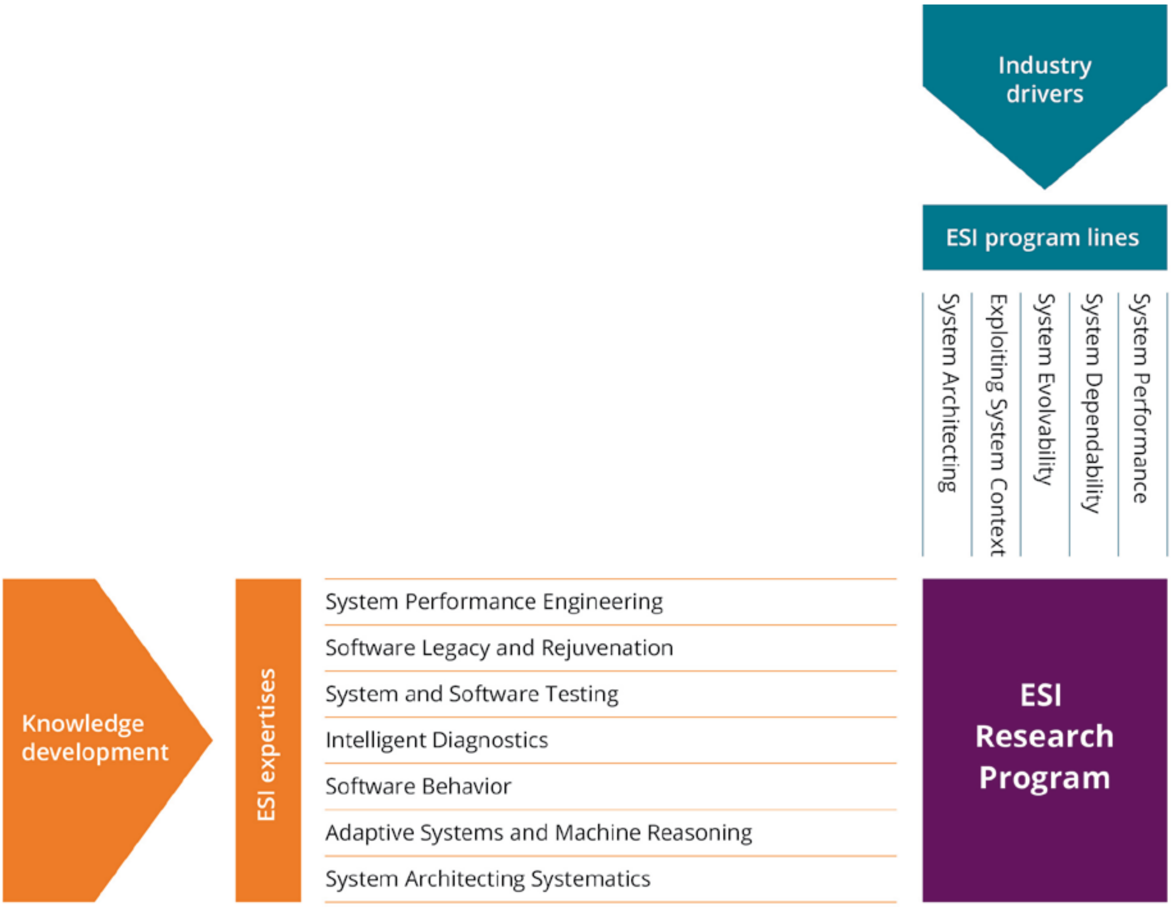


Research program

Driven by the industry's needs, ESI's activities are clustered in five program lines. "System performance" looks at quantitative design criteria for high-tech systems and their resource utilization in trade-off with cost. "System dependability" focuses on aspects like availability, reliability and maintainability. "System evolvability" aims to prolong the useful economical life of products by rejuvenating existing systems and anticipating future developments and minimizing any possible negative consequences. "Exploiting systems context" focuses on enhancing the value of systems by integrating them more closely into the customers' systems, workflows and constraints. "System architecting" addresses the challenge of getting the system right by helping customers to translate market, product and technology choices into system concepts.

The research in these program lines can draw from the knowledge ESI has developed over the years. This knowledge is clustered in seven expertise areas, each cultivated by its own team of experts. We have expert teams focusing on system performance optimization, the systematics of systems engineering, software legacy and rejuvenation, intelligent diagnostics, software behavior, system and software testing and machine reasoning.

ESI's projects are situated at the intersection of the program lines and the expertise areas. They fall under one of our five general research topics and make use of the knowledge from one or more of our expert teams. Between the projects within a program line, there's extensive cross-fertilization of insights and results.



Results

The ESI results generally have the form of engineering methodologies. These consist of formalisms (e.g. for modeling system behavior or requirements), techniques (e.g. for retrieving information from models), methods (e.g. for solving a design problem using models) and tools (e.g. for efficiently applying new modeling techniques). The methodologies developed are tuned to the specific needs of the high-tech industry and the ability to apply them is transferred to our industrial partners.

Ideally, a newly developed or improved methodology can be applied using existing tools. Often, however, there's no tool available that meets the requirements and one is created as part of the research. To enable adoption by the industry, the newly created tooling is transferred to tool vendors such as Obeo from France to be professionalized, brought to the level of industrial applicability and made public as "managed open source" via the Eclipse Foundation. This gives us the best of both worlds: industry-relevant tooling with professional support.

To help us put the project results to practical use, we've started working with implementation partners. We teach and train them to work with our methodologies and tools so they can apply them at their customers. The first we're doing this with is Capgemini Engineering. Joining forces with a large organization like this gives us access to a sizable contingent of specialists to disseminate our project results. Thus, the implementation partnership acts as a lever to grow the industrial adoption of our methodologies and tools, turning ESI's crowd of 60 into a multitude.

Vision and mission

ESI's way of working, the bridging role between academia and industry, our international network, our research program and the effort we take to create full impact with our results all follow naturally from our vision of the future and our ambition to fulfill the mission we've set for ourselves.

ESI vision

A world-leading Dutch high-tech systems industry that creates consistent and increasingly positive economic impact and societal value by constantly improving the effectiveness of its way of working.

ESI mission

Embedding cutting-edge methodologies into the Dutch high-tech systems industry to cope with the ever-increasing complexity of its products.

The Dutch high-tech

The Dutch high-tech equipment industry is developing world-class systems for a wide range of markets. They include wafer scanners, healthcare imaging systems, professional printers, electron microscopes, warehouse automation solutions and agricultural robots. For the development of these systems, the industry relies on strategic collaborations in a large network of OEMs, suppliers and SMEs.

Despite their different focus areas, the Dutch high-tech equipment makers have a lot in common. They all target the high end of their respective markets, serving an international customer base. They all make complex systems in relatively low numbers – typically in the hundreds per year and sometimes even less. And all the systems remain operational in the field for a long time – twenty years is no exception. Last but not least, the companies also share a common business driver to further digitalize their products and solutions. ESI supports the industry through the development of new methods and techniques for system design and engineering to leverage this digitalization.

The Dutch high-tech equipment industry also commonly recognizes the importance of joint innovation. Technology is ever-advancing and customer demands are ever-increasing. To remain world-class, the industry realizes that collaboration in research and development to manage the ever-increasing complexity is key. Being non-competitors, ESI partners are also very much open to working together. Our open-innovation model enables them to learn from each other and stay on top of market developments.

Changing business

Over the years, the business focus of the industry has gradually expanded. Initially, companies concentrated all their efforts on delivering high performance. At some point, having become a critical part of customer processes, the systems also had to become dependable and always-on. And during all their time in the field, they had to remain up-to-date, providing the latest features. The next step is connecting them to other systems in the customer context. Digitalization is the main driver and enabler here, bringing new challenges of complexity and system dynamics.

As the product requirements grew from just high-performance to also dependable, evolvable and integratable, the companies went from competing in total cost of ownership to competing in end-user integration. While, originally, they could focus their attention on being good in engineering along every quality axis, they now have to deliver both high-quality and care-free systems. They also need to be prepared to adapt their products to customer-specific requests and they have to integrate them into customer processes and deal with the consequences.

The ESI approach

The Dutch high-tech companies all follow this business route and ESI helps them along the way. In our industry-as-a-lab approach, we do applied research in close collaboration with industry as well as in strong association with the fundamental scientific work of academia. We structure the knowledge gained, professionalize the results, facilitate their exchange in an open ecosystem and help put them into practice through competence development and implementation partnerships.

ESI is the linking pin between Dutch high-tech companies and between industry and academia. Not only do we define a common and shared roadmap, but we also bring the two sides together in research projects. Through our competence development program and special interest groups, we connect them to outside perspectives from facilitators, teachers and others from our network. In addition, we frequently organize more informal meetups, like our ESI Symposium and various other get-togethers.

Academic collaboration

Universities have three primary responsibilities: research, valorization and education. All three are equally important for the high-tech ecosystem and therefore for ESI. To support these activities, we maintain close ties to the academic world and our partner universities in particular – Delft University of Technology, Eindhoven University of Technology, Radboud University Nijmegen, the University of Amsterdam and the University of Twente.

In research, valorization and education, we're closely collaborating with departments like computer science, control engineering, electrical engineering, mathematics and mechatronics. Systems engineering is becoming increasingly relevant as well. From these fields of study, we have professors working part-time at ESI and, conversely, people from ESI working part-time in academia.

Research and valorization

Many developments in industry are rooted in scientific research. The academic world is a rich source of fundamental knowledge about formalisms, techniques and methods. Through our partner universities in the Netherlands and our international peer institutes, we're also offered a valuable window on global advances in the various disciplines.

At ESI, we build on this academic knowledge by applying it, validating it and combining it in an industrial context, together with our partner companies. Thus, valuable academic results and knowledge are brought to real value in industrial practice. In doing so, we also come across fundamental questions, which we relay back to our partner universities. Next to that, we directly link our academic companions to the industry and the industrial challenges, e.g. through internships, graduation projects and PhD projects. This creates a triangular interplay between universities, companies and ESI, with ESI as the orchestrator.

As part of this orchestration, we're involved in the creation of Dutch and European roadmaps. We also have a role in influencing the academic agenda. A notable example of this is the MasCot program.

> The Mastering Complexity program (MasCot) is a 6-year partnership with the Dutch research council NWO to investigate and deliver the next generation of engineering methodologies to help manage the increasing system complexity at low cost and enable future industrial systems to be quickly developed while providing high quality. This program was set up by ESI with the explicit goal to create a continuous, seamless pipeline from academic research at technology readiness level (TRL) 1-4 into our regular research program at TRL 4-7 to final utilization in industry at TRL 7-9.

The Mascot program currently encompasses four projects, involving a total of 11 PhD students, one postdoc and one scientific assistant. The projects are executed at the universities in Amsterdam, Delft, Eindhoven, Leiden, Nijmegen and Twente, using a case from one or more of our partner companies. Thus, the researchers get to experience both the academic and the industrial practice. All projects are linked to running ESI activities.

Education

The students at the university level are the industrial engineers of the future. That's why it's also paramount for ESI to be involved in their academic education. We offer them a direct link to industry.

ESI plays this role in different ways. Our industrial cases are being used as teaching material in classes. Parts of our competence development program are being adopted by universities to train their students. Through internships, graduation projects and PhD projects, we give our future engineers a taste of the industrial practice.

Together with our partner universities, we're also working to improve the educational program. For example, while we see that systems engineering is a major trend in industry, academic training in this area is still rather lagging in the Netherlands when compared to regions like Scandinavia and the US. As part of the Next Generation High Tech program of the Dutch National Growth Fund, we're participating in a consortium with industrial partners, educational institutes and Brainport Development to create a continuous systems engineering development track from vocational education through higher education and universities to professional education.

The partner board

ESI's strategic direction and research program are defined in close consultation with our partner board. Currently, we have seven members from industry: ASML, Canon Production Printing, ITEC, Philips, Thales, Thermo Fisher Scientific and Vanderlande, five members from academia: Delft University of Technology, Eindhoven University of Technology, Radboud University Nijmegen, the University of Amsterdam and the University of Twente, plus TNO. This balanced composition puts us in an excellent position to connect industrial demand and scientific supply.

Our partner board has an advisory role. Leveraging the interests and insights from industry and academia, it has a big guiding influence on both the course and the contents of our research. Every two years, we sit together with our partners to discuss their market and business strategies and check these against current academic developments. Based on the outcome, we derive the innovation needs, which we then map onto our program lines and translate into concrete projects. When needed, it may also lead to the creation of new program lines or the redefinition of existing ones.

Every project typically includes one partner company and is carried out on their premises. This ensures a focus on a concrete industrial case. It also helps safeguard confidentiality when proprietary product information is being used. Projects usually run for multiple years, but at the end of each year, we evaluate the results and determine whether or not to add another year. Every couple of months, we get together for an intermediate assessment.

Commitment

With ESI's birth as a foundation, twenty years ago, the partner board started as a supervisory board. When we became part of TNO ten years later, the board moved into its current role. The resulting structure strongly resembles other external collaborations TNO has set up, like Holst Centre (with Imec) and Brightlands Materials Center (with the province of Limburg), albeit, in our case, with a group of partners.

The partner board has played a pivotal part all through ESI's history. The members' contributions provide us with the financial means to go all the way and make sure the work is also put to practical use. Thanks to our partner board, ESI is much more than a project organization, as it allows us to extend our activities from merely consolidating the results to also professionalizing, industrializing and disseminating them.

Some of our partners have been with us since the early days. This indicates that our collaboration gives them real value. In turn, these long-term commitments ensure that our research portfolio is always filled.

Benefits

The results of our joint work are available for reuse and as a basis for further extension and refinement in open innovation. Doing so allows us to boost the innovative power of the Dutch high-tech ecosystem as a whole. This way of working, which we refer to as the "knowledge multiplier," is one of the key benefits of partnering with ESI.

Through the ESI ecosystem, partner companies also have access to each other's local supplier and knowledge networks, enriched with ESI's international academic and industrial network. This "ecosystem multiplier" is a second key benefit of joining forces with us. Companies are stronger together than they are alone.
The third benefit of an ESI partnership is the "financial multiplier." Many a little makes a mickle: the financial contributions of many partners quickly add up.

ASML | Canon CANON PRODUCTION PRINTING | itec equipment + automation tech | PHILIPS

Radboud University | TNO | TU Delft | TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

THALES Building a future we can all trust | ThermoFisher SCIENTIFIC | UNIVERSITY OF TWENTE.

UNIVERSITY OF AMSTERDAM | VANDERLANDE

TNO

TNO, the Netherlands Organization for Applied Scientific Research, was founded in 1932 to enable business and government to apply knowledge. Regulated by public law, it's an independent organization, i.e. not part of any government, university or company. Employing over 3,600 professionals, it connects people and knowledge to create innovations that boost the competitive strength of industry and the well-being of society in a sustainable way.

Believing in the joint creation of economic and social value, TNO always seeks collaboration with partners. These partnerships focus on transitions or changes in nine societal themes. ESI is part of the Information and Communication Technology unit (TNO ICT)

ESI and TNO

With ESI's 20th anniversary, we're also coming up to our 10th anniversary under the wings of TNO. It has been almost ten years since we joined. During this time, we've grown from separate entities to a unity in which we've come to respect each other and are now extensively sharing knowledge, best practices and specialists.

ESI brought with it a way of working based on intimate relationships with companies and universities at a level that was of interest to TNO as well. Having such close ties to industry and academia, with a partner board providing guidance, was also new. Almost ten years later, these 'idiosyncracies' have gained broad recognition within TNO. They've included systems thinking, commercial excellence and customer-centricity – some of our best practices – as key focal points in their strategic plan for 2022-2025 and several TNO scientists are now participating in our system architecting acceleration program.

Meanwhile, ESI has retained its own identity. We're proving our worth, capitalizing on our partners and stakeholders and way of working while, at the same time, leveraging the power of TNO. We're doing more and more projects that combine our systems thinking approach with their technical domain knowledge, both in high-tech and non-industrial use cases. For example, in several of TNO's digital twin projects, our knowledge of model-based systems engineering is now being used in a wide range of application areas, while their data analytics expertise helps our diagnostics projects. In ten years, the integration of ESI into TNO has resulted in a great synergetic relationship – a true win-win situation.

Transition

Many thanks go out to the Dutch Ministry of Economic Affairs (EZ) for making this happen. It was the ministry's funding that got ESI going in the first place and that kept us going in the first half of our existence. And when they pulled out ten years ago, they cushioned the blow by providing us with a dowry that paid for our transition to TNO. Without EZ, we wouldn't be where we are now.

Many thanks also to TNO for taking us in and letting us share in their regular annual government funding. In the four years following our joining, TNO's board of management built up their financial involvement as the ministry's parting gift gradually ran out. If it wasn't for TNO's willingness to step in and take over from EZ, we wouldn't be where we are now either.

Last but not least, we owe many thanks to our partner board. Together with EZ and TNO, they made the transition happen. Not only did they supply valuable additional funding, but their invaluable guidance also supported us every step of the way.

TNO unit ICT

ESI is part of TNO's ICT unit. Focusing on digital innovations and digital transformation, this unit is a natural fit in terms of competencies and ways of working. Digitalization plays a key role in the complexity challenges of our stakeholders and the software-intensive methodologies we develop.

From the start, we were given free rein by the ICT unit to show our added value. In close coordination, we were enabled to grow into the TNO organization while maintaining our identity. Over the years, we've intensified our collaboration with the rest of the unit, leveraging its strong expertise base in fields like data science, IT systems, cybersecurity and distributed systems. In this atmosphere of trust and collaboration, we've been given the confidence and the space to follow our own path to success.

Open innovation

Open innovation is the process of combining internal and external resources for developing and bringing to the market new technologies and products. The use of the term has been promoted in particular by Henry Chesbrough from the University of California, Berkeley. The benefits and driving forces behind increased openness, however, have already been noted and discussed as far back as the 1960s.

Embracing open innovation isn't easy. Companies in the Dutch high-tech equipment industry have a long history of doing R&D on their own. With the increasing system complexity due to digitalization, they've come to see that this approach doesn't cut it anymore. They're all in the same boat – they all need to develop new methodologies to combat system complexity.

Open innovation offers a multitude of business advantages. Companies can keep focusing on their core activities. For what's non-core, they have access to up-to-date knowledge through a high-quality shared R&D competency. As a bonus, such open-innovation initiatives with industry-wide benefits are usually eligible for government funding, making them also financially attractive.

Swayed by the benefits, the Dutch high-tech equipment industry has embraced open innovation. At ESI, we support the companies' collaborative efforts through our research and network. We link them with our partner universities and international peer institutes and, like a flywheel, drive the broader adoption of the results and the growth of knowledge.

The ESI model

ESI's partner board plays a key role in our open-innovation model. From its input, we derive the industry's innovation needs. Together, we create and regularly update a common agenda and roadmap.

The partners have agreed to freely share all the results not specifically tied to one of their products. This sharing is facilitated through various networking opportunities, such as the ESI symposium and different webinars and workshops. We also organize special interest groups, where professionals from mixed backgrounds discuss a variety of topics, including system architecture, system integration and software legacy. Our open-innovation ecosystem isn't confined to our partner board. We involve all stakeholders. This includes our implementation partners and other service companies and tool vendors. Thus, we create a complete chain from an initial idea to a sustainable solution that has societal and industrial impact.

Our model brings three main benefits. Knowledge gathered by one is available to all (the knowledge multiplier). Individual ecosystems are connected to form a powerful network allowing the knowledge to be shared and reinforced (the ecosystem multiplier). The costs of research into common solutions are shared (the financial multiplier).

This approach has proven to be successful. The secret to this success is mutual trust. Open innovation only works when the participants trust each other and see each other as partners instead of competitors.

History

Compressing 20 years of ESI history into a few pages is a challenge. After all, many people have made a valuable contribution to what ESI is today, often from their positions in companies, universities, NGOs or public authorities. Frans Beenker, who has been involved with ESI from the very beginning, recounts various developments that played a decisive role from different angles. Broadly speaking, the history is divided into three periods of seven years: the construction phase, the growth phase and the maturation phase. This overview aims to give an idea of the course of events without presenting an illusion of completeness.

Construction phase from 2001

The Dutch Ministry of Economic Affairs stimulates innovation using natural gas revenues in the Economic Structure Enhancement Fund (Fonds Economische Structuurversterking, FES) program. The idea arises to establish a national institute in the field of embedded systems. The starting point is the Eindhoven Embedded Systems Institute (EESI), founded by TU/e in 1998. With the cooperation of TU/e, TU Delft, University of Twente, TNO, Philips, ASML, Ericsson, CMS and Océ, the first initiatives are undertaken in August 2001 under the leadership of Martin Rem. EESI is transformed into an industry-driven program.

On 21 August 2002, the establishment of the ESI Foundation becomes a reality. Led by three directors – Martin Rem, Marloes van Lierop and Frans Beenker – ESI builds up its own group of researchers with academic backgrounds and experience in industry. The characteristic working method, 'industry-as-a-lab,' comes into being. In 2002, the first research project, Boderc with Océ, begins in the Technological Cooperation subsidy scheme, followed by Tangram and Ideals in collaboration with ASML. Through the Investments in Knowledge Infrastructure Decree (Besluit Subsidies Investeringen Kennisinfrastructuur, BSIK), ESI receives a €25M subsidy in 2004. The ESI Foundation is managed by a supervisory board consisting of representatives from universities, industry and TNO. This board is decisive in the success of ESI. The mission, vision and strategy are determined and a one-page strategy is formulated with clear KPIs. Academic researchers remain on the payroll of the participating universities. The ESI profile is put in place.

From the outset, projects are carried out at the customer's location in an open-innovation setting: the sharing of results, academic research as part of the project and a contribution from industry. ESI is responsible for the agenda, project execution and outcome. Experienced project leaders from industry are hired.

Many projects in collaboration with industry and academies follow: Trader, Darwin, Falcon, Condor, Poseidon, Octopus and BSIK007 with respective partners Philips Semiconductors (later NXP), Philips Healthcare, Vanderlande, FEI (now Thermo Fisher Scientific), Thales, Océ (now Canon Production Printing) and ASML. Knowledge sharing receives attention through a competence development program, which still benefits system architects from the sector to this day.

Chairs of the Supervisory/Partner Board

2002 Wim Orbons (Océ)

2004 Martin Schuurmans (Philips)

2009 Anton Schaaf (Océ)

2013 Harry Borggreve (ASML)

2015 Ton Backx (TU/e)

2020 Harry Berghuis (Philips)

2022 Paul Hilkens (Canon Production Printing)

Growth phase from 2008

Whereas in the first seven years, the institute is established, the organization is arranged, the staff is appointed and the processes and funding take shape, the second period of seven years is targeted at growth. The role of the Ministry of Economic Affairs and the Netherlands Enterprise Agency (formerly SenterNovem) contributes significantly to the success of ESI. Financial and strategic support are each decisive in the establishment of innovation programs and in obtaining funding for innovation. The BSIK funding is followed by funding from the COMMIT program, which continues the collaboration between industry and academia.

ESI initiates a review of the research by an international advisory board that includes experts from leading organizations. ESI can be proud of the conclusion of the first external review in 2011: “ESI received excellent results and all goals have been realized. There are no institutes worldwide that manage these aspects of high-tech systems engineering at such a high level. ESI is recognized for its contribution, creates impact and is valued by its partners.” In 2013: “High-impact and practice-oriented research in support of the design and construction of complex high-tech systems, with an open-innovation approach. The industry-as-a-lab concept has a proven track record on valorization.”

The Dutch innovation approach is changing. Since 2007, ESI has been an active part of Point One and the established High Tech Systems Platform with figurehead Math de Vaan (Berenschot), later succeeded by the highly engaged Amandus Lundqvist (TU/e). The Technological Top Institutes (TTI) scheme ends and is replaced by the Top Consortia for Knowledge and Innovation (Topconsortia voor Kennis en Innovatie, TKI) scheme.

This removes the existing financial basis for ESI. With the cooperation of the HTSM Top Sector, the Ministry of Economic Affairs and TNO, ESI becomes part of TNO on 1 January 2013. The name of ESI is retained, although the I of Institute is changed to Innovation.

The first ESI Symposium takes place in 2008 with keynotes Lothar Thiele (ETCH) and pillar since the very beginning Anton Schaaf (Océ). An ideal opportunity for networking with an appealing theme and workshops and demonstrations focusing on the latest developments.

Management

2001
Martin Rem
Marloes van Lierop
Frans Beenker

2005
Reinier van Eck
Ed Brinksma
Frans Beenker

2009
Boudewijn Haverkort
Reinier van Eck
Frans Beenker

2013
Reinier van Eck
Frans Beenker

2016
Wouter Leibbrandt
Frans Beenker

2022
Wouter Leibbrandt
Jacco Wesselius

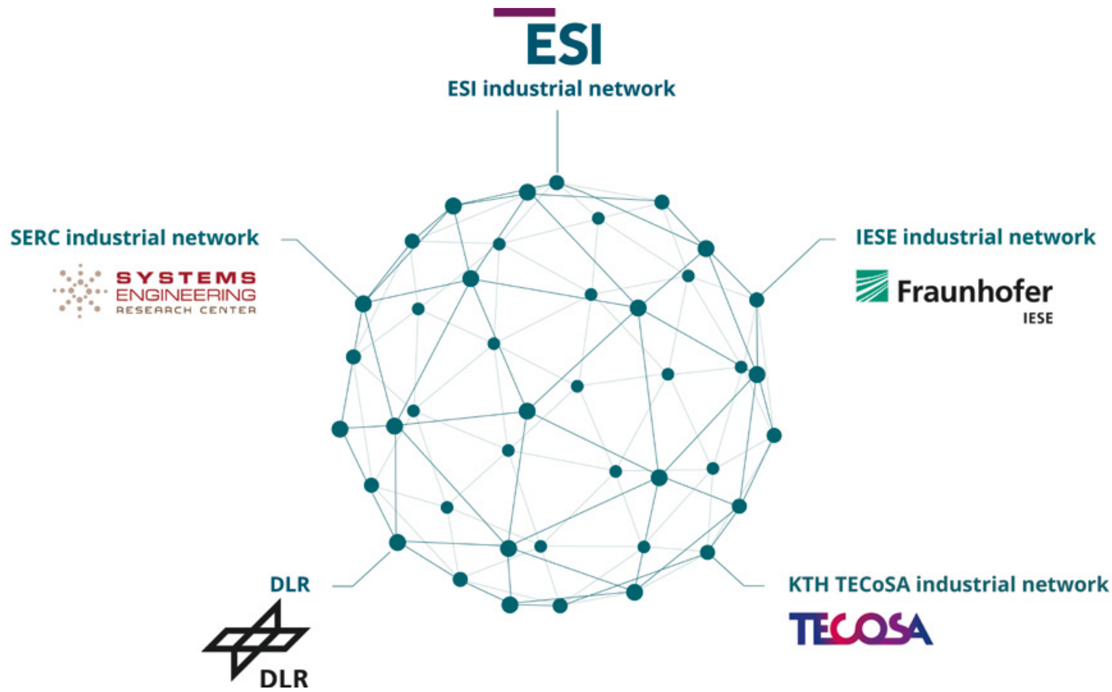
Maturation phase from 2015

ESI strengthens its positioning nationally and internationally. Systems engineering receives its own roadmap within HTSM. The international network is expanded. Exchanges with the Stevens Institute in the US (now affiliated to SERC), led by Dinesh Verma, are frequent, particularly in the field of modeling. In 2016, the first MoU is signed with Fraunhofer IESE from Kaiserslautern (Germany). The mutual industrial backers meet during a number of workshops. The network expands with OFFIS/DLR (Germany) and KTH TECoSA (Sweden). The term “embedded engineering” is gradually replaced by “systems engineering,” a key technology with its own roadmap within HTSM.

In 2017, ESI moves from the TU/e campus to the High Tech Campus, a hotspot of innovative companies in the middle of Brainport, referred to as the smartest square kilometer worldwide. The ESI ecosystem expands. Parallel to the research program, training courses are given and experts exchange knowledge via SIGs. The ESI Symposia, organized once every 1.5 years, give a boost to the network. The number of participants grows. In 2021, due to the measures taken as a result of Covid-19, the tenth symposium is transformed into a successful, international online version with the cooperation of the network.

Methodologies and tools are professionalized and maintained on the basis of open source. TNO becomes a member of the Eclipse Foundation and enters into a collaboration with Obeo in France. The first implementation partner, Capgemini Engineering, joins in 2021. Due in part to this, results are consolidated and transferred in an open and professional innovation setting.

ESI is still growing and there are numerous challenging developments, as can be seen further ahead in the chapter on the future of ESI.



International network

- Stevens Institute and Systems Engineering Research Center (USA) – focus on systems engineering methodologies
- Fraunhofer Institute for Experimental Software Engineering (Germany) – focus on sharing agendas and results, joint workshops with local industry
- OFFIS/DLR, currently the DLR Institute on Systems Engineering for Future Mobility (Germany) – focus on verification, validation and autonomous systems
- KTH TECoSA (Sweden) – focus on trustworthy edge computing systems and applications

System performance engineering

Wafers per hour on a litho scanner, pages per minute on a printer, time to resolution on an electron microscope – system performance is the amount of useful work done by a system, measured as the production speed of products of a predefined quality. For many systems in our ecosystem, this is a key aspect, often bringing a competitive advantage.

To meet market demands for product quality, product customization and total cost of ownership per product, systems need to meet ever more ambitious performance targets – relating to productivity but also to aspects like product quality, cost, reliability, security and customizability.

Challenges

Because system performance is such a central aspect, it should receive explicit focus across the full system lifecycle. Designing for performance, as this is called, is essential for making the right design decisions and for optimizing operational performance. It addresses the challenges in today's practice, where performance problems often materialize late – during integration or, later still, during use.

Designing for performance is far from straightforward. Architects have to think about how to cover all realistic use cases and how to quantify performance requirements with incomplete information. They also need to consider how choices in the system decomposition affect performance.

Designers and engineers also face many challenges. Languages, methods and tools are often targeted to a single discipline or specific domain. The performance challenges in system design, however, are cross-cutting and require a holistic view. Currently, there's a lack of solutions that can be reused in multiple disciplines and domains during early system design.

Tackling the challenges

System performance engineering (SysPE) focuses on tackling these challenges. In our ESI system performance program, we aim to develop industrially mature, domain-specific model-driven SysPE methodologies based on the latest academic results. Models act as a single source of truth and form a basis for the automated generation of implementation artifacts like schedulers or controllers.

With accurate system-level performance models, the right design decisions can be made during the early development stages. This leads to less rework later and minimizes over-dimensioning. It also improves performance for system variants and operating conditions by taking into account variability and system context.

Where are we today?

At ESI, we've been actively working on model-driven SysPE for almost 20 years. By applying the latest academic results and tailoring solutions to industry, we're developing methodologies that can be used across companies facing similar performance challenges. Based on our experiences and feedback from the international performance engineering community, we've created an overview of the SysPE field, structured around five focus areas, which together cover the full system lifecycle. For each focus area, we've identified best practices.

An example of model-driven SysPE is our approach to monitor performance targets during system operation and diagnosing unexpected performance degradations using domain models inferred from execution traces. This approach is explained in more detail on pages 28-31.

Another example is our LSAT performance engineering approach for flexible manufacturing systems. By capturing the essential delays, LSAT makes it possible to efficiently analyze activity sequences for performance properties, allowing designers to effectively explore system productivity. This approach is explained in more detail on pages 32-35.

Clearing the critical software path

For a highly complex machine like the TWINSCAN to be able to operate smoothly, its system control should run without any unnecessary interruptions. Within the Concerto project, ASML, ESI and TU/e have developed a model-based methodology to analyze the software execution and keep computational tasks out of the critical path as much as possible. The partners see great potential for the approach to be widely adopted in the high-tech industry.

In ASML's lithographic systems, the TWINSCAN stage simultaneously moves two tables, each holding a silicon wafer. While one wafer is being exposed to – deep or extreme – ultraviolet light containing the chip pattern to be printed, the other is measured by the machine's metrology sensors to optimize alignment. The tables are propelled electromagnetically, allowing frictionless acceleration as high as 7G.

Every move the TWINSCAN stage makes has been precisely calculated by the system's software. To ensure a smooth journey from A to B and prevent a wafer table from missing a turn, the computations need to be completed in time. "Imagine you're on the highway, following the instructions of your navigation system," ASML's Jos Vaassen makes a comparison. "If the system takes too much time to calculate the route, you're going to drive right past your exit. Likewise, we don't want our scanners to miss a turn because our software is missing a deadline."

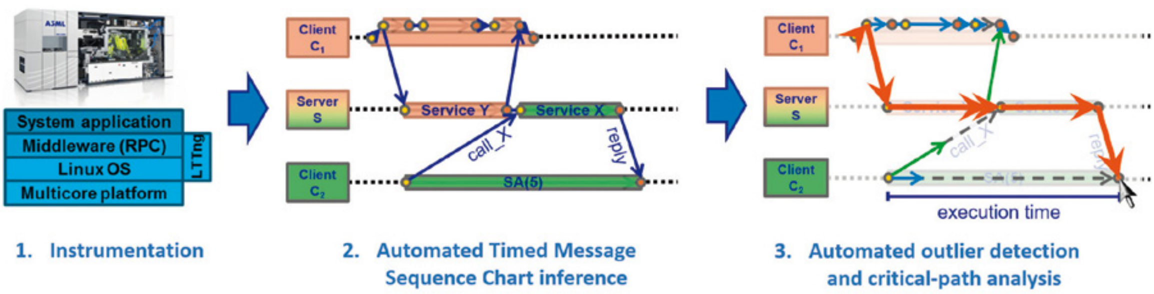
As the chip patterns to be printed continue to shrink, however, the lithographic scanner grows ever more complex, requiring an increasing number of computations to get the job done. This raises the likelihood of missing a turn and having to stop for some time to recalculate and get back on track. Such an interruption has widespread consequences. For example, it affects the focus of the lens system and the alignment of two subsequent chip layers. Ultimately, it will impact the machine's performance.

To prevent that from happening, the computations, realized in software, need to be continuously monitored. With that goal in mind, ASML and ESI set up the Concerto project in 2016, together with Eindhoven University of Technology (TU/e). In four years, they developed a model-based methodology to diagnose, predict and optimize system timing and throughput and to keep computational tasks out of the critical path as much as possible.

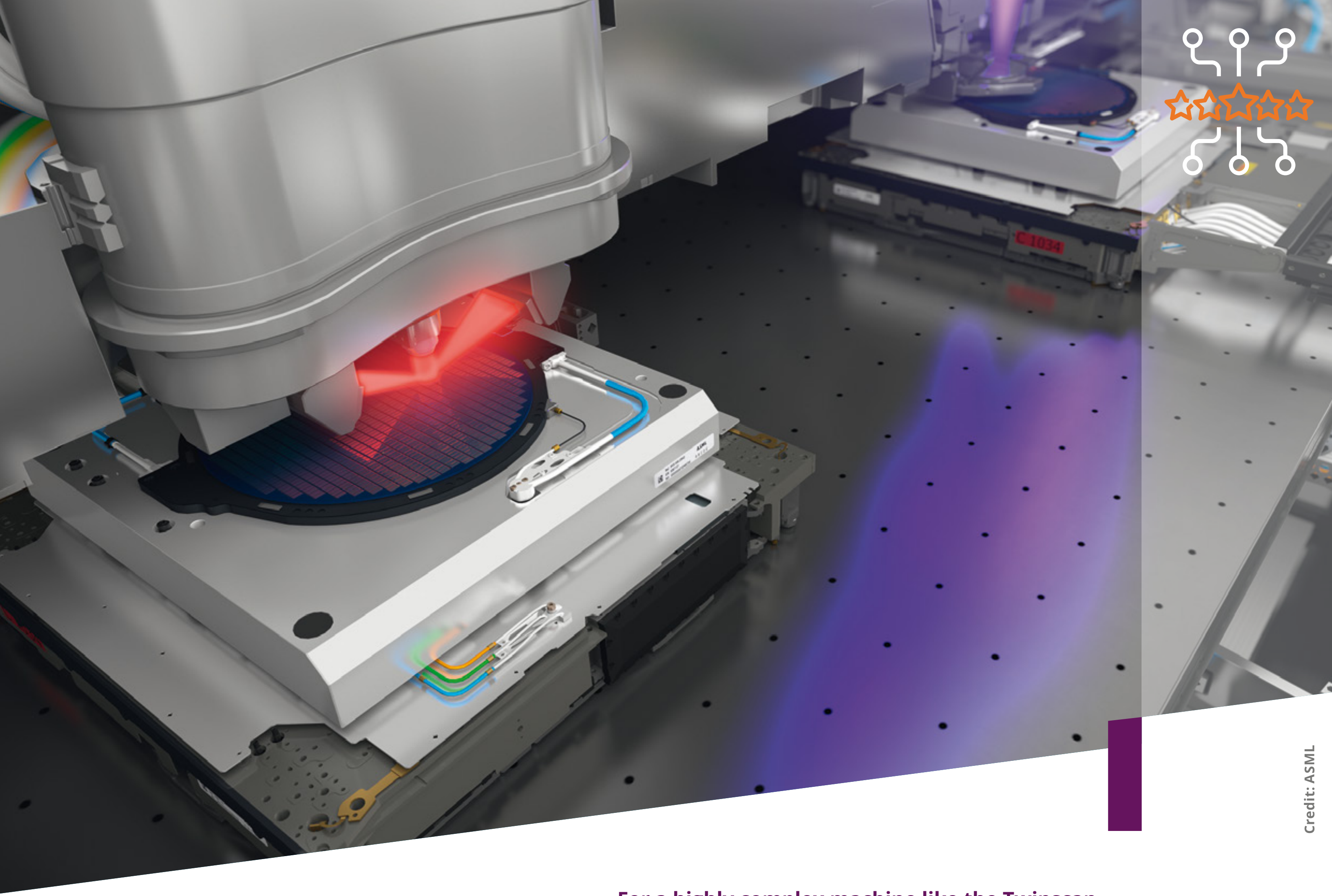
Performance bottlenecks

"A TWINSCAN machine contains a great number of software components," explains Jeroen Voeten, professor Cyber-Physical Systems at TU/e and initiator of the Concerto project when he worked as a research fellow at ESI. "All those components are doing their share in completing the computational work at hand. Finding out which of them is causing the delay when the calculations are taking too much time is a daunting task, almost

Credit: Concerto project



The measurement-based approach developed by ASML, ESI and TU/e to obtain insight in the system runtime behavior.



Credit: ASML

For a highly complex machine like the Twinscan to be able to operate smoothly, its system control should run without any unnecessary interruptions.

impossible to do manually because of the sheer number of components. The tooling developed in Concerto makes it possible to quickly pinpoint the root cause. Knowing this cause is the first step in fixing the problem."

A delay in machine execution can be due to a simple software error. A problem like that is generally easy to fix, says Joost Gevers, ASML's software product architect responsible for the installed base of NXT systems in the field. More challenging are delays caused by a tight processing budget, i.e., when the system's to-do list is pushing the limits of the available compute power. "When timing budgets aren't met, we could remove some of the computational tasks from the critical path by executing them earlier."

Fellow software product architect Vaassen, during Concerto responsible for the installed base, emphasizes that fixing the problems was outside the scope of the project. "The tooling developed focuses on finding the critical path and the components on it. Once that has been mapped out, it's up to the engineers to resolve the performance bottlenecks."

"We've developed a model-based approach to do the root cause analysis," recaps Bram van der Sanden, ESI research fellow and liaison between ASML and TU/e. "The tooling constructs an overview of the system's execution over time, showing which component is performing which task at what moment. By analyzing this execution, it can then show the locations of the bottlenecks. This provides insight into the design changes that need to be made to ensure a smooth operation."

Formal foundation

The approach developed within Concerto starts with collecting data about the system execution. "We do that by instrumenting the machine software at strategic locations, i.e., by adding little bits of measurement code there," clarifies Van der Sanden. "For every component, it allows us to track the starting and stopping times of the functions being executed and the messages being passed to other components. This gives us the traces we need to automatically create the formal behavioral models with which we can analyze the system's performance."

"We're not instrumenting components individually," underscores Van der Sanden. "We're instrumenting function interfaces and the middleware through which they talk to one another." TU/e's Voeten adds: "Because ASML has a nice component-based architecture with a decent middleware layer, that communication is readily accessible for automatic instrumentation. That's very important – doing it by hand would be too much work and it would be very hard to get the required information out."

"As we're looking to find timing bottlenecks, it's also key that we don't interrupt the realtime performance ourselves," Voeten goes on to point out. "So we've made sure the instrumentation is as non-intrusive as possible, having negligible impact on the system's operation. Thanks to a very efficient implementation, it's now ready to run on systems in the field." ASML's Vaassen: "We can't have our customers losing productivity because we've added instrumentation to our software, which is why we really took our time to minimize the impact."

From the log data, the Concerto tooling generates so-called timed message sequence charts. These are Gantt-like diagrams, plotting the software components against the functions they execute over time, supplemented with arrows depicting task dependencies. "The TWINSCAN's highly repetitive work, for example, is clearly reflected as frequently reoccurring task groupings," illustrates Voeten. "The charts map out the complex interplay between components in an insightful way. They perfectly fit ASML's architecture, and software architects and engineers are already used to working with them in the system specification phase – they're very common in design documentation. Contrary to standard practice, however, we're generating them, after the fact, from execution data."

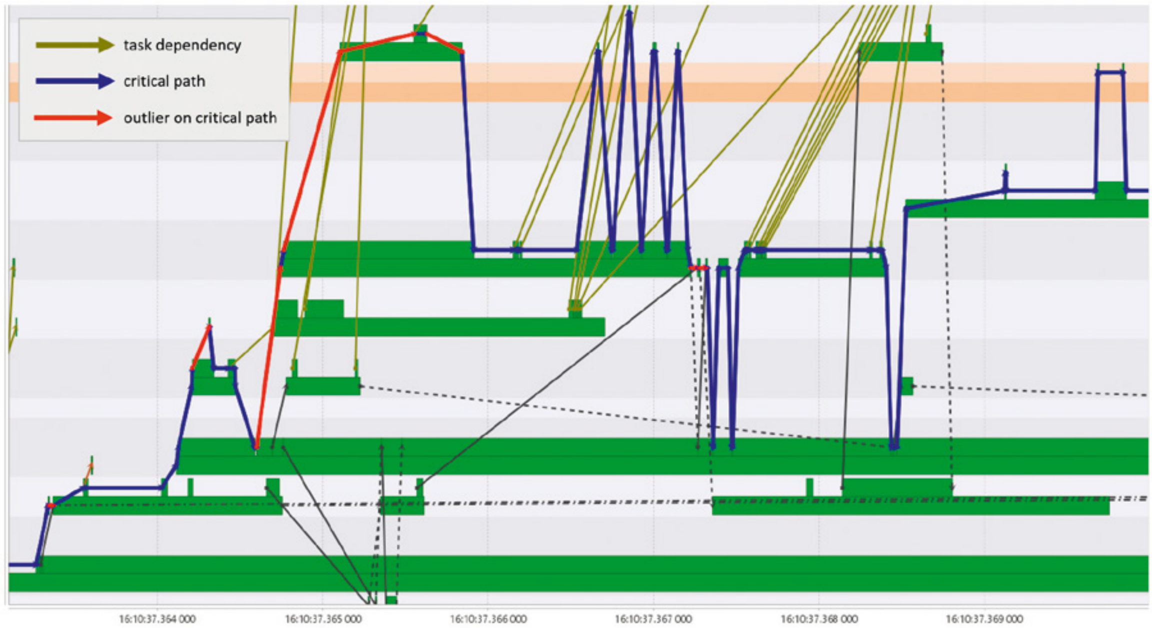
The timed message sequence charts provide the formal foundation for the final step of automated performance analysis. "We can apply different mathematical techniques to them," notes Voeten, "not only to calculate the critical path and find the root causes of bottlenecks but even to formally verify system properties."

Major milestone

The tooling is being industrialized at ASML. "We haven't fully deployed it yet but in the pilot phase, it has already helped us uncover a couple of bottlenecks," states Vaassen. "For example, when two tasks communicate, they can do so on a fire-and-forget basis: they send each other messages and continue their business without waiting for a reply. When the message is too big to be conveyed in one go, however, it gets chopped up into multiple parts, which the sender can't just fire and forget anymore; it has to wait for a reply before it can send the next part. Thus, fire-and-forget can still result in a task being blocked. Since the interface for sending messages is abstract, this isn't visible in the code. Concerto has really opened our eyes to these kinds of potential problems."

"We don't have to rely so much anymore on good fortune and in-depth knowledge to find a bottleneck in a day or two," summarizes Vaassen what he sees as the project's main added value. "Without having to dive into the design documentation to determine the exact configuration, we can get an overview of what's going on in a system. The tooling can just generate that by looking at the execution. It accelerates problem-solving."

Vaassen's colleague Gevers concurs: "By giving us the complete picture, it allows us to more easily pinpoint performance bottlenecks. We have an excellent proof of concept, showing that it really works. I'd like to see ASML invest big in rolling it out to the company's entire software community – with the ultimate goal of using it to analyze execution data collected in the field and fix issues at customers in a heartbeat."



Credit: Concerto project

From the log data, the Concerto tooling generates timed message sequence charts, plotting the software components against the functions they execute over time, supplemented with arrows depicting task dependencies.

To Voeten, the project marks a major milestone for the model-based paradigm. "For decades, we've been trying to get the industry to create models for specification and code generation – to little avail. With Concerto, we've moved to automatically generating them from complex systems – the right side of the V model – and we're already gaining traction. We've managed to connect twenty years of academic research to the high-tech practice. Bringing the ability to efficiently analyze millions of lines of code in a day, I think this has real potential of catching on."

ESI's Van der Sanden, too, sees great benefits for the ecosystem. "We've developed a model-based methodology to quickly and systematically assess the impact of timing variations. We intend to open it up to other companies. Those with a similar component-based architecture, such as Philips or Thermo Fisher Scientific, could benefit from it as well."

What-if scenarios

Meanwhile, the Concerto partners have teamed up once more in a follow-up project, called Maestro. "We started at the end of 2019, again for four years," tells Van der Sanden. "One of the aspects we're working on is raising the abstraction level and going from software to system tasks. By enriching the generated Concerto models with multidisciplinary domain knowledge, we want to be able to do a machine-level diagnosis, pinpoint the bottlenecks in system functionality and then zoom in and run a root cause analysis on the associated software tasks."

Collaborators also want to loop back to the left side of the V model, the system specification. Returning the message sequence diagrams to their natural habitat, so to speak. "After that abstraction step to describing system activities, we're looking to take it one step further, to specifying system behavior and using that in the development process to make predictions," philosophizes Van der Sanden. "That would allow us to ask questions like: what would be the impact on system timing if we were to change the order of tasks or even add some new computations? Would that require us to add more processing power or are there other ways to keep the system on track? Being able to run what-if scenarios like this is another long-term objective of Maestro."

The system conductor

Eindhoven University of Technology, ASML and ESI are working on LSAT, a model-based tool that bridges the gap between systems engineering and the mono disciplines. Especially during the first design phase, this is of the utmost importance for the optimization of the overall system behavior and the supervisory controller. Through formal methods and domain-specific languages, LSAT quickly provides insight into which architecture will be most productive. The three partners hope to set a new industry standard with the tool.

Flexible manufacturing systems can be seen everywhere, from the semiconductor industry to car factories. What these systems have in common is their complexity. This is often because they have to be able to adapt to different part types, strict timing requirements, a large number of production steps and a platform with many shared resources. For the process to run smoothly, all those modules and components have to play together like a symphony orchestra.

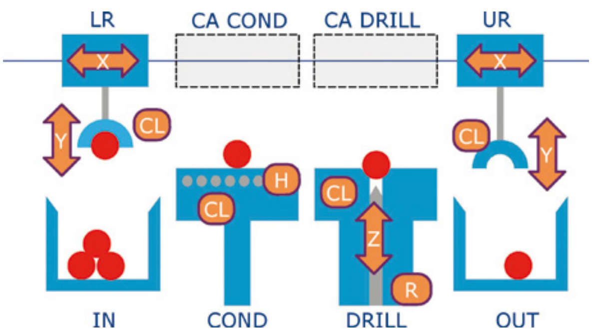
The performance of such a flexible manufacturing system is highly dependent on the configuration and conducting capabilities of the controller. You won't achieve the highest possible output just by putting your faith in the knowledge and skills of the system architect. It simply takes too much time to manually calculate all design options. Moreover, it's extremely difficult to make solid statements about unique systems that have never been designed before, purely based on past experiences and a series of set rules.

About ten years ago, Eindhoven University of Technology (TU/e) and ASML put their heads together to tackle this problem. Later on, ESI and ICT-NL worked on the implementation of a tool and the research was expanded in the ESI Concerto and NWO RCPS programs, and at the European level in Arrowhead Tools. The result is a model-based development tool with which designers can specify system behavior via domain-specific languages and optimize them at an early stage for the throughput of the final machine. The package is called LSAT, which stands for Logistics Specification and Analysis Tool.

Provably the best

"LSAT is a tool that allows you to quickly explore all design options and discover in the first phase of the development process which system configuration will yield the highest throughput," says Jeroen Voeten, professor Cyber-Physical Systems at TU/e. Wasn't there already a solution in the large toolbox available to designers today? "In system control, you have different layers. Mechatronics typically deals with movements from A to B. Above that is a layer for coordination and above that a layer for planning. The higher you go, the less standardization there is. Most companies do have a supervisory control layer, but they all have chosen a different approach. And that means there are no uniform modeling tools that can deal with this problem."

"One of the best-known tools for model-based design of dynamic systems is Simulink from Mathworks," Voeten continues. "Very widespread in the mechatronics world because it allows you to optimize for things like speed and accuracy of individual robot movements. LSAT hits a higher level where it's about the coordination of all these movements. That's really something different. It's about productivity and all the design decisions you make to optimize it."



To provide insight into the application of LSAT, the researchers use the fictitious Twilight system, which represents a very simplified version of a wafer handler.

Credit: ITEC

At ITEC, developers have applied LSAT to model die bonders.

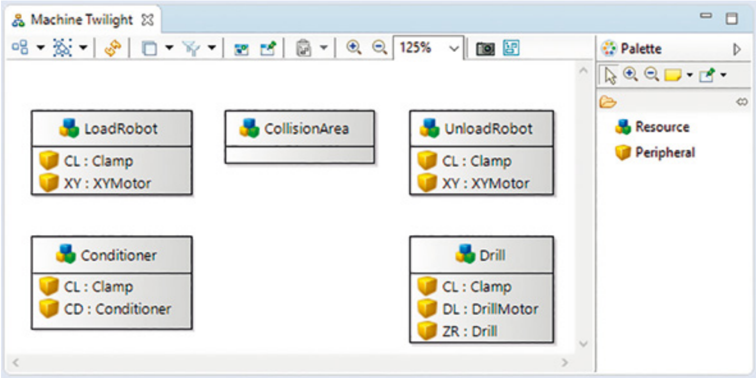
This optimization is done with formal methods, Voeten explains. “These are mathematical techniques with which you can provably determine the best solution. That’s a step further than simulation, in which you only play out one system configuration. With LSAT, you look at all possibilities and calculate which one will get you the best result.”

DSL

Optimization with formal methods is a subject perfectly fitted for the researchers at TU/e. After all, it’s a fundamental technology for which you only need to have limited knowledge about the domain for which you’re developing it. Within the multidisciplinary partnership, this topic is entrusted to the scientists in Eindhoven.

ESI is closer to the industry and focuses on the domain-specific languages (DSLs) needed to describe the design in jargon that LSAT understands. “In a generic language like Java or Python, and also in Simulink, you can write anything you want – as a designer, you have enormous freedom,” says Bram van der Sanden, a research fellow at ESI. “But the more possibilities you have, the harder it is to properly analyze the result. Within LSAT, we work with a domain-specific language, in this case focused on flexible manufacturing systems. To give an example, in ASML’s wafer handler, engineers talk about wafers, loading and unloading robots, and setpoints. These are typical model elements that you don’t see in a generic language but are normal in a DSL. As a result, a designer can use the jargon from his day-to-day work and doesn’t have to become a software expert. Even non-specialists can quickly write down what they want to achieve because it’s in the same language they speak every day.”

A DSL contains a number of architecture rules. “You can describe exactly what is and what isn’t allowed,” says Van der Sanden. “When drawing up the model, users receive much more support because they get immediate feedback if they do something that’s not allowed. And because they can set up their model much faster, they can cover a much larger design space in an efficient way.”



During the modeling process, the peripherals are assigned to the system resources. The collision area doesn't have peripherals, but the two robots can claim and release it to prevent collisions.

Twilight

To provide insight into the application of LSAT, Voeten and Van der Sanden, together with co-researchers Yuri Blankenstein from ESI and Ramon Schiffelers from ASML, use the fictitious Twilight system, which represents a very simplified version of a wafer handler. Twilight consists of four main parts: a loading robot and an unloading robot that transport balls, a conditioning station that heats the balls to the correct temperature, and a processing station that drills a hole in a ball. The balls must pass both stations in the correct order and of course, the robots mustn't collide.

"To arrive at the optimal configuration and the highest throughput, we need to do two things: model and analyze," explains Van der Sanden. "First, we model the platform: we specify the peripherals such as grippers and motors. We then assign these to system resources."

All movements are then given a profile so that LSAT knows exactly how much time it has to allocate for them. "These are the building blocks with which we can model all kinds of actions in the system," says Van der Sanden. "For example, to get a ball from point A to point B requires a whole recipe of partial actions that have to be performed successively or simultaneously."

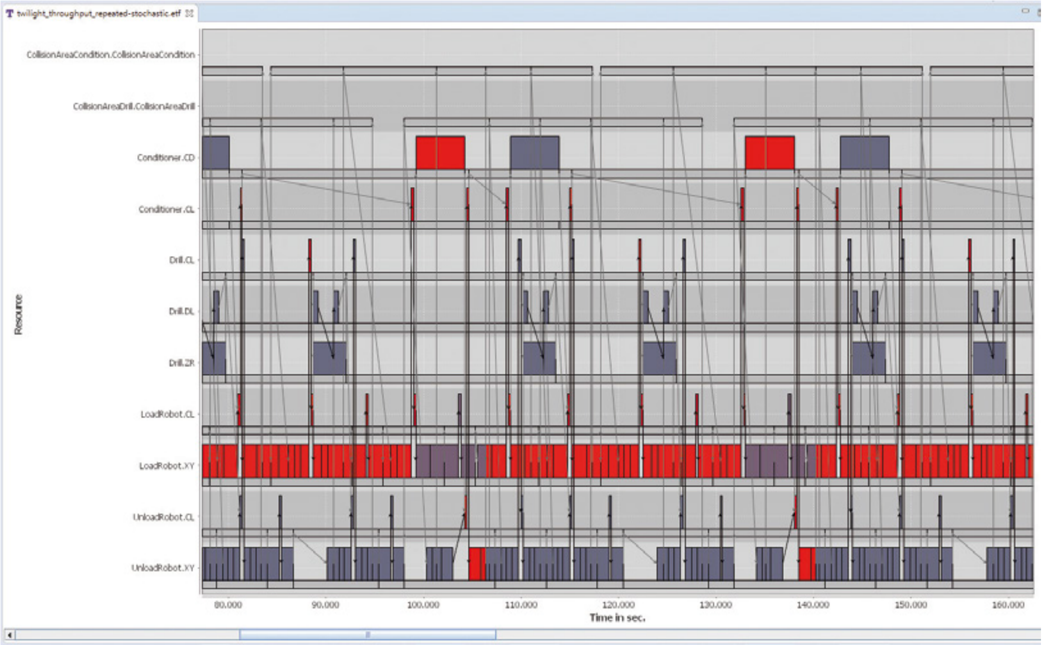
When all actions have been entered, LSAT determines the optimal order for the best possible product flow. This is how the system will have to be programmed. The tool also displays the result visually in a Gantt chart. Van der Sanden: "This shows you the critical path and the resources that might form a bottleneck. Within Twilight, that turns out to be the loading robot. If you want to speed up the process further, it might be wise to take a look at that part first. Maybe you can choose a faster robot. Or you may need to introduce a third robot into the system. The great thing about LSAT is that you can quickly calculate such an alternative by updating the model. The building blocks are already in place."

ITEC

Twilight is a relatively simple and small system, but LSAT works just as well for very complex machines. For example, ASML has used the tool in the development of the wafer handler that processes the flow of silicon wafers to and from its lithography machines. With LSAT, the developers in Veldhoven evaluated what effect changes to the mechanical platform and the supervisory controller had on the system performance.

Voeten: "You can also use the LSAT model as a specification. Those are often written in Word files, but you can't execute them and they're full of ambiguities. The LSAT model is an excellent blueprint for the rest of the development." This is precisely what ASML has done in its collaboration with VDL ETG. To develop and build the wafer handler, VDL ETG used ASML's LSAT models to create UML diagrams and generate code.

At ITEC, developers have applied LSAT to model their die bonders. This gave them insight into the critical path, budget and productivity. "Modeling systems in LSAT leads to more complete specifications because the dependencies between components must be made explicit," said Sam Lousberg, at the time a mechatronics engineer at ITEC.



The results in a Gantt chart provide information about the critical path and show what resources might be the bottleneck.

Open source

"LSAT does scale up to very large systems, but there's a limit to the state space that the tool can handle," Voeten admits. "In general, you can easily solve this by constraining the state space manually – for example by defining more boundary conditions – or by dividing the process and optimizing each piece separately."

Voeten believes that LSAT is an ideal multidisciplinary tool. "All machine builders who want to improve the logistics in their machine and increase throughput can get started with LSAT," he says. "It's a lightweight tool that doesn't require every company to reinvent the wheel. We've already seen that it works in the ASML domain and the VDL ETG domain. The assessment at ITEC gives us a lot of confidence that it's more widely applicable."

ESI is currently working hard to make the tool available in open source through the Eclipse Foundation. Van der Sanden: "We want to move toward an ecosystem where companies can make their own additions, and where we as researchers can launch new optimization and analysis techniques. A joint tool that everyone can benefit from."

The ambition is even greater. Voeten: "We'd like to see LSAT become the industry standard. We have all kinds of extensions on our roadmap for this. Ultimately, we want you to be able to specify the entire supervisory control layer, including exceptional behavior, and then generate code directly and automatically from the LSAT models. That's the dream. We'll need another ten years before we've achieved that though."

Software legacy and rejuvenation

As industrial software evolves over many product generations, with a great many people contributing to it, it grows in size and complexity. This results in large legacy codebases that are hard to oversee and maintain. The maintenance efforts are even growing at the cost of developing innovative features for customers. The expertise area of software legacy and rejuvenation aims to support the developers of industrial software in understanding their codebases and (semi)automatically improving them and reducing the accidental complexity.

Challenges

Because of the sheer size and complexity of the codebases, it's extremely challenging to keep an overview, even for experienced developers. At the same time, it's incredibly difficult for new people to gain insight and get up to speed. Knowledge about the code may even get completely lost as original contributors move on to other projects or even other companies. As a result, developers spend more and more time on analysis – leaving them less and less time to create added value.

Industrial software has also come to contain loads of accidental complexity. Over the years, a lot of code has been added that has no real functionality or that even obfuscates functionality. This can be because the software was repurposed without proper insight into the intent of the original developer or because multiple contributors have different perspectives on that intent. As a result, codebases capitalizing on legacy run the risk of becoming much less coherent.

Tackling the challenges

Industrial legacy software can be scrutinized and rejuvenated using static code analysis – like going with a comb through the lines of code and untying the knots. An expert in the codebase first specifies the patterns to be improved (the knots to be untied) and how to improve them (what patterns to use instead). In subsequent iterations, alternating analysis with transformation, the codebase is processed. The result of the analysis can be a summary of the findings or a graphical representation of the dependencies, but also actual input for a tool that automatically transforms the code accordingly.

Central to this method is the expert in the loop. In the analysis phase of the loop, information is extracted from the code. In the transformation phase, the code can then be rejuvenated by replacing problematic patterns with improved ones.

Where are we today?

At ESI, we've successfully applied this methodology in several projects. Together with Thermo Fisher Scientific, we analyzed part of their codebase and automatically extracted graphs, allowing for the straightforward assessment of dependencies and the subsequent disentanglement. This project is explained in more detail on pages 38-41.

In collaboration with Philips, we analyzed and transformed code that implemented some nested state machines. We first ran a small automated refactoring to simplify the code. Next, combining analysis and transformation, we extracted a model, transformed the model, inferred from this how to modify the code and then changed the code automatically.

In another project with Philips, we semi-automatically redesigned an organically grown industrial adapter component written in C++. Guided by a reference design, we extracted a legacy model from the code that captured the functionality in a form close to the original structure. We then flattened the model with a focus on external interfaces. By analyzing the flat model's variation points, we constructed a new model for model-based code generation.

ESI helps Thermo Fisher Scientific and Philips grease their software machines

When new feature development for its electron microscopes started to get hampered by older software, Thermo Fisher Scientific called on ESI to clean up its code and make it future-proof. The approach developed in their public-private partnership was also successfully adopted by Philips to tackle the legacy challenge in its medical scanners.

Credit: Thermo Fisher Scientific

A highly advanced system such as an electron microscope contains a boatload of software to make it work.

Highly advanced systems such as an electron microscope or a medical scanner contain a boatload of software to make them work – millions to tens of millions of lines of program code. This software is the product of years, sometimes even decades, of development by a team that constantly changes shape. New engineers come in, while others move on, leaving their work behind as a legacy to be cared for by the rest.

As this legacy codebase grows, it's getting increasingly difficult to maintain an overview of the inner workings and keep the software well-structured. "Most of the code was written a long time ago and over the years, many people with advancing insights have contributed. The result is less than optimal, to put it mildly. Layer upon layer, reflecting the software engineering progress the company has made through the years," explains Dennis Dams, a senior research fellow at ESI.

The lack of overview and structure makes it increasingly hard to add new functionality – a challenge recognized all too well at Thermo Fisher Scientific in Eindhoven. "The software of the electron microscopes we develop here uses old interface technology to enable subsystems to communicate with each other," illustrates senior manager software development Arjen Klomp. "We used it all over the place, the code was practically polluted with it. This hampered our new feature development and brought down the efficiency of our software engineers to such an extent that we needed to do something about it."

Thermo Fisher Scientific called on ESI for help. Together, they embarked on two consecutive public-private research projects to develop a novel approach to tackle the legacy challenge, Renaissance (2016-2018) to clean up the existing code and Phoenix (2018-2020) to prepare the software architecture for future extension. This resulted in initial tooling. The approach was effectively turned into practice at Thermo Fisher Scientific. Klomp is very pleased with the results. "We've managed to significantly reduce the time it takes us to build a new version of the software. We're now releasing four times as often compared to three or four years ago."

The results haven't gone unnoticed. Also assisted by ESI, Philips has adopted the tools developed within Renaissance to take on a similar legacy challenge in its medical scanners – with similar success. "The quality of our code has increased significantly," says Hans van Wezep, a software architect within Philips' Image Guided Therapy business. "By giving us more insight, the tooling really opens new doors in innovation for us. We've integrated Renaissance in our own collaboration project with ESI, Vivace, to build on these promising results."

Renaissance and Phoenix

To tackle its legacy software challenge, Thermo Fisher Scientific in Eindhoven embarked on two consecutive research projects together with ESI. In Renaissance (2016-2018), they cleaned up the existing code and in Phoenix (2018-2020), they prepared the software architecture for future extension.

Software dishwasher

One way to clean up a mess is to throw everything away and start over. Thermo Fisher Scientific's Klomp explains why this is not an option: "We've added a huge number of features over the years. When you start from scratch, it will take an enormous amount of time and money to get to the same level of functionality – actually, more than it will cost to clean up your existing codebase. When your dishes are dirty, you could throw them away and just get some new ones. But if you do that every time, in the end, buying a dishwasher is much less expensive. So that's what we did: we got ourselves a dishwasher for our software."

In essence, the approach developed by Thermo Fisher Scientific and ESI in the Renaissance project cleans up the code by uncovering structural bottlenecks and offering pointers to resolve them. It does this automatically. "Instead of using a dishwasher, we could also clean the code manually, but that's very labor-intensive, so again very expensive, and very tiresome for the engineers, so highly error-prone," clarifies Klomp. "Automatically cleaning the code not only reduces the effort but also lowers the risks as it ensures that the software remains working."

Cleanup consists of two phases: analysis and transformation, in multiple iterations. First, the ESI team analyzed the software architecture by creating pictures of it and discussing them to pinpoint suboptimalities. During this process, they were supported by a lead engineer from Thermo Fisher Scientific, who provided in-depth knowledge about the code. Klomp: "He helped them understand the architecture pictures. What does it all mean? Why is it like that? What should it look like? For this domain knowledge, we sometimes had to go shopping within our organization, ask around, do some digging."

Following the findings, the code was restructured to get rid of the suboptimalities. "This transformation wasn't actually fully automated from the beginning," Dams from ESI notes. "We started with dividing the cleanup into about 70 parts. The first parts we optimized mostly manually. This provided insight into what needed to be done and brought to light recurring patterns, enabling us to write some tool support to do those things automatically. As we proceeded, piece by piece, we automated more and more patterns and by the time we got to the last parts, most of the work was automated. Not everything, because there will always be some work that requires manual intervention."

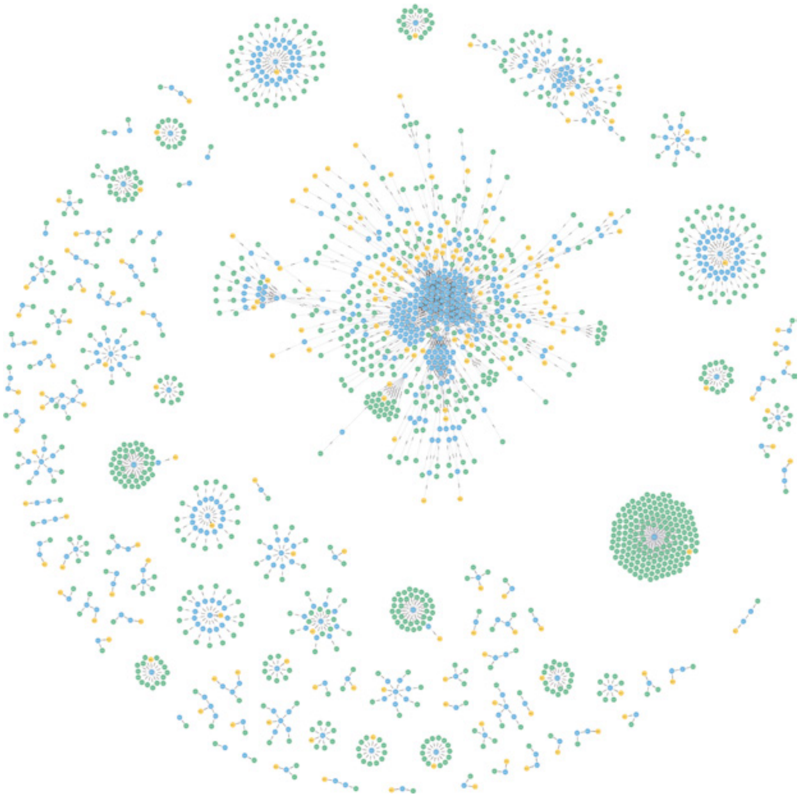
All the while, the cleanup crew worked in parallel to the Thermo Fisher Scientific team developing new features. Or, as Dams puts it: "We were repairing the plane while it was flying." For Klomp, this is one of the project's big achievements. "At times, it was challenging to merge the cleanup crew's work with new feature delivery, but in the end, we managed to break nothing. Had we disturbed the ongoing software engineering process, that would have generated a lot of complaints and resistance and the project wouldn't have been anywhere as successful."

The tools are now in broad, day-to-day use at Thermo Fisher Scientific. Klomp: "Our architects and senior designers use them to monitor the code structure to see whether it's deteriorating again or, even better, improving."

Code browser on steroids

Early in 2018, the positive sounds from Thermo Fisher Scientific reached Philips. "We have an older product line of interventional X-ray scanners, for which we don't do much new development anymore but mostly maintenance. A lack of insight into its dependencies made it quite challenging for us to coach new engineers to work with it or do a redesign for it," recollects Hans van Wezep. "Through ESI, we learned about the Renaissance project at Thermo Fisher Scientific and we quickly realized that what happened there could be beneficial for us as well. We adopted the analysis tooling and, with only some minor changes, had it up and running very quickly."

Together with ESI, Philips successfully applied the Renaissance approach not only to the scanner software but also to the build environment in which that software is being developed. Van Wezep: "We use a standard Microsoft setup and we saw people not using it to its full potential. Dropping all the code they'd written into the same folder and compiling it from there, for example. However, when you create separate packages and layers in your development environment like you do for the software that you're building with it, as well as clean up the dependencies, it's much easier to see how everything's connected and it makes that environment much more efficient."



Credit: ESI

As the software grows, it's getting increasingly difficult to maintain an overview of its inner workings and keep it well-structured.

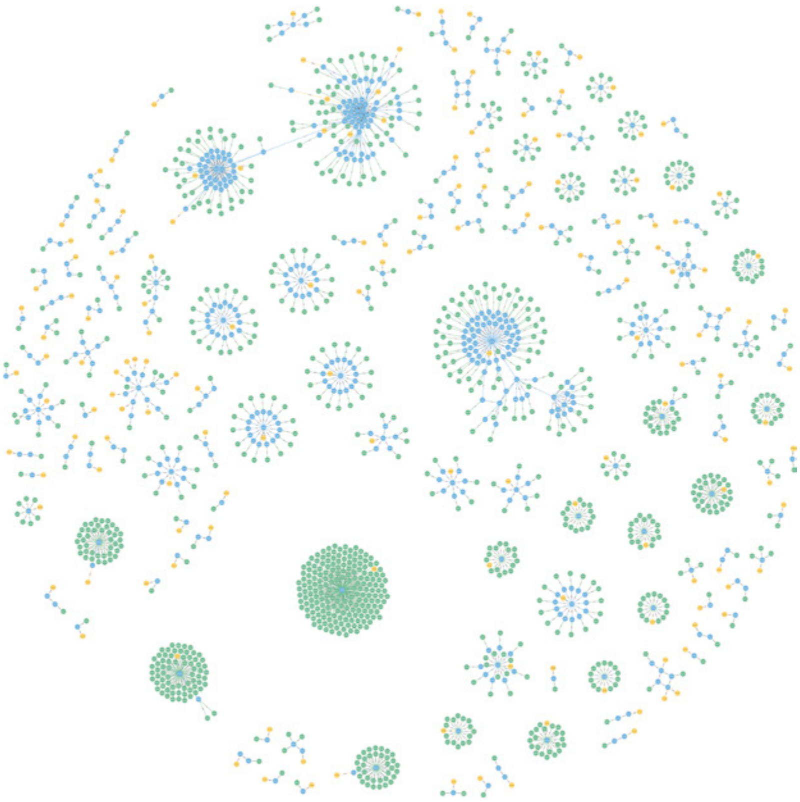
"One of the great features of the tooling is the extensive way in which it presents its findings," states Van Wezep. "It's very easy to see whether you're dealing with a recurring pattern, which you then can have resolved automatically, or an exception, which you have to fix by hand. Most issues, about 80 percent, can be handled by the tooling; the rest requires manual effort. We had someone do everything by hand and it was very nice to see how effortless the tooling worked in comparison."

"I had a similar experience at Thermo Fisher Scientific," says Dams. "After having used the tooling, one of the developers came up to me and said, 'Your tooling does in 3 seconds what I've been working on for three weeks and it even does it better!' Because the work is so tedious, it's very easy for a human to make a mistake."

The visualizations play an important role here, Dams thinks. "There are several systems available for browsing your code. In most of them, you click on links to go to another level – up or down. It's very easy to lose sight of the bigger picture. Our tooling gives a bird's eye view. You see all the important items and their dependencies in one glance, making it very easy to spot the structural suboptimalities. I heard someone call it a code browser on steroids."

Maintenance hell

With the important transformations done, the focus at Thermo Fisher Scientific has changed from cleaning up the legacy code to making the software future-proof. "That's what we're looking at in the Phoenix project," explains Klomp. "With future business requirements in mind, what architecture do we need and how do we get from the current architecture to the desired future architecture? And, equally important, how do we keep people from violating that architecture? We want to be able to detect those violations at an early stage and fix them quickly."



Credit: ESI

The approach developed by Thermo Fisher Scientific and ESI analyzes the software and restructures it to get rid of the suboptimalities.

Inspiration is now also flowing in the opposite direction, from Philips to Thermo Fisher Scientific. “Instead of their using the results of our project, we’re using tooling from them, to better manage the interfaces between the components in our software,” specifies Klomp. “We see a huge demand for further automation based on technologies like artificial intelligence. We’re building new subsystems for that and we need to fit them and their interfaces neatly into the new architecture. The interface tooling from Philips can help us do that.”

Halfway through, the Phoenix project is already bearing fruit. “We’ve found a software component intended for one product line to be suitable for reuse in another product line,” gives Klomp as an example. “We’re now in the process of isolating it and giving it proper interfaces. Reusing instead of redoing – that’s a clear benefit. We expect to find more of these components. Because of the progress we’re making, our work is getting more traction and more visibility within the company, not only in Eindhoven but at our other development sites as well.”

“The collaboration with ESI enables us to really focus on the features that add value in the microscopy domain while keeping our software and software engineering process state-of-the-art,” summarizes Klomp the key takeaway. Van Wezep agrees: “The same for Philips. Getting trapped in a software maintenance hell would hold us back from being more innovative. With the help of ESI, we can now separate our main concerns from the grunt work that should be automated away.”

Dams, quoting a previous project manager of his: “At ESI, our job is to understand our customers’ problems, develop cool solutions at the forefront of technology and get them integrated into the day-to-day workflow. That’s exactly what we’re doing here. We’ve developed a novel approach and a matching set of tools for practical use, not only at Thermo Fisher Scientific and Philips but potentially at other companies as well. The long-term maintenance that this requires is outside our scope of work, though, so we’re now looking into ways to transfer the tooling to third parties for industrialization.”


TEM Bright field
SA 3400 x
Exposure
Stigmator
Intensity
Dark field
Diffraction
Multifunction Y
Magnification
Focus
R1
R2
R3

System and software testing

System complexity is increasing, also because more and more software is being added over time. At the same time, systems are increasingly being integrated into their environment, for example at customers, to create so-called systems-of-systems. Moreover, more and more disciplines are contributing to the development. Lastly, the systems remain in the field for a long time, sometimes over thirty years, during which they evolve and many new configurations are created.

As a result, the testing efforts have exploded. The expertise area of system and software testing is looking at new model-based and model-assisted techniques to keep up and support all the developments. The goal is to improve the effectiveness and efficiency of testing while reducing the effort required.

Challenges

At the system level and beyond, the traditional ways of testing aren't sufficient anymore and new approaches are called for. With more and more disciplines being involved in the development, each using their own language, communication becomes more and more challenging as well. The growing number of different system configurations, sometimes in the thousands, makes it increasingly difficult to run updates and test everything afterward, so smart solutions are needed to deal with this variability. Furthermore, in the decades that the systems are in the field, large test repositories are created that have to be maintained.

Tackling the challenges

In a so-called model-assisted approach, different kinds of models at different abstraction levels are created and used as the basis for testing. Automation is introduced where this is useful, for example with test generation. In brownfield development, there are usually already a lot of tests, so test selection becomes important to determine which tests to run. Test coverage techniques help keep the overview during the entire process. Models are equally instrumental in reducing the time spent on maintenance. A divide-and-conquer approach, in which the system is broken down into parts and each part is qualified separately to get some guarantees for the whole, helps to keep the testing manageable – but also creates new challenges, like the integration of parts into the whole system.

Where are we today?

At ESI, we've made considerable progress in developing methodologies and tools for model-based testing. We're now taking the first steps in brownfield situations, where companies already have tests that they don't want to throw away. They usually don't want a completely new approach but something in between. We're going in that direction, connecting to their way of working and creating real value with models.

In the Vivaldy project, for example, we're looking into test selection and test generation when specifications change. We're seeing some promising results there and are now rolling out the methodologies at Philips. The Matala project is a similar collaboration with ASML, focusing on model-based test generation. Together with Radboud University Nijmegen and the University of Twente, we recently started a research project with PhD students at ASML and Canon Production Printing, Testing in Times of Continuous Change (TiCToC). There, the focus is on testing evolving systems. In all of these projects, testing systems with high variability is a recurring theme.

ESI 20 YEARS

ESI

Sid and Ally provide AI assistance in bug resolution

Together with ESI, Philips is eyeing the use of artificial intelligence to improve its software development efficiency. The advanced bug search interface they developed in the Accelerando project is already spreading like wildfire within Philips, while their AI-enabled bug triage tool is saving the company's software experts hours of administrative work.

"Azurion is our advanced platform for image-guided minimally-invasive therapy, with over 4 million treatments already performed using the system. It's the result of many years of development and for many years to come, it will be enabling the innovation of procedures in hospitals across the world. And throughout the platform's lifetime, our developers will be keeping systems in the field continuously up-to-date," says Patrick Bronneberg, principal cloud architect at Philips Image Guided Therapy (IGT). "The challenge is to strike a balance between this ongoing system maintenance and our ability to create innovations, while at the same time ensuring that the knowledge gained during product support is fed through into our development efforts."

To address these challenges, Philips IGT partnered with ESI in the Accelerando project. "Thanks to methods such as Agile and Lean, our engineering and manufacturing processes have continually improved over the years. Together with ESI in the Accelerando project, we're now looking at how we can use AI to improve our development processes," explains Bronneberg, who specializes in developing cloud and AI-based services. According to Ronald Begeer, Accelerando project lead at ESI, the challenge is industry-wide. "All across the Dutch high-tech ecosystem, we see a growing need for the more efficient deployment of people and resources. In Accelerando, which started in 2020, we're working on optimizing R&D processes using natural language processing and other forms of machine learning. For the AI expertise, we're collaborating closely with TNO's Data Science department."

Credit: Philips

Philips' Azurion platform for image-guided therapy. The challenge is to strike a balance between the ongoing system maintenance and the ability to create innovations, while at the same time ensuring that the knowledge gained during product support is fed through into the development efforts.


Credit: Philips

Automation

The Accelerando project focuses specifically on issue tracking – the process of reporting, assessing and solving software defects. "When a software bug is found, a description of the problem is entered into the bug database," details Bronneberg. "Every few days, an accumulation of recent issues is sent to the project's change control board – a panel of software experts who determine the severity of each problem, try to identify the component that caused it and assign the issue to the development team responsible for the component, who then attempt to fix it."

Bronneberg points out that there are several places where this process can hit a snag. "The problem may already have been reported, for example, resulting in a duplicate entry in the bug database, or it may be difficult for the change control board to pinpoint the root cause, potentially resulting in the issue being handed to the wrong development team. And even if the defect does land in the right team, an inadequate bug description may make solving it harder than necessary."

Today, software teams in the high-tech industry are spending about 60 percent of their time not on bug fixing but on the administration around it. "But the chances are a lot of information about similar bugs is already available in historical bug reports," Bronneberg reckons. "Through AI-enabled automation, we want to connect our developers to that information and make it come to life, giving them a better insight and allowing them to focus their attention where they can add value."

Sid

The Accelerando project is addressing three stages in the issue tracking process: duplicate detection, bug triage and bug localization. "Resolving duplicates is a time-consuming process," finds Dennis Dams, a senior research fellow at ESI. "Taking off-the-shelf search technology that uses NLP, natural language processing, we've developed a tool to help speed up duplicate bug detection. In a web-based application, developers can enter a short description of the issue and they get back a list of possible duplicates as well as a couple of suggestions for related keywords to improve their search."

Credit: Philips

The Similar Issue Detector (Sid), as the application has been dubbed, is trained on a collection of bug reports that Philips has built up over the years. Employing an established NLP technique called Term Frequency-Inverse Document Frequency (TF-IDF), words and phrases are grouped together that often appear together in these bug descriptions. The user input is then correlated in real-time to the TF-IDF output, after which the top results are displayed. "Every once in a while, usually at the start of a new project, we retrain Sid on the latest version of the report database," notes Dams.

Sid was very well received by the software developers at Philips – much better even than Bronneberg had expected. "We started with a rudimentary search interface, the idea being to streamline it later on but generate enough clout in the meantime. But it spread like wildfire. People who witnessed colleagues working with Sid wanted to get in on it as well. It not only helps them detect duplicates earlier, it also stimulates them to make a general problem description more specific, and it gives them an all-round better search interface to our bug database. More and more of our developers are using it on a daily basis, so we're even more intent on integrating it into our existing toolchain."

Ally

Once an issue has passed duplicate detection, it's assigned to a software team for resolution in a process called bug triage. "Triaging is a very difficult task, with first-time-right traditionally being a near impossibility," explains Dams. "To support the process, we developed a second tool using similar NLP technology and integrated it into the change control board workflow. New issues are delivered to the board in an Excel spreadsheet, which we've enhanced with functionality to automatically identify the software teams that are the best candidates for assigning a defect to. With the push of a button, a top-three is generated for each issue."

Ally the Algorithm, as the Accelerando partners have dubbed this tool, is again trained (and occasionally retrained) on Philips' collection of bug reports. In a supervised-learning approach, the algorithm is fed with problem descriptions from the past, each labeled with the team that ultimately fixed the issue. Using the TF-IDF technique, Ally builds up a model that connects terms appearing in a report to likely candidates for bug resolution. A new problem description is then correlated in real-time to this model, yielding the three best matches together with scores that indicate how confident the algorithm is about each match. It's then up to the change control board to pick a team and assign the issue.

When comparing Ally's performance to that of the human panel, the algorithm achieves virtually the same accuracy but in a fraction of the time. "Supported by its AI capabilities, Ally produces a result in a matter of seconds and, as a result, one of our change control board members told us they're now able to shorten their meetings by 20 minutes. With seven or eight experts convening three times a week, that's a massive time saving – valuable time they can spend on expert tasks rather than administrative work," says Bronneberg. "It also gives us a much better insight into common triaging pitfalls. When one of the change control boards was presented with a matrix of historical data showing where they initially assigned an issue and where it ultimately got fixed, they realized how powerful Ally was in moving towards first-time-right decision-making. Insights like this really make the data come to life and help us increase our efficiency."

Chatbot

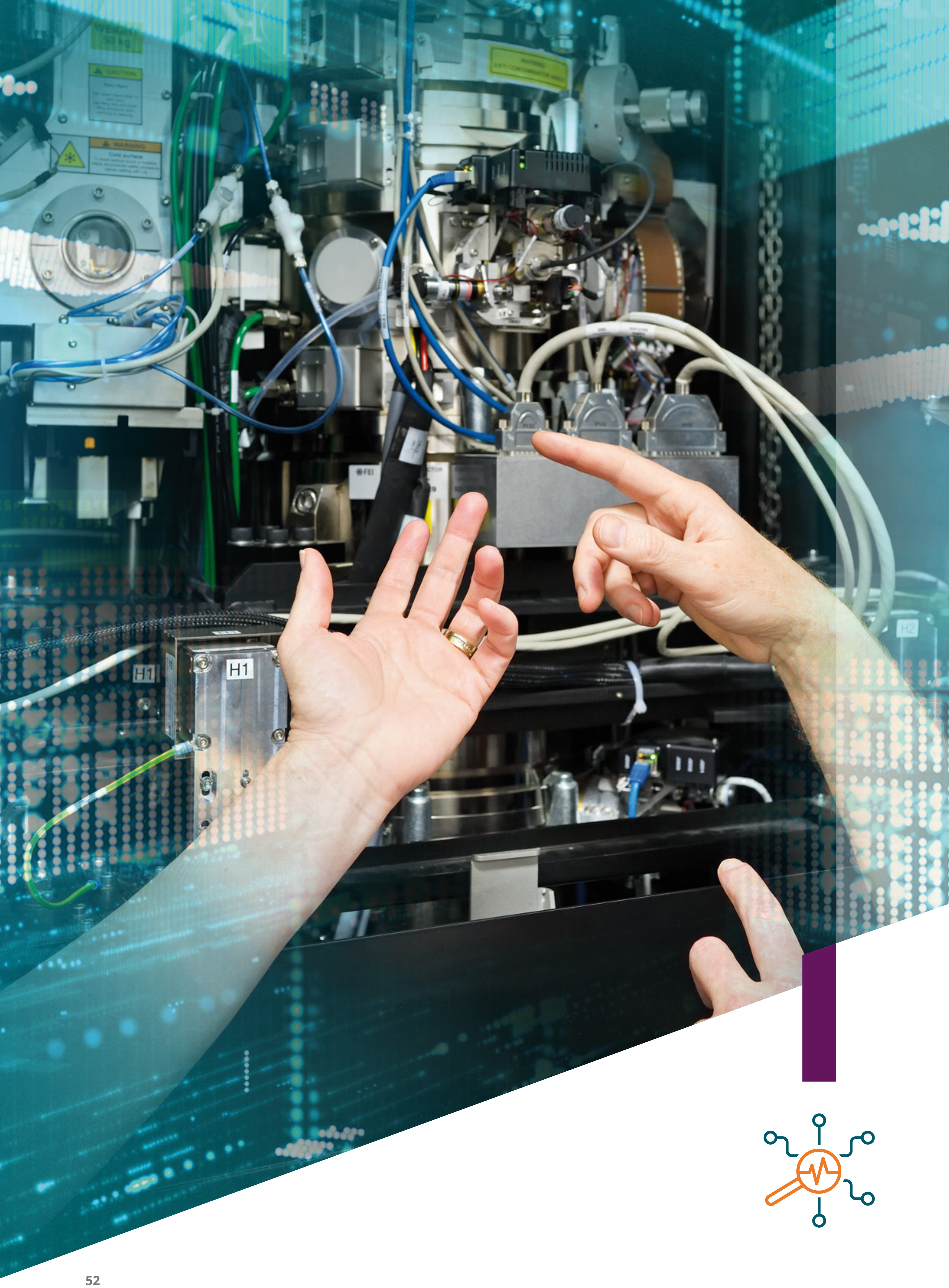
A software team that gets an issue assigned to it for resolution needs to localize the bug first, and research is now ongoing to see how AI can assist in performing such an analysis efficiently. "We've run some experiments with different techniques to see how AI can help. We've found that it's not enough to only look at the text of a bug report," observes Dams. "We also need to examine the log files that come with the report."

Another avenue of research is improving the accuracy of Sid and Ally by improving the quality of reporting. "Seeing their performance plateau despite the application of more advanced AI techniques like deep learning is a sign that perhaps the input data needs some work," Dams reckons. "That's why we're now also looking at raising the quality of the bug reports – the more precise the problem descriptions, the easier it becomes to detect duplicates and do bug triage."

Bronneberg's vision is to have a chatbot-like tool. "Improving the quality of reporting has always been a challenge, with the risk of getting bogged down in detail. No developer wants to have to fill in fifty fields. Instead, we'd like to gather the required information in a structured yet user-friendly way. Think of a chatbot, guiding you through the process and only asking you for information that can't be deduced automatically or extracted from your earlier responses."

Benefit

Bronneberg heralds the power of data science as his main takeaway. "By leveraging that power, you'll gain so much more insight into your business processes." ESI project lead Begeer agrees. "Applying standard data science techniques provides a wealth of process information. All of high-tech can benefit from it."


H1
H2

Intelligent diagnostics

When there's a performance issue with a system in the field, certainly in case of a complete breakdown, the problem needs to be fixed as soon as possible. A service engineer is sent out to assess the situation and come up with a solution, preferably on the spot. The expertise area of intelligent diagnostics looks to speed up this process by automatically providing the right information at the right moment, thus reducing the knowledge required for troubleshooting. The goal is to achieve first-time-right diagnostics and, ultimately, zero unscheduled downtime.

Challenges

Today's high-tech systems are still somewhat removed from the ideal of first time right. There are quite some problems that only pop up after deployment as the equipment ends up in contexts unforeseen during development.

Also, the systems have grown so complex that service engineers almost need to know as much as the original developers. To convey this knowledge, a lot of effort is put into creating service manuals that can be tens of thousands of pages long, rendering them practically unmanageable. What's more, they don't even contain all there is to know as a wealth of information stays in the heads of the developers. This makes efficient troubleshooting very hard, and for more generalist service engineers virtually impossible.

Tackling the challenges

The key to efficient troubleshooting in the field is to provide service engineers with the necessary system information at the exact time they require it. As model-based systems engineering is gaining traction, a lot of this information is already available in design models. By reusing these models for diagnostics and making the system knowledge available in a user-friendly tool, with the help of probabilistic reasoning techniques, service engineers can be efficiently guided through the problem-solving process.

In a so-called hybrid AI approach, the system knowledge from the design models can be supplemented with system data collected in the field. Such a fusion of knowledge and data will make the guidance even more efficient.

Where are we today?

At ESI, we're running several projects to integrate diagnostics into the design process. Together with ASML, we've shown how design schematics can be translated into a diagnostics model and used to assess the level of diagnosability as well as to generate input for service tooling.

In collaboration with Canon Production Printing, we're developing hybrid AI technology to support service engineers. Based on system design knowledge, we defined diagnostics models that can be used to guide troubleshooting in the field and advice on service tests. This project, called Carefree, is explained in more detail on pages 50-53.

With Philips, we're working on software diagnostics in the verification and validation phase of development. In the Accelerando project, we're looking at the use of AI-based tools to resolve bugs. This project was explained in more detail on pages 44-47.

ESI 20 YEARS

ESI

Streamlining troubleshooting in the field

Canon Production Printing has teamed up with ESI to identify potentially failing parts and predict potential issues in production printers. In the Carefree project, they're developing hybrid AI technology to support service engineers.

"At Canon Production Printing, predictive maintenance has been a topic of interest for quite some time," says Peter Kruizinga, a lead technologist at the Venlo-based company. "The aim is to get an accurate picture of when a printer or one of its components is going to break down by analyzing the machine's data, and then dispatch a support engineer to preempt the issue. We've already successfully implemented this for the transport belts in our systems: a significant rise in power usage of a belt drive is a direct sign of impending failure, so when we see that happening, we can send someone over to replace the motor before it breaks down."

Not all cases are so clear-cut, though. "A printer has thousands of parts," notes Kruizinga. "For a lot of them, unfortunately, there are no such direct indicators, so in order to ascertain the source of a problem, we need a lot of indirect information, which often is unavailable. For a specific part, the failure mode can also vary from one situation to another. We realized that doing predictive maintenance on a structural basis requires us to solve a large number of puzzles, and to get all the pieces, we have to make strides in problem diagnosis."

With predictive maintenance being the ultimate goal, the initial focus is on corrective maintenance. Kruizinga: "Although our printers come with an extensive manual and have some diagnostic tests built-in, we see our service engineers mostly relying on their experience when they're called to fix a problem in the field. We want to guide them in finding the root cause faster. The first step is to improve our diagnostic capabilities after the fact, so after an issue has occurred. The next step is to also be able to see problems coming in advance."

For the paper input module, the Carefree project has constructed a so-called Bayesian network to find the most likely root cause of a problem.

In the Carefree project, a two-year effort that started in January 2020, Canon Production Printing (CPP) has teamed up with ESI to identify potentially failing parts and predict potential issues. Combining models derived from the development phase with current machine data, the partners are building a self-learning diagnostic system that can pinpoint the cause of a future problem or advise where and how more information needs to be gathered to do so. As a result, timely action can be taken to prevent long and expensive downtimes.

Abstracting Bayes

The Carefree approach is based on so-called Bayesian networks. "Given a problem, such a network can lead you to the most likely root cause," explains Jos Hegge, senior project leader at ESI. "It allows you to enter 'evidence,' like error messages, and it then calculates the effects of this evidence, updating the probabilities and returning new conclusions on the most likely causes. It can also streamline your quest by suggesting which tests to run to get to a result more quickly."

The main drawback of Bayesian networks is that they tend to get really big really fast. "For a professional printer, consisting of thousands of components, creating such a network by hand is a rather hopeless enterprise, even more so considering you have to change the network each time you tweak your system design," observes Hegge. "Which is why we've added an abstraction layer in the form of a high-level language. We can use this language to describe a diagnostic model and then automatically convert it into a Bayesian network."

At the moment, the higher-level models are still manually created. "The ultimate goal is to be able to generate the Bayesian network from a printer's design documentation," Hegge points out. "As that places very high demands on the source documents, we've decided to settle, for now, for this intermediate abstraction layer. Later, at a moment of its choosing, CPP can pick up where the project left off."

The same paper input module has been used extensively in a wide range of printers in the field, giving a detailed insight into the frequency with which errors occur in practice.

For CPP's Kruizinga, the higher-level description language is actually one of Carefree's notable achievements. "Creating a Bayesian network requires really specific craftsmanship. Thanks to the abstraction layer, we can now put the specification work into the hands of software designers who don't have that very specialist background but who can do an excellent job just using their knowledge of the application. Also, we're not faced with the daunting task of having to manually update the Bayesian network each time we change the system design."

Honing in

The higher-level diagnostic models created in the Carefree project consist of abstract representations of printer components, their capabilities and observable erroneous behavior. "Each printer has a paper input module, with a paper tray that moves up a little every time a sheet is fed into the machine," takes Kruizinga as an example. "For this module, we've constructed a model with components such as the lift motor and the tray sensors. The lift motor has as its capabilities that it can move up and down, while one of the sensors is capable of detecting the up position. Both the upward movement and the up position detection can trigger the observation that the lift table has arrived at the up sensor too late."

The model includes for every component the chances of it being in a specific state, for every capability the chance of it being enabled and for every error the chance of it occurring. These probabilities are based on factory specifications or they can be derived from historical machine data. "The same paper input module has been used extensively in a wide range of printers in the field," Kruizinga goes on to illustrate. "This gives us a detailed insight into the frequency with which components and capabilities break down and errors occur in practice, allowing us to more accurately determine the a priori probabilities."

By adjusting the probabilities based on actual evidence gathered in the field, it's possible to hone in on the root cause of a failure. Kruizinga: "When the printer displays an error message indicating the lift table has arrived at the up sensor too late, the probability of that error occurring can be set to 100 percent. This change then propagates across the model, affecting the other probabilities. It raises the chance of the lift motor being faulty, for example. Additional evidence can be obtained by running diagnostic tests on the printer. If one such test shows that the up position can be detected successfully, the probability of that capability can be set to 100 percent. Eliminating the sensor as the source of the problem increases the likelihood of a faulty motor to almost 100 percent, giving us the probable root cause."

With the help of artificial intelligence, the project partners believe the diagnostic performance can be given a further boost. "We're looking to make the probabilities even more accurate by applying AI techniques to actual sensor data from the printer. We're also considering using AI to get useful information from user profiles – it might very well be that customized print jobs result in different wear and tear than mass printing," argues Hegge. "This hybrid AI approach, refining human knowledge with computer-generated insights, has been well received within Canon."

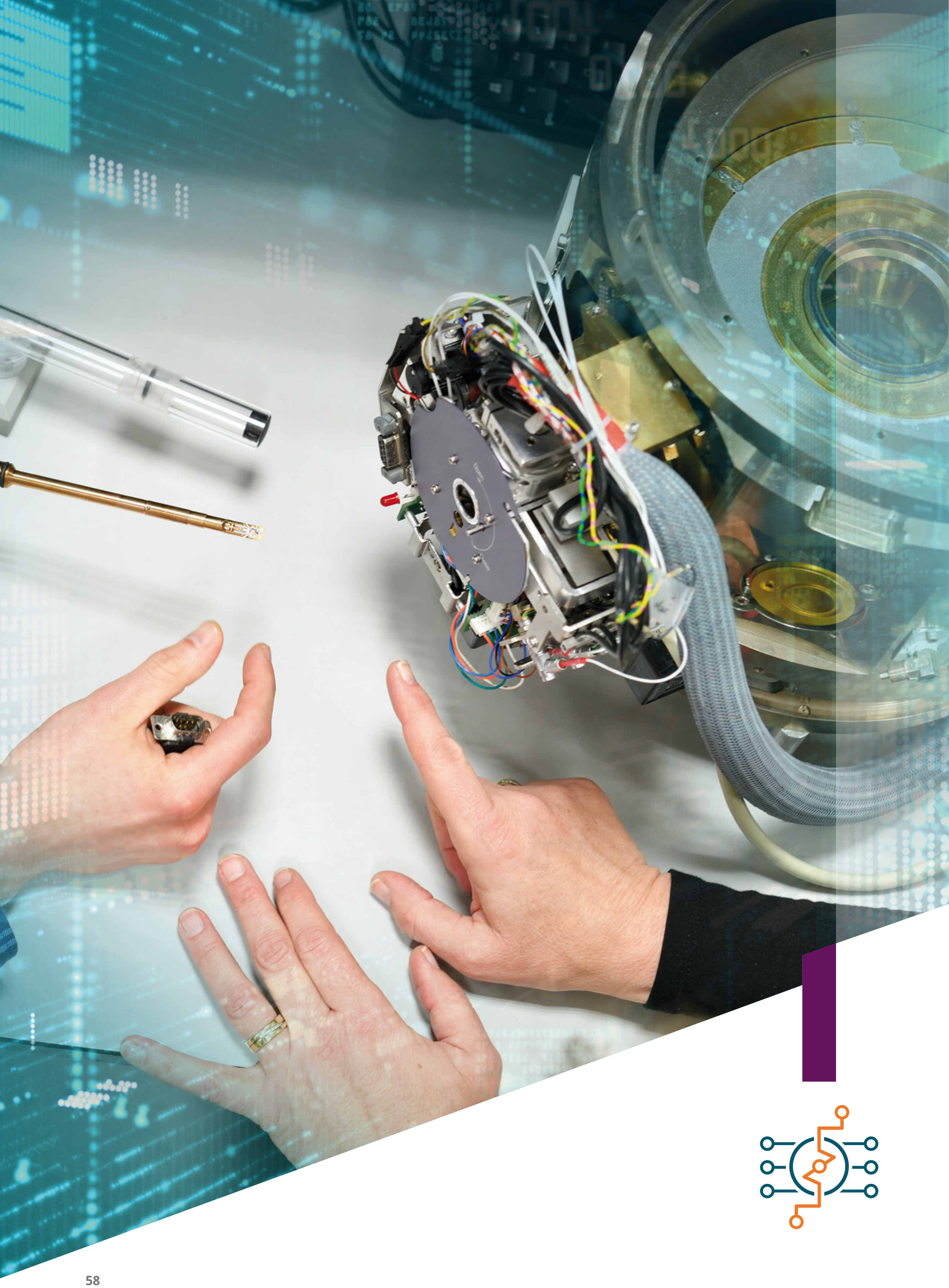
Design for diagnostics

Although the preliminary project results are already very promising, Hegge sees much more potential to unlock. "There are still many relationships to uncover within the printer domain – all kinds of thermal and mechanical interactions spring to mind. We haven't included those in our models yet, but they could be highly instrumental in improving the decision engine."

The technology could also be helpful during system development. Hegge: "In an earlier project at ASML, we've applied it to do design for diagnostics. If it's not possible to go from an error message to the trigger of the error in one step, that's a good reason to introduce additional sensorics." Kruizinga: "Using the technology in system design is an interesting avenue of research for us as well. What additional tests can we build into the printer to improve the diagnostics?"

As the main deliverable, Carefree will lay down the technology foundation for CPP to incorporate into a maintenance tool for easy troubleshooting in the field. "Whenever there's an issue at one of our customers, we dispatch a service engineer, who, after feeding the tool with the error messages and other 'evidence' found at the scene and running some additional tests as suggested by the tool, swiftly ascertains the cause of the problem by combining the results from the tool with his own expertise," envisions Kruizinga. "This combination of computer smarts and human skills is key. The tool isn't meant to replace our service engineers but to support them in doing their work even better."

"At ESI, we've built up a lot of experience with this kind of diagnostic challenges – in the collaboration with ASML but also with Thales, for example. In the Carefree project, we're bringing this knowledge to Canon," concludes Hegge. "Our goal is to generate a hybrid AI approach that can be used to manage the maintenance of professional printers, but that's also beneficial to the high-tech industry as a whole."



Software behavior

Traditionally, high-tech systems are described in terms of their structure – what are the constituent parts and how are they interconnected? Documenting their behavior tends to receive much less attention. The expertise area of software behavior aims to resolve this by capturing in different ways how the building blocks affect their surroundings. This includes both the static and dynamic behavior of a single interface, multiple interfaces, an entire component, multiple components, up to the complete architecture.

The aim is to create a single source of information with complete descriptions of the behavior. Writing everything down only once helps to achieve correctness by design. From this single source, different artefacts can be (automatically) generated, such as code and essential tests, which also contributes to achieving correctness by design. Having complete descriptions makes it possible to fully analyze the behavior and thereby detect issues early in the development process. Also including information about behavioral changes across different releases reduces the risk of software upgrades.

Challenges

Because the components can be highly complex, it's far from trivial to exhaustively describe their behavior. It's often quite difficult to determine what they're supposed to do exactly. This is already very challenging for existing systems but all the more so with behavior that changes over time due to software upgrades.

In constructing a single source of information for software behavior, it's important to write everything down in one place. The problem is that during development, a lot of descriptions are produced that aren't in sync with each other. The challenge is to establish a consistent basis from which as much as possible can be generated.

Tackling the challenges

These challenges can be tackled by capturing and centrally storing the essence of the behavior in formal and complete model-based descriptions, e.g. using domain-specific languages. The necessary input is gathered from available channels such as documentation, logs and tests. The ultimate goal is to use such a single source of information as a basis for (further) development efforts in a continuous integration and deployment approach.

Where are we today?

It takes some time for companies to embrace this approach. It often starts as a parallel track to show the benefits, after which the practices are gradually adopted and the switch to a single source of information is initiated. At ESI, we're seeing the first small-scale successes here.

Our research focuses on precisely and completely modeling components and their interfaces, analyzing behavior (e.g. by checking properties), using models to generate artefacts (e.g. documentation, monitors, tests, middleware) and tracking the evolution of behavior (e.g. by comparing software artefacts and selecting tests based on changes).

One of the prominent results is the Comma approach for modeling and analyzing software interfaces. It has been gradually extended to deal with components with provided and required interfaces, and recently to compound components. This approach is explained in more detail on pages 56-59.

Research on software evolution is carried out in projects with ASML and Philips. These projects address learning of behavioral models from logs, extracting models from test scenarios, comparing models and logs of evolving software components at different levels of abstraction and selecting/generating tests that cover changes.

Comma interfaces open the door to reliable high-tech systems

Once a research project, initiated by ESI and Philips, the Comma framework is developing into a mature product for creating and managing software interfaces. Thales is also looking to use it to streamline its software engineering, as are Thermo Fisher Scientific and Kulicke & Soffa. “Comma is the place where you express everything you want and from there, you generate everything you need, like documentation, monitoring, simulation, visualization and, as of recently, test cases.”

“Our medical devices are growing bigger and bigger,” observes Daan van der Munnik, technical department manager software at Philips Healthcare in Best. “We have to chop them up in smaller subsystems to keep their development manageable, but also for validation purposes. Up to a year ago, we validated a complete device in one go – a huge effort. By chopping it up into smaller subsystems, we can focus our validation efforts on the parts of the system we actually touch for a particular feature. We do need to show that when we put everything together, it still does what it’s supposed to do. Both the disassembling and reassembling call for good interface management.”

Subsystems are also increasingly being farmed out to subcontractors. “We’re really moving to a system-of-systems development, where we make some parts ourselves and some parts come from outside,” notes Van der Munnik. “For instance, in one of our image-guided therapy systems, we have three types of patient tables. One is developed by us, two are made by other companies. From a user perspective, however, they all have to appear to be an integral part of the system – the user experience, for instance when moving or tilting, has to be exactly the same. This means that, for our subcontractors, the interfaces need to be clearly defined, both on a low technical level and a high subsystem level.”

Last but certainly not least, good interface management is key for system evolvability. Van der Munnik: “Our medical devices have very long lifetimes. We need to ensure that, over their lifetime, they’re expandable and suitable for form/fit/function replacements.”

Evolving interfaces

Fellow high-tech company Thales faces similar challenges. “Traditionally, we developed, built and qualified our combat management and radar systems, delivered them to the customer and mostly touched them to replace obsolete components – to avoid unnecessary risks, functional changes were rather limited and implemented at long intervals,” explains Pepijn Noltes, a software architect at the Hengelo-based company. The last ten years, however, the operational scene is changing more rapidly at extended operational lifecycles, with customers increasingly demanding new features. Thales is adapting to this need by looking for ways to implement software updates more frequently, including incremental enhancements.

But that’s easier said than done. “For complex software-centric systems like ours, it’s very expensive to change something, integrate and test it – especially so in the military domain that we’re in, where you may have to do live firing trials to really validate the system,” says Noltes. “Also, even the tiniest update may cause an avalanche of changes. It then boils down to the question: how well can you revise part of your system without touching the rest?”

Noltes has learned that to be able to continuously update a complex system, you need to keep the changes local and to do that, you need to focus on the interfacing. “We tend to touch the interfaces as little as possible because they’re expensive to change. Bigger problems that you can’t work around will eventually get fixed, but small issues will remain, as a result of which the code quality will slowly deteriorate. We’re now looking at evolving interfaces to facilitate the need for change.”

Single source of truth

Enter Comma (Component Modeling and Analysis), an ecosystem supporting model-based component engineering. “It started as a research project between ESI and Philips,” recalls Jozef Hooman, a senior scientist at ESI. “We began using domain-specific languages for all kinds of purposes, generating code, analysis tools

Credit: Philips

Thanks to Comma, it's easier for Philips to enhance its medical systems.

and much more. While doing this, we noticed that a lot of issues Philips had with its software were due to interface problems and, gradually, the insight came to us that these DSLs were especially useful for describing the interfaces. So, in small steps, we moved from general-purpose languages to a domain-specific language, Comma, which we reused for many different interfaces."

Although a research project, the development of Comma wasn't driven by research considerations, Hooman points out. "We really looked at what the engineers at Philips needed and adapted the language accordingly. We started with a state machine describing the interface protocol, i.e. the interaction between client and server. Based on user feedback, we modified it to make it more user friendly and include things like timing and data constraints. The patient table, for instance, is very sensitive to both timing and data – when the controlling joystick stops, the table should stop too within a certain amount of time and without moving too much."

Step by step, Comma developed into what it is now: "the single source of truth," as Hooman calls it. "This DSL is the place where you express everything you want and from there, you generate everything you need, like documentation, monitoring, simulation, visualization and, as of recently, test cases. Monitoring, especially, is very important. You can use that to see if your implementation satisfies the specification by running the system, collecting traces and check whether the execution conforms to the interface. If an interface changes, you can re-generate everything, and if your developers, or your third-party suppliers for that matter, introduce a software update, you can check that for conformity – all with the push of a button, continuously, as an integral part of your test process."

Forming an ecosystem

At Philips, Comma is now firmly embedded in the company's software engineering practice. Van der Munnik: "We use the DSL to write the interface specs and generate documentation and code. As part of our continuous integration pipeline, we check interface conformance against the Comma specs when executing our automated test scenarios. We've created a maturity matrix, which sets off our interfaces against these development stages, and we're now raising the bar for all of them. Thanks to the unambiguous definition of interfaces and the subsequent automatic validation, Comma brings us a huge amount of business value as we find interface issues early, well before integration."

In 2018, Thales started the Dynamics project to research dynamic system updates in collaboration with ESI. "We're looking into evolvable interfaces and so-called adapters to keep the old and the new working together," clarifies Noltes. "So when you introduce a client with an updated interface, you also generate an adapter that connects it to your existing server and provably ensures that nothing breaks down. ESI did a small technology survey on interface specifications and Comma came out as the solution that best fits our needs. Although Dynamics is still ongoing research, Comma is already useable out of the box and we're busy including it in our component development framework. By doing more at design time, we hope to eliminate much of the risk in projects."

Slowly but surely, Comma is conquering the Dutch high-tech. "We're also working with Thermo Fisher Scientific in Eindhoven, for instance," illustrates ESI's Hooman. "For some critical interfaces, a model has been made and a monitor has been built into the nightly smoke tests, which automatically checks the log files. In the morning, they can see what properties have failed. And Kulicke & Soffa, also from Eindhoven, is looking into making a generator for its middleware layer." Senior research fellow Benny Akesson, ESI's liaison to the Dynamics project, adds: "It's interesting to see this ecosystem starting to form."

Backward compatible

According to Akesson, there are basically three ways of using Comma. "This monitoring facility has already been there for years. If you have an interface and all the traces you run through the monitor are compliant, you know you're in a good place. When you update your interface, you can automatically generate a new monitor and feed it the same traces to see whether they still work. As you don't have to do a complete impact

Comma

Comma (Component Modeling and Analysis) is an ecosystem supporting model-based component engineering. It's a combination of domain-specific languages (DSLs) in which the interface between a server and its clients can be specified by three main ingredients: the interface signature, the allowed client-server interactions and the time and data constraints. The interface signature consists of groups of commands, signals and asynchronous notifications. Commands are synchronous: the caller is blocked until a reply is received, whereas signals are asynchronous: they do not block the caller and do not require a reply. State machines are used to describe the allowed client-server interactions, such as the allowed order of client calls and the allowed notifications from the server in any state. Finally, Comma enables the definition of constraints such as the allowed response time, notification periodicity and data relationships between the parameters of subsequent calls.

Camera.interface

```
import "Camera.signature"

interface ICamera version "3.14"

machine camera {

    initial state Off {

        transition trigger: PowerOn
                do:
                reply(Status::OK)
                next state: On
            OR
                do:
                reply(Status::Failed)
                next state: Off
    }

    state On {

        transition trigger: TakePicture(Time timestamp)
            do:
            reply(*)
            Click
            next state: On

        transition trigger: PowerOff
            next state: Off
    }
}

timing constraints
TC1 in state On command TakePicture - [ 10.0 ms .. 75.0 ms ] -> notification Click
TC2 in state Off command PowerOn and reply(Status::OK) -> [ .. 100.0 ms ] between events

data constraints
variables
int nr1
int nr2
DC1 reply(nr1) to command TakePicture;reply(nr2) to command TakePicture where nr1 < nr2
```

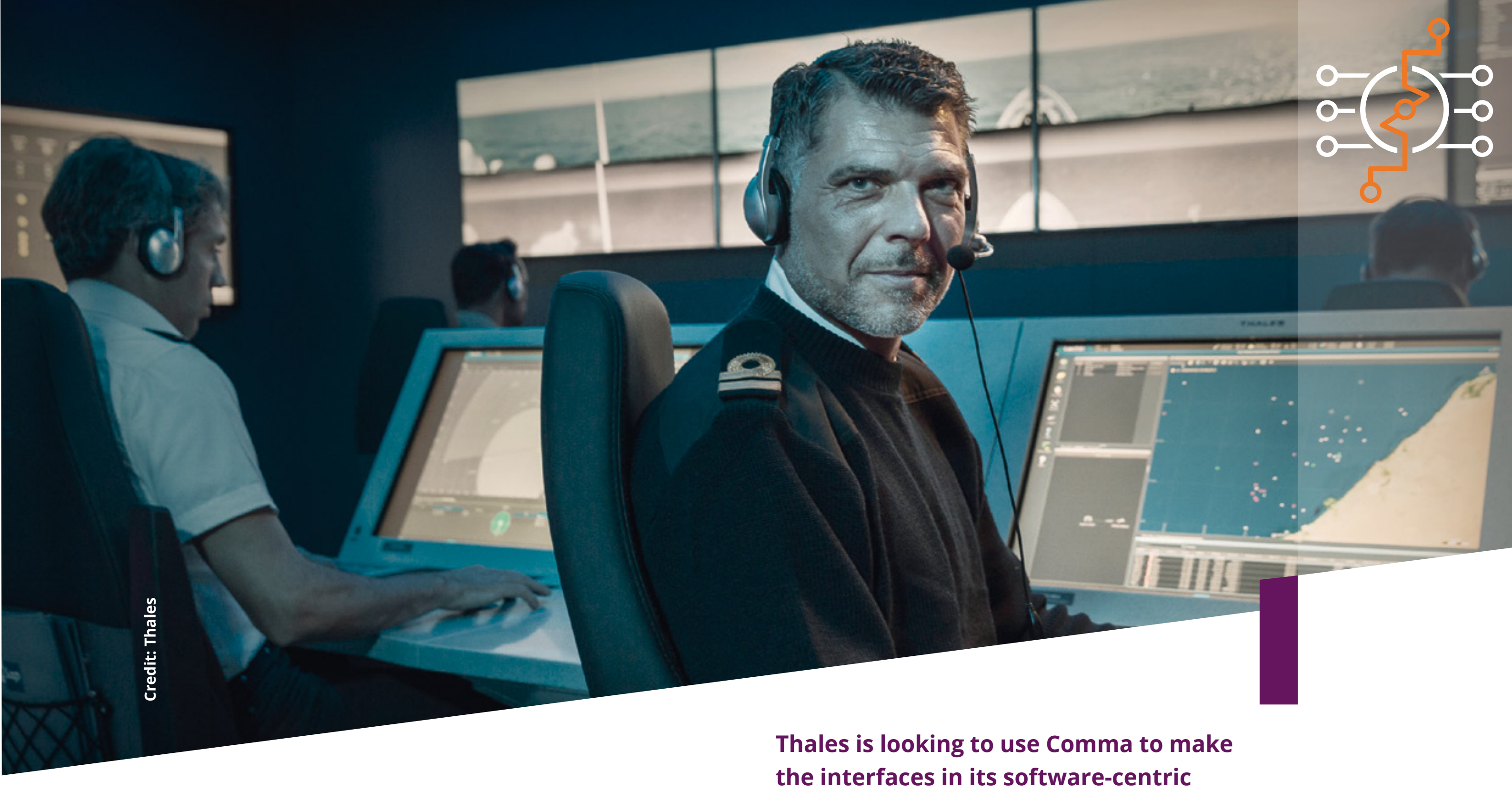
Camera.signature

```
import "Basic.types"

signature ICamera

commands
Status PowerOn
int TakePicture(Time timestamp)

signals
PowerOff

notifications
Click
```

Basic.types

```
type Time
enum Status {
    OK
    Failed
}
```

An example Comma model.



Thales is looking to use Comma to make the interfaces in its software-centric systems evolvable.

analysis of the change manually, that saves you time. The problem with this is that you're now no better than your traces. You need to have traces that are representative of all the desired system behavior."

Together with its industrial partners, ESI is working on a solution based on so-called Petri nets, a formal method using state transition models to study concurrent and distributed systems. "By generating Petri nets for an interface, you can see the possible state transitions that can occur in the protocol," explains Akesson.

"You can then produce tests that cover those possible transitions and thus systematically explore the state space." Philips is now using Petri nets to do exactly that: to create test cases from the Comma specifications. A third approach is to play by a slightly more restricted playbook, continues Akesson. "This is what we're doing in the Dynamics project with Thales. By not using certain constructions in Comma, and using Petri nets in a different way, it's possible to build tooling that can statically tell you whether your new interface is backward compatible and if not, automatically generate an adapter – if one exists. We're now lifting this from a proof-of-concept command-line tool into the Eclipse-based Comma environment, providing immediate developer feedback on why a change is or isn't backward compatible and whether an adapter can be generated."

Open source

This static checking is high up on Philips' wish list as well, divulges Van der Munnik. "The main benefit for us at the moment is still the dynamic conformance checking while running the test cases, but maybe some of that can also be done statically. Furthermore, we want to extend the Comma framework with the ability to create smart stubs and simulators for clients and servers. And we're looking into reverse-engineering interfaces by automatically constructing Comma models from execution traces – but this is still more in the research phase."

At Thales, Noltes is hoping to get Comma out of that research phase and into the modeling practice. "As part of our work with Philips and Thermo Fisher Scientific, we're extending Comma with the concept of components, i.e. objects with multiple interfaces," states Hooman. "These interfaces are often interrelated, which means that if you do an action on one, the state of another changes as well. We're developing a component that lets you express the relations and possibly the timing constraints between the interfaces. We're also looking into testing multiple interfaces."

To further the spread, the partners are working on open-sourcing the framework. Hooman: "We're defining a kind of Comma core in the form of an Eclipse plugin, which others can extend, for instance with their own generators." Van der Munnik underlines the importance of this development: "It adds to the maturity of Comma. What started as a research project is now a product that can actually be used by developers, in terms of UI, speed, ease of installation and so on. By making it open source, we're hoping that others will contribute back into Comma, thereby extending and improving the framework even further."


I ON
LN2 Level ind interface
1116664
H2

Adaptive systems and machine reasoning

To ensure their lasting fitness, effectiveness and efficiency, tomorrow's high-tech systems learn and reason to adapt to their dynamic environments and tasks. They increase their resilience against disturbances or attacks and also become increasingly autonomous, requiring less human supervision and intervention. This has at least two consequences: these systems change over their lifecycle and they're partly or fully based on AI.

Building such systems is to a large extent uncharted territory. It requires a true extension of state-of-the-art systems engineering, as both consequences call for novel or adapted systems engineering processes and methodologies. These new approaches need to cover the full lifecycle of the systems from cradle to grave, acknowledge and address the differences between AI and traditional, control-based systems and integrate the very different development techniques of data or knowledge-driven AI. The expertise area of adaptive systems and machine reasoning is focused on their development.

Challenges

Adaptive systems profoundly challenge the traditional assumptions of systems engineering and control. For example, it's no longer possible to always know how exactly AI-based systems are going to perform if they adapt their behavior during operations – but that breaks traditional certification processes, mandatory for medical equipment and other safety-critical systems. Integrating machine learning and reasoning has far-reaching implications during the design phase as well: our understanding of these techniques is limited. We fail, for example, to grasp tipping points within system behavior, where minor deviations in complex situations lead to very different results – breaking functional flows and their analysis.

Acceptance is another big challenge. When people die from a human mistake, that's generally accepted as a sad fact of life. However, a machine making the same mistake is unacceptable, even if the chances of that happening are demonstrably much smaller – people want systems to run flawlessly. For systems to be perfect, however, we would need perfectly engineered and manufactured technology – which is not attainable.

Tackling the challenges

Adaptive systems monitor their environment, analyze the observations, plan, execute based on that analysis and then observe the effects. We base future systems on this loop. It encompasses inference and learning, for which the AI toolbox contains techniques suitable to build adaptive systems, including probabilistic reasoning and reinforcement learning.

Where are we today?

It's early days for truly adaptive systems. Tangible use cases are found in the most advanced industries, such as automotive, where carmakers are experimenting with self-driving vehicles. The high-tech industry is cautious regarding that level of autonomy but is taking steps towards smart manufacturing and Industry 4.0, smart buildings and other applications.

Consequently, ESI's activities in this expertise area are largely research oriented. Together with TNO Automotive, for example, we're looking at adaptive energy management for electric vehicles. The idea here is that reasoning about the situation, the state of the vehicle and the driving behavior allows for optimizing the system's performance and range.

One of the projects exploring this area in the high-tech industry is Asimov. In this collaboration with Thermo Fisher Scientific, as part of a broader European program, we're using digital twins to train AI for optimal machine calibration and verification. The Asimov project is explained in more detail on pages 62-65.

AI trains itself match fit on a digital twin

With the ever-growing complexity of high-tech systems, it's increasingly desirable that they can optimize themselves. The European Asimov project, co-initiated by Thermo Fisher Scientific, aims to achieve this through the combination of digital twins and artificial intelligence. With the help of TNO's ESI as one of the partners, this should result in a generic approach for fully virtual training of AI algorithms.

Artificial intelligence continues to best people in all kinds of games. It all started in 1997 when IBM's supercomputer Deep Blue defeated chess champion Garry Kasparov. Another well-known example is Alphago, developed by Google Deepmind, which even the best Go players can't outsmart since 2017. Recently, the card game bridge also fell prey to the advance of artificial intelligence.

"Wait a minute," thought Remco Schoenmakers, director Data & Artificial Intelligence at Thermo Fisher Scientific. "Those deep learning algorithms could be interesting for electron microscopes as well. After all, you can compare the alignment of such a system to a game. You win if you move the microscope from an unknown state to the correct configuration the fastest and the most effective. I saw a big similarity."

Together with his colleagues, Schoenmakers started an investigation into whether and how he could apply advanced AI techniques to the calibration of Thermo Fisher Scientific's electron microscopes. "The adjustment differs all the time," says Schoenmakers. "We want to keep the beam precisely on the right spot, within the nanometer. The alignment and calibration have to be meticulous for our customers to obtain the correct metrology results when researching new energy-efficient transistors in the semiconductor industry or determining the structure of proteins and viruses in the development of new drugs and vaccines. There are all kinds of influences that you have to take into account. Think of temperature variations, pressure changes and wear - the sources are plentiful. Some things are quite stable and things run smoothly for months. Other things you have to adjust almost every day."

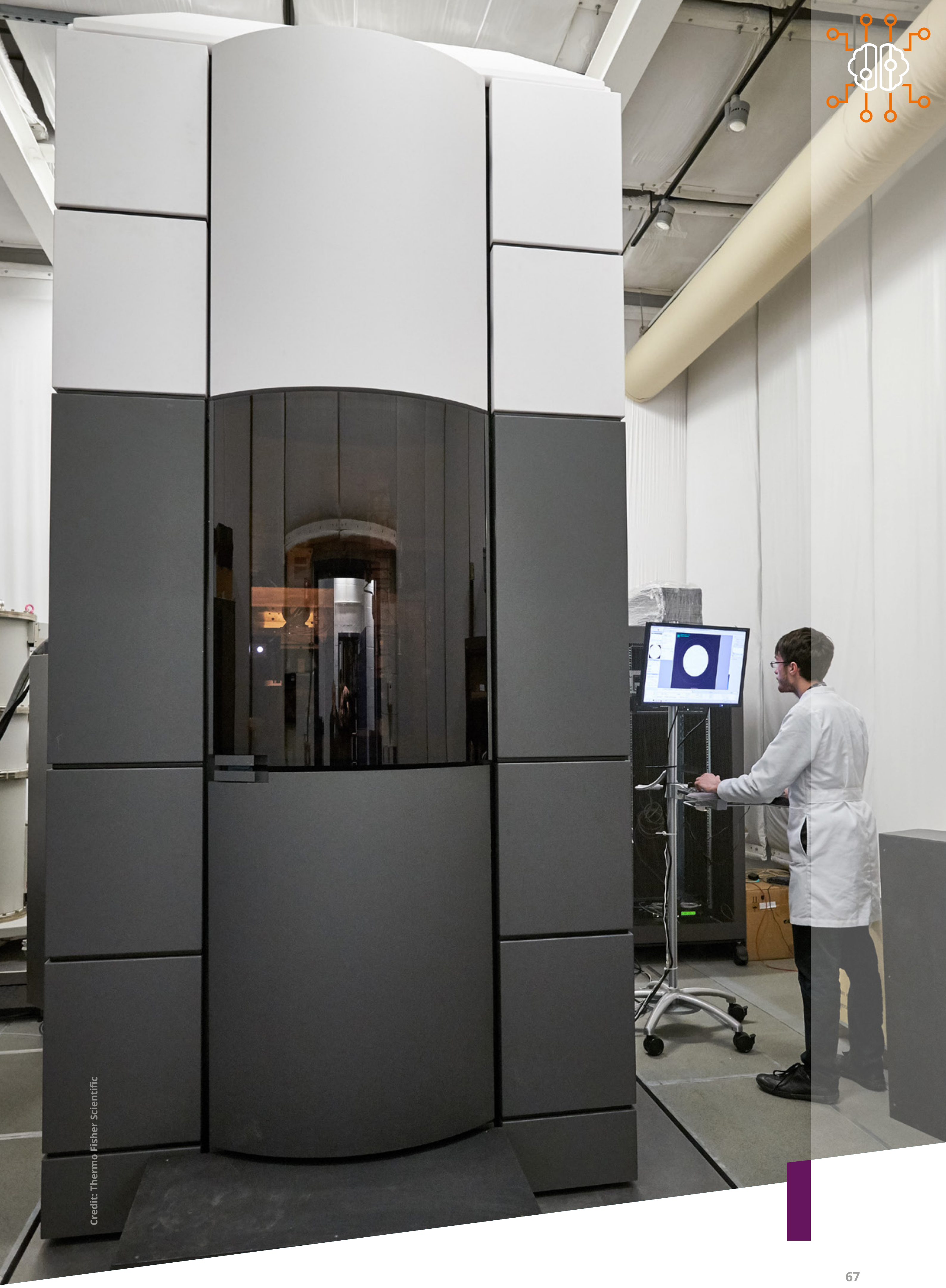
Digital twins

Thermo Fisher Scientific first contacted ESI. Soon after, Eindhoven University of Technology (TU/e) and consultancy firm CQM joined. "Together, we looked at where the opportunities and challenges lay," says Schoenmakers. "We realized pretty soon that we wouldn't be able to train the AI algorithm on a real microscope. The amount of variations required to understand what's going on is just way too large. There may be billions."

The idea arose to build a digital twin of the electron microscope. Schoenmakers: "Let's model the relevant aspects of the instrument. By applying variations on those models, you can create synthetic data and use it to train your neural network."

A virtual copy of a complex system like an electron microscope is easier said than made. "It would take us five years before we would have developed a complete digital twin," Schoenmakers laughs. "There are so many aspects to it, from the controls and physics to the system modeling. Ultimately, all of this must be captured in a digital twin to ensure that it responds in the same way as the real system. We've opted for an incremental approach."

"On the one hand, you want to make an effort to get the link between the digital twin and the real instrument as close as possible," TU/e professor Maurice Heemels adds. "On the other hand, you don't want to lose yourself in this because the learning algorithms must also be able to deal with the differences. Do we expect the trained AI to work on the microscope in one go? Or will it go in steps and does the algorithm get better with every iteration? Can we adjust our digital twin based on measurements on the real system to go through a faster learning curve? All these questions are still open, especially if you look at the problem more generically. One specific approach may work best for an electron microscope, but for other use cases, it might be wiser to take a different route."


Credit: Thermo Fisher Scientific

Credit: Thermo Fisher Scientific

Broadly applicable

The AI exploration of Thermo Fisher Scientific isn't a solo trip. Partly on the initiative of the Eindhoven company, a European project has been set up. Within Asimov ("AI training using simulated instruments for machine optimization and verification"), it's all about the combination of digital twinning and artificial intelligence for the optimization of system performance. A consortium has joined in Germany that's applying the approach to the control of autonomous vehicles. In Finland, a cluster of companies is working on process optimization for the paper and pulp industry.

This generalization is crucial for the other three parties that are affiliated with the Dutch arm of Asimov. "When we stepped in, we immediately asked ourselves how we could apply this approach more broadly," says Jacco Wesselius, a project manager at ESI and the project leader at Asimov. "We already cover a broad spectrum with the German and Finnish cases, but we also approached the Dutch high-tech industry, explained our idea and asked for input. Not everyone has to join right away, but our role is to ensure that the ecosystem benefits from this development. So not only are we supporting Thermo Fisher Scientific in building the digital twin and evaluating the technology, but we're also closely following the other use cases and looking at the approach in a broader context. For example, we map the wishes from the industry to this project, so that we gain better insight into how we can serve other parties and sectors well. At the same time, we validate our plans and ideas against a broader background. That's an important objective for us in the project."

CQM also became interested in the Asimov ideas because of their relevance and applicability in other applications. "We've been building digital twins for forty years, although they weren't always called that," says Jan Willem Bikker, a consultant at CQM. "The logistics world is one of our most important sectors. To give a practical example: in distribution centers, roll containers for supermarkets are filled, but there's often a little air left at the top. That's not efficient of course. By smart optimization, you can minimize that space. As there are many trucks with containers driving around every day, you can save a lot of money. It would be really cool if we could apply the combination of digital twins and artificial intelligence in logistics."

Reinforcement learning

In reinforcement learning, a so-called software agent learns good strategies in interaction with an environment, usually a simulation. The program does this by performing an action for which it's rewarded or punished. If it's rewarded, it knows that it's on the right track and it can build on that; if it's punished, it has to change course. By trying out a lot, the agent learns which moves are smart in which situations. With the aid of a neural network, he can generalize the strategy to circumstances he didn't encounter in the simulation.

Trained AI

TU/e also has a more generic view of the development. The Asimov project uses the AI strategy of reinforcement learning. "That in itself is nothing new; the approach has been around for a long time," says Heemels. "But we're now applying it to a real instrument, where you encounter completely different issues in terms of complexity and where the real-time aspects are very different. In the application at Thermo Fisher Scientific, but also in the other cases within Asimov, all kinds of problems come to light. We're trying to develop methodologies to deal with that."

The TU/e researchers are looking to translate this to other devices and distill the generic knowledge. "I notice in the interaction with Thermo Fisher Scientific that the engineers know their system very well and often know exactly what to do. A lot of things happen automatically," observes Heemels. "Our role is to make it explicit. What's the exact process that you go through to achieve a successful implementation? What steps are you taking? And why? I want us to be able to explain this method in a very detailed way to other parties in the future."

Wesselius: "I strongly believe that given enough time, you can train AI algorithms with which you can perfectly calibrate a digital twin. But what if you do that on a real machine? How robust is that solution in practice? How big can the difference between the digital twin and reality be? The hypothesis is that AI trained on a virtual copy can also do its job effectively and efficiently on a real device, without having to go through all kinds of training sessions."

Demonstrator

Asimov took off in mid-2021 and will run for three years. "We started to describe all use cases a little more clearly, to put requirements on paper and to record deliverables," says Wesselius. "Where's the overlap between the different cases and what are the application-specific properties? In general terms, it corresponded of course; otherwise, the parties wouldn't have stepped in, but it's good to make very concrete what you all have in mind. A conceptual architecture has emerged from this exercise. That's an excellent basis for discussing the approach with outside parties as well."

The next step is to fill in all the details. Schoenmakers: "We've already made a first physical model of our electron microscope with which we can view a certain calibration within the system. We ran it to generate a lot of training data. The question then arises: what's right and what's wrong? Once you figure out which picture is better or worse, you're not there yet. The world isn't that simple. Sometimes you have to go in the wrong direction twice to get to the optimal point. So how often are you allowed to go the wrong way? And how big should the steps be in the first place? We're investigating that now."

Wesselius again: "In Thermo Fisher Scientific's case, we noticed that you're deep in the machine very quickly. This makes it more difficult to go public with the result because it becomes too specialized or because it contains confidential knowledge. That's why at ESI, we're building a demonstrator that from a distance resembles an electron microscope. It's a more general setup where we've defined another problem, including knobs you can turn to make an image better or worse. It's a nice public deliverable without specialist knowledge of Thermo Fisher Scientific. The demonstrator is a perfect platform for us to experiment on a scale that's much less complex than a real electron microscope, but on which we can master the combination of digital twinning and reinforcement learning. We can show that the technology works and where there are still hurdles to take."

Asimov's ambitious end goal is to have a system that calibrates itself and never crashes. Schoenmakers doubts whether this is feasible within the timeframe of the project, but "Asimov must in any case provide the technological direction and define the preconditions within which we can maneuver. On the one hand, it's still very explorative while on the other hand, I can already see the first successes emerging. I'm confident that we'll lay the technological foundation for self-optimizing high-tech systems."


1001
0010
0001
Dark field
Diffraction
Wobbler
Eccentric focus
Magnification
Focus
Fine
Coarse

System architecting systematics

High-tech systems are getting more and more complex. They're becoming more and more data and software-intensive and more and more integrated with their environment to form systems-of-systems. At the same time, development is speeding up, moving to continuous value delivery, while seeing a growing demand for customization. These trends also cause business models to change.

The expertise area of system architecting systematics investigates and develops approaches, methods and tools to advance the art of architecting and help R&D departments and system engineers deal with these disruptors.

Challenges

In systems engineering, it's all about keeping the overall overview. However, not all of the necessary information is readily available. A lot of it is still buried in stacks of documents and the heads of architects. Making all the knowledge explicit is a big challenge.

Many of the experts still operate on their own island, staying within the confines of their discipline or department and communicating in their domain-specific language. This makes it even harder to create an overall overview. Breaking down the walls and connecting all the islands is another big challenge.

System engineers are working hard to address these challenges. Stimulated by methodologies like CAFCR, they've been using architectural diagrams for years to connect the views of the different stakeholders. Now, models are becoming more and more part of the equation. Unfortunately, there's a gap between these engineering models and the high-level diagrams. This gap needs to be bridged.

Tackling the challenges

The key is bringing together all the information from all the stakeholders. Several methodologies already exist that can help here, including CAFCR and model-based systems engineering. In their adoption, it's important to capture state-of-the-art architectures and their rationale, maintain consistency over the multiple views, stimulate feedback and enable (re)use in new designs. Next to these methodologies, reference architectures help to consolidate the architecture know-how.

Where are we today?

ESI supports R&D departments in increasing their effectiveness through system architecting. Using our methods, system engineers can model relations between business and technical aspects. These models show and formalize the rationale for high-level technical and architectural decisions. Furthermore, they serve as means of communication between the stakeholders.

Using industry as a lab, ESI is leveraging a wide variety of approaches, methods and tools to advance the art of architecting. At Canon Production Printing, for example, our DAARIUS methodology was successfully applied to make trade-offs for the design of a core subsystem. DAARIUS provides a traceable underpinning for key design decisions and uses the simple executable models available in systems engineering. At Thermo Fisher Scientific, we helped introduce a reference architecture and establish a standard way of working for one of their product families, while building up and expanding our methodology for reference architecture design.

With Vanderlande, we created a formal configuration model to link the stakeholders from development and sales. This project is explained in more detail on pages 68-71. We're now looking with multiple partners at the transition from a document-based way of working to model-based systems engineering.

ESI 20 YEARS

ESI

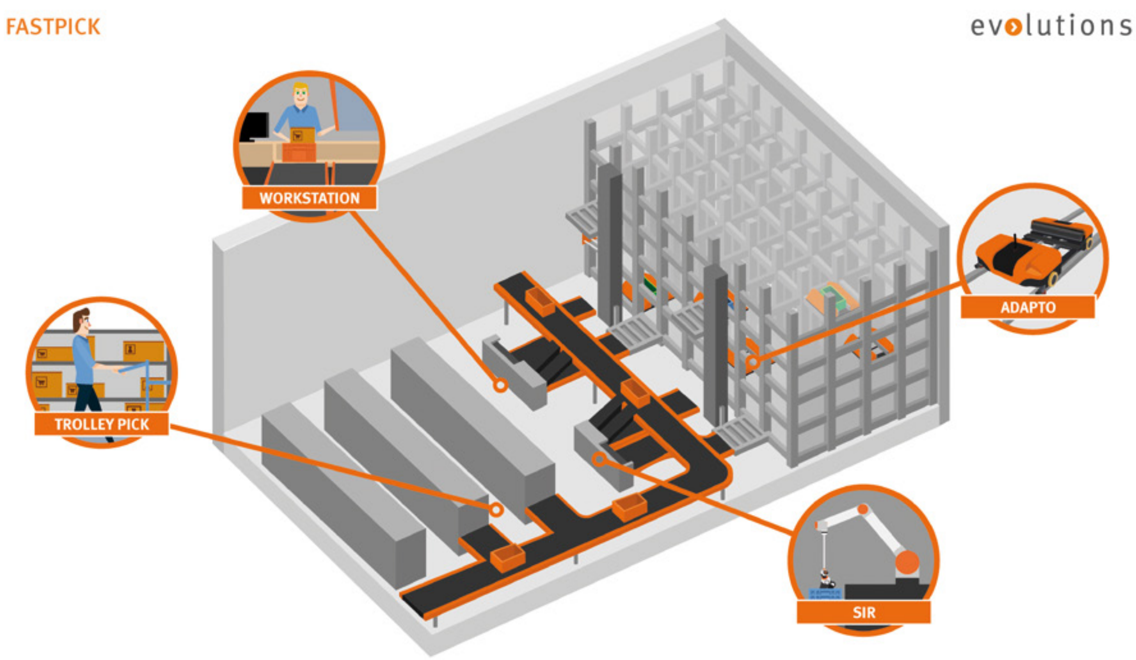
Linking product development and sales in a model-based environment

Configuring, instead of re-engineering, complex high-tech solutions increases development efficiency, reduces errors and saves money. For this, constructing a stable link between configurable modules and customer-facing variations is paramount. Vanderlande and ESI have developed an approach to overcome this configurability challenge.

Not so long ago, Vanderlande built its material handling systems as highly specific solutions tailored to the needs of each customer. Recently, however, the company has changed its course, moving away from such an engineer-to-order concept and switching to configure-to-order, where it creates a portfolio of pre-defined products that can be customized to fit a client's wishes. This new approach increases development efficiency, reduces errors and saves money.

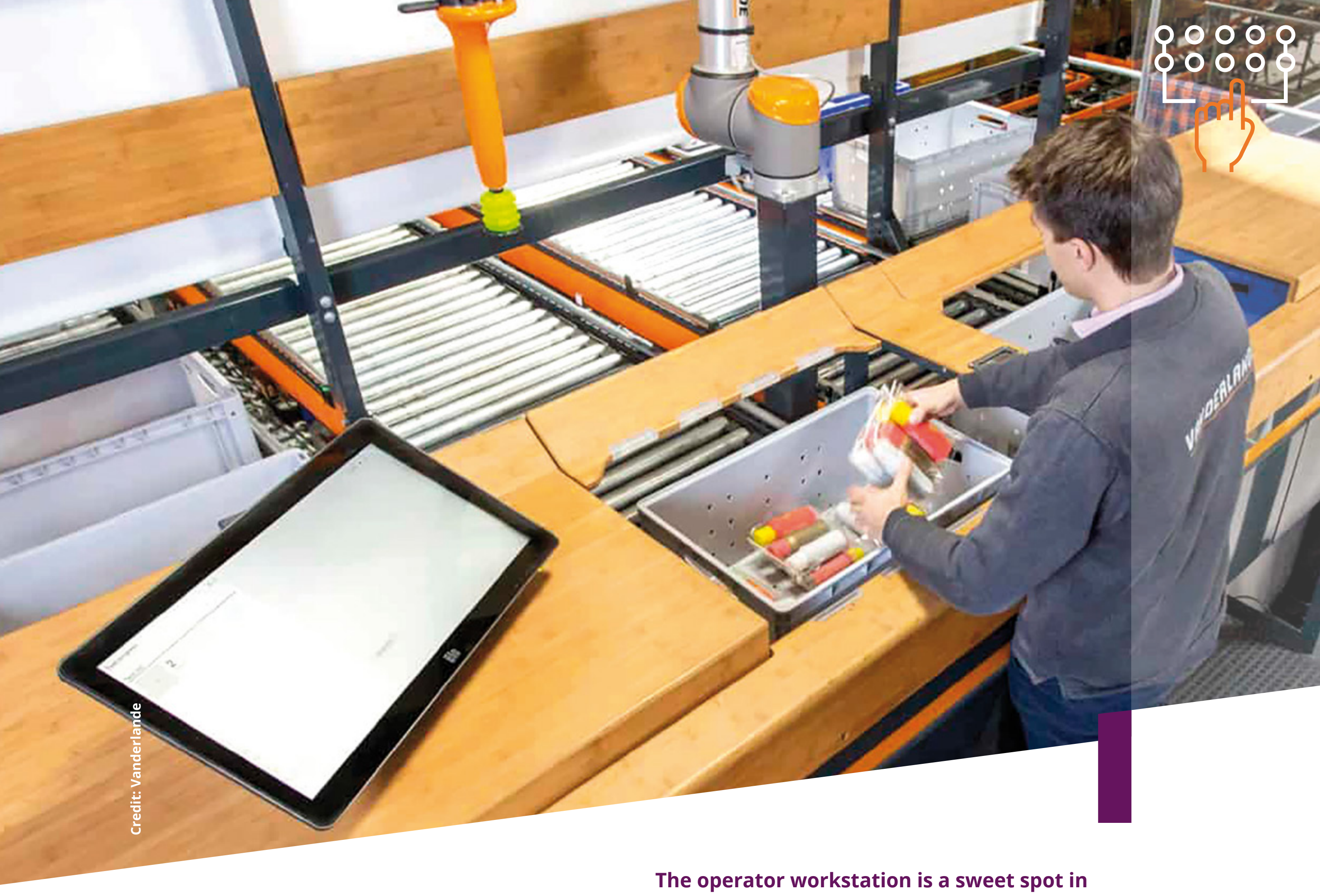
A successful configure-to-order approach requires a very close alignment between the product development and sales pipelines, even more so because it constitutes a major change of process. Sales should offer only configurations as supported by the development roadmap, while the development activities should create designs and decompositions that support the required configurability in customer solutions. "It's a move from making what you sell to selling what you've made," summarizes Ben Pronk, a system architect at ESI.

Pronk and his ESI colleague Alexandr Vasenev, a system architecting researcher, supported Vanderlande in creating a formal configuration model to bring together the stakeholders from development and sales. The project started at the beginning of April 2021 and ran for three months. Its results are now being incorporated at the Veghel-headquartered material handling specialist. Tailored to the specific working environment, they can also be reused at other companies.



Credit: Vanderlande

A typical warehouse automation solution consists of a variety of manual or automated workstations and storage areas connected by a transport system, all managed by a control application.



The operator workstation is a sweet spot in development. Many innovations start there.

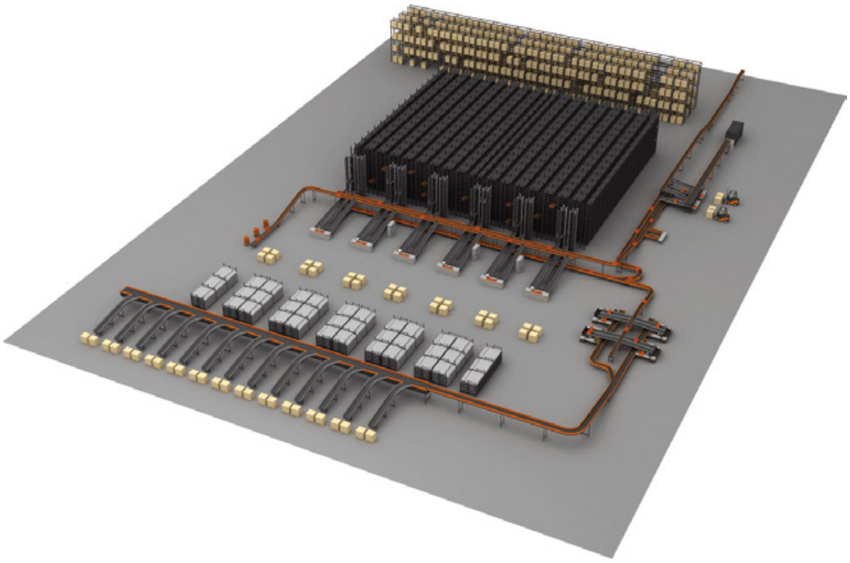
Set combinations

Vanderlande operates in various domains, including airport, warehouse and parcel automation. The project with ESI focused on the warehouse domain. There, a typical solution consists of a variety of manual or automated workstations and storage areas connected by a transport system, all managed by a control application.

"With ESI, we took one such workstation as an example," says Sergey Libert, a system architect at Vanderlande. "In an engineer-to-order approach, to put it black and white, you develop the full layout and ergonomics from scratch for every project. With configure-to-order, you define a limited number of basic elements, from which a confined number of layouts can be constructed. This way, you can make a lot of variants but not everything you can think of – just the ones you want to sell."

This baseline has been developed in a separate process, using requirements derived from market research. "We've created what we call platforms, several set combinations of hardware and software modules providing pre-defined functionalities that our research has shown to fit specific business areas," explains Libert. "For a client, we pick the platform that best suits their purposes and then configure it to further align it with their wishes."

"Of course, you can't satisfy a customer only with what you have," notes Fatih Erkan, who was involved in the ESI project as a system architect at Vanderlande. "That's why we leave some room for tailoring – we're selling at least 80 percent from the baseline; up to 20 percent can still be custom development work."



Credit: Vanderlande

The SysML model developed by Vanderlande and ESI captures all the configuration options of a warehouse solution – starting with the operator workstation.

SysML

The model developed by Vanderlande and ESI captures all the configuration options – starting with the operator workstation. "We organized weekly workshops to determine what information was already available within the company, which tools were being used and who the stakeholders were," says ESI's Vasenev, describing the initial project efforts. "In total, we had more than twenty meetings with product experts, salespeople and management to map the information landscape."
The team found a bunch of information spread across different paper and digital documents. "A team of experts had already defined a standard solution for a workstation. They had done so in a number of Excel sheets describing the possible features and configurations," illustrates Libert. "We took all the information and formalized it in a model for the workstation, linked to a higher-level model, all in SysML."

"At the start of the project, we investigated different modeling languages, and SysML came out on top," adds Erkan. "The choice was made easier because development at Vanderlande used it in their model-based systems engineering efforts as well, together with the Sparx Enterprise Architect tooling. So, rather than reinventing the wheel for a project of only three months, we picked the low-hanging fruit."

"We realized that it would be very beneficial to link the configuration model to the development models," argues ESI's Pronk. "SysML makes it possible to define them in the same environment." Libert: "It allows us to put the maintenance of the configuration model in the hands of the people who also develop the platforms."

Two worlds

A proof of concept was created to show the configuration model's value. Pronk: "We connected the model to an existing UI tool used within Vanderlande. This demonstrates that the generic interface works and that it's feasible to supply the model data to a customer-facing environment."

"We're envisioning a GUI that sales can use to play around with configuration parameters, together with the customer," outlines Libert. "In the back-end, the model maps the implementation functionality on configuration options, defining which parameters can be tuned and which variants can be selected. In the front-end, with the push of a button, the configuration result can be turned into input that can be used to engineer a solution."

"On the customer-facing side, there's going to be this sales tool. Under the hood, it uses the model, which is formally and unambiguously described in SysML," recaps ESI's Vasenev. "On the other side, there's development, which also uses the model and elaborates it to create the systems. So, the model brings the two worlds together, with SysML simplifying the communication with sales and management, while at the same time, forming a solid basis for development."

"Building the envisioned sales tool wasn't part of the project; the goal of the collaboration with ESI was to create the configuration model," Erkan points out. "The example operator workstation model is now in place and being used by our primary stakeholder to 'sell' it to his stakeholders in sales. They've started to look at it and we've learned that they're finding it a big improvement compared to the Excel sheets they're using now. Exporting the model to a tool they know well has been very instrumental in proving its value for sales, too."

Scaling up

There's also still some homework to do, observes Libert. "We need to decide on the sales tool to use for the configuration. In the meantime, we need to make sure that our proof of concept remains top of mind. We're using it in development, but we really need to break through that sales barrier." Erkan: "It's key to convince the business stakeholders, but it takes time to sell something to a salesperson."

Another important item on the to-do list is scaling up. "The operator workstation is very much a sweet spot in development. Many innovations start there. That's why we picked it," says Erkan. "In the project with ESI, we've modeled three more stations. These configuration models are also in place but aren't being used yet."

As a material handling solution has many more constituent parts, the ultimate goal is an integrated configuration model of the entire system-of-systems. "There are spreadsheets on all system levels," Erkan notes. "By formalizing those, a model can be constructed of the complete solution, with multiple levels of configurations connected to each other in a system-subsystem-like architecture."

For Libert, that's one of the key project findings: being able to create one big model from such a diverse set of modules and functionalities. "We have workstations, we have storage areas, we have conveyors. Despite their very different physical and functional characteristics, we can construct an overarching meta-model that can be used in a customer-facing environment. A meta-model, moreover, that's language, tool and domain agnostic – it's not bound to SysML or Sparx and it can be applied outside Vanderlande."

Reusable

This paves the way for ESI to deploy the approach at other high-tech companies. Many have similar model-based systems engineering (MBSE) discussions with configuration as an important topic. Pronk: "I see a lot of potential for broad application in the high-tech industry."

"It's definitely reusable," reckons Vasenev. "It's actually built on knowledge gathered from ESI's MBSE network. After interviewing partners like ASML and Canon Production Printing, as well as external companies on their best practices, we tailored the findings to the Vanderlande context, which sped up the integration into existing processes, tools and modeling approaches. And now, we can inject the results back into the network."

Share, learn, accelerate

What makes developing high-tech systems complex? It's not just about applying new technologies but also about the ability to face multidisciplinary challenges in a changing, multi-stakeholder business context. To manage the increasing complexity, the skills of engineering professionals must be in line with their role, tasks, responsibilities and the development phases in which they operate. ESI's competence development program not only helps strengthen their skills but also distributes the research results from one partner across the entire ecosystem and truly embeds them in the processes, people and organizations.

Challenges

In managing their development process, companies make use of multiple phases. Each phase has its own goal, deliverables and way of working and requires a different mixture of people and competencies. In general, three engineering phases are used: shaping the problem and solution domain, guiding and balancing the design and applying engineering approaches, methods and tools.

In the early phases of system development, the level of complexity and ambiguity are high. It's important to understand both the problem and solution domain. To develop a high-level system concept, architects and other key stakeholders need to focus on customer value, the value chain and the business proposition.

To develop a design at a (sub)system level, domain architects and lead designers guide teams and need to understand (sub)system interactions and behavior. Besides functionality, their main focus is on making conceptual choices and trade-offs and managing dynamic behavior of the system.

When realizing a system that covers all stakeholder requirements, engineers work in a team and need to understand the different ways of working, processes, tools and technologies. Their main focus is on the completeness and correctness of the component they're responsible for.

Tackling the challenges

A competency consists of a combination of knowledge, skills and mindset, i.e. what you *know*, *do* and *are*. To develop a competency, all three should be taken into account. At an engineering level, the emphasis is more on the application of approaches, methods and tools in a team environment. Moving 'up' to design and architecting, the focus shifts more towards system-level reasoning, the ability to make trade-offs at a conceptual level and the development of a mindset capable of dealing with the increasing complexity and ambiguity.

Where are we today?

At ESI, we offer three learning concepts to accommodate the development of knowledge, skills and mindset. Firstly, we support *sharing* by organizing multiple networking opportunities, including our System Architecture Forum, our System Architecture Study Group, our Special Interest Group on "System and Software Testing" and user communities for state-of-the-art tooling like Comma (interface management), DAARIUS (system architecting) and LSAT (system performance). Secondly, we accommodate *learning* with practical training courses directly connected to state-of-the-art research results, e.g. in workflow modeling, software rejuvenation, interface management and system performance. Thirdly, we enable the *acceleration* of system competencies through in-house system architecture and system integration programs and multi-company programs for system companies and their strategic supply partners.

Piecing together the systems puzzle is a collaborative effort

In the past twenty years, high-tech systems have become incredibly complex. ASML and its supplier VDL ETG discuss how this has impacted the role of the system engineer. Architecting, they conclude, is becoming more and more a team effort – and so is the training of architects.

Driven by trends like digitalization, globalization, servitization and systems-of-systems, the complexity of high-tech equipment is growing by the day. "More and more disciplines have gotten involved and the amount of information that needs to be managed has increased substantially," observes Ton Peijnenburg, deputy general manager at VDL ETG and fellow at Eindhoven University of Technology's High Tech Systems Center (HTSC). "It's becoming harder and harder for architects to keep the overview."

Roelof Klunder, competence lead for ASML's electrical architects, concurs. "Our machines are getting bigger and bigger, containing ever more parts. While most architects can handle a single subsystem, overseeing the impact of changes on a system level is quite a different story. But that's exactly what's required from system engineers."

According to Jan van Vlerken, vice president of System Engineering at ASML, collaboration is key to keeping his people's work manageable. "The increasing complexity has made it almost impossible for a single system engineer to have a complete overview. For a group, the odds are much better. That's why architecting is becoming more and more a team effort."

Training

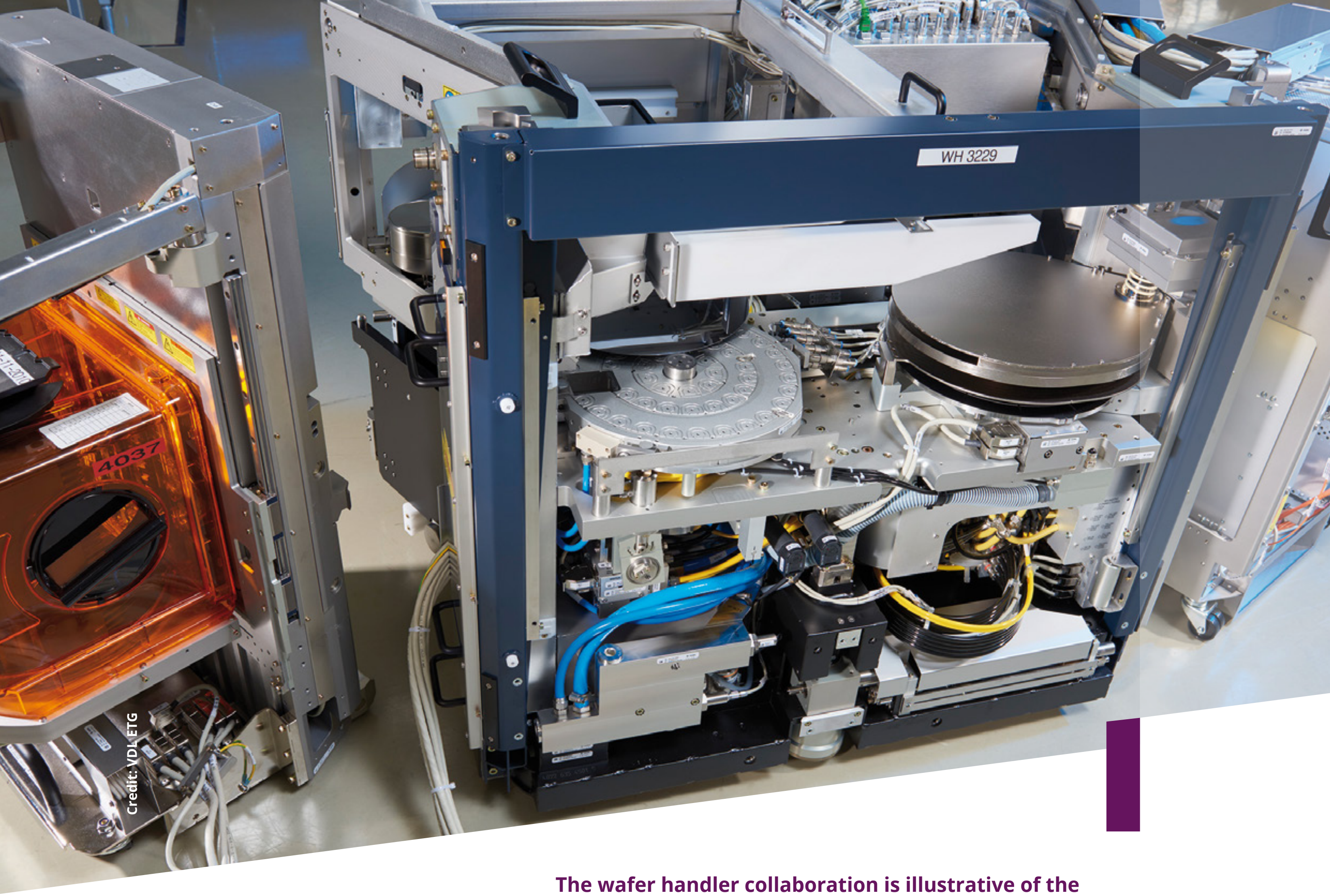
As systems engineering is evolving into a team effort, so is the training of architects. Over the past years, organizations like TNO's ESI have shifted their educational focus from teaching individual courses to providing (in-company) programs involving multiple stakeholders on multiple levels, sometimes even from multiple companies. Taking a real-life business case in a learning-by-doing approach, these programs explicitly aim to develop the participants' leadership and people management skills, in addition to their technical competencies. Peijnenburg highly values the hands-on approach. He points out that although architects have to have a solid background in the basics of systems engineering, their skills get honed on the job, so that's also the logical place for training. "You need to realize that you're not the architect because you know all the details but because you can mold the general picture. You mostly learn that by doing."

A key prerequisite is technical assertiveness – the ability to give pushback to stakeholders. "It's a combination of persuasiveness and steadfastness," explains Peijnenburg. "As a system engineer, you engage in discussions with stakeholders using substantive arguments based on your intimate knowledge of the general picture, and

ESI's competence development program

2022 not only marks the 20th anniversary of TNO's ESI but also the 20th anniversary of ESI's competence development program. What started as a collection of separate technical training courses two decades ago has evolved into a comprehensive in-house portfolio geared toward advancing the leadership skills of system engineers in a changing business context. The program is based on the BAPOC model, in which the classic business, architecture, process and organization dimensions have been augmented with a focus on culture – in an increasingly international playing field, the impact of culture becomes ever more important.

In celebration of the 20-year milestone, ESI has initiated a virtual roundtable to put system architecting and competence development in a broader and historical perspective. With its partner companies, the changing role of the system engineer is being discussed. Joris van den Aker, manager of ESI's competence development program and moderator of the roundtable, concludes that this role has become increasingly complex, not just technically but in all BAPOC aspects as well. Collaboration in multidisciplinary teams has become key. The challenge for companies is to bring together the technical and BAPOC dimensions. ESI is supporting them in this change process through programs that approach system architecture from a strategic point of view.



Credit: VDL ETG

The wafer handler collaboration is illustrative of the ecosystem's growing involvement in ASML's product realization, making this process more complex and the role of system engineers even more important.

the details to a certain extent. But there are always people looking for holes in your reasoning. So, not only do you need to be convincing, but you also have to stand your ground in the dialogue. ESI's system architecting program has proven to be very instrumental in mastering these skills."

"During my time at Philips CFT, thirty years ago, starting engineers were given two years to grow into their new role, two years to earn themselves back and another two years to make money for the company," Peijnenburg recalls. "Today, we don't have that luxury anymore. Architects are thrown in the deep end straight away. And then they get so consumed by their day-to-day work that they hardly have the time to reflect on what they're doing and discuss it with their peers, while I see a growing need for just that. Coaching and training on the job are efficient ways to fulfill that need."

ASML has its own supplementary program for system engineers. "Starting architects get assigned a senior colleague as their coach," details Klunder. "Next to that, they can participate in our Architect Development Program, with all kinds of technical and non-technical exercises involving stakeholders from different disciplines. In the next level of our architect training, the Senior Architect Masterclass, their social and behavioral skills are deepened. As things change continuously, there's also a need for a regular kind of training. We still need to close this gap."

Common sense

System engineers are trained to perform a balancing act. "You're never going to build something that 100 percent adheres to all the requirements because then you're either never going to finish it or it's going to be very expensive," asserts Van Vlerken. "It's up to you as the architect to decide what's important and what's less important in order to arrive at a system that's good enough." Peijnenburg also sees a balancing act between two responsibilities. "On the one hand, system engineers need creative space as lead constructor. At the same time, they need to structure information to make decisions as leader of a design team."

Methodologies and tools can help bring structure, but they can also be perceived as constraining. "When we tried to introduce requirements engineering at VDL to supplant the traditional Word-based process, this was met with a lot of resentment," Peijnenburg gives as an example. "People are very reluctant to change their way of working, fearing that it will make things more bureaucratic and less efficient." Klunder has a similar experience at ASML. "We've also been piloting and are now rolling out improved ways of working and tooling for requirements engineering because we, too, see the advantages of having everything written down and managed in a structured way. At the same time, we see the bureaucracy that this brings. The challenge is to find a balance such that the advantages of requirements engineering are evident and the creativity and flexibility of the architects aren't impacted."

"It's not about checking boxes for the sake of checking boxes, but it's not just about creativity either," Van Vlerken adds. "It's also about common sense. As a system engineer, you should be creative but at all times use your sound judgment." Peijnenburg agrees. "You need to have the common sense to switch from the details to the general picture," he says. "When you focus too much on the details, you won't be able to argue about the general picture – which is an essential capability for an architect."

To keep a grip on things, many companies are jumping on model-based systems engineering (MBSE), in collaboration with partners like ESI. "In the aerospace and automotive industries, they completely rely on modeling and they're very successful in using it to create products that are extremely reliable," notes Peijnenburg. "In our high-tech equipment industry, however, MBSE hasn't really taken off yet."

Van Vlerken has a hard time seeing it fly on a system level anytime soon. "Mechanics, electronics, software – every domain has its own digital version of the system. How can these models help a system engineer make the right choices, in a way that the benefits outweigh the costs? In my view, we'd need to have a system-level metamodel, but how do we construct and maintain something like that? Our systems are really complex and require regular revisions over time, and therefore maintenance. We have yet to find a practical and pragmatic solution for a digital twin on the system level."

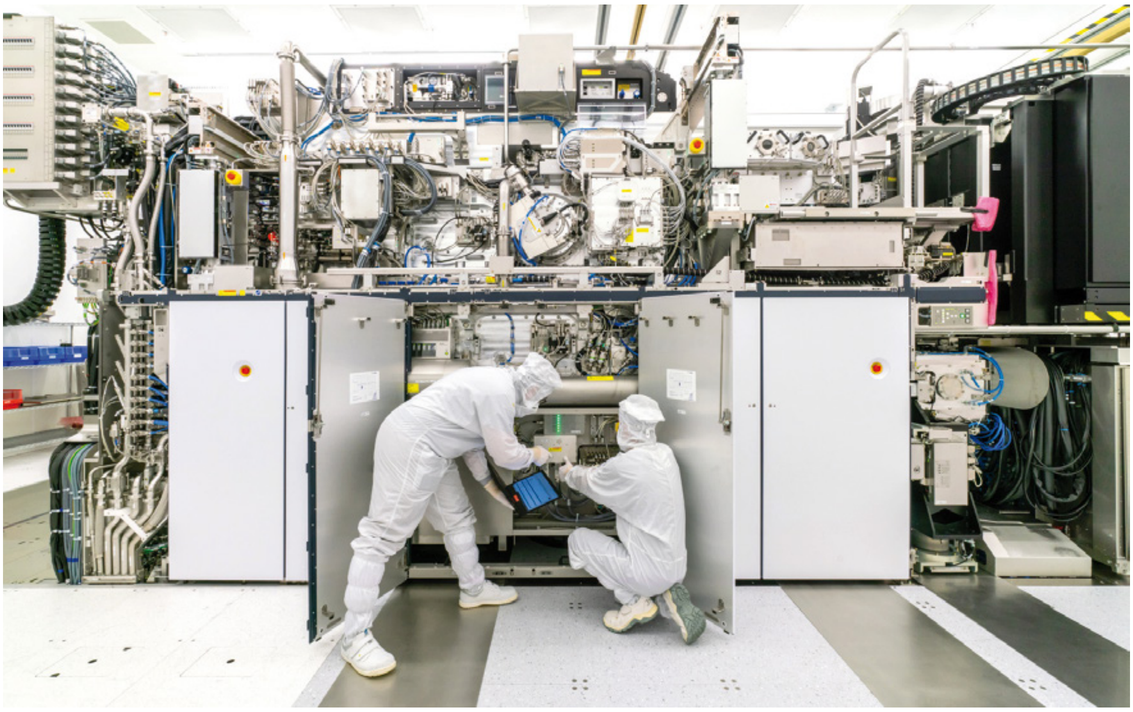
"It's becoming harder and harder for architects to keep the overview," notes Ton Peijnenburg, deputy general manager at VDL ETG and fellow at Eindhoven University of Technology's High Tech Systems Center (HTSC).

Credit: Bart van Overbeeke

"Having external modules like VDL's wafer handler takes the stakeholder game to a whole new level," observes Jan van Vlerken, vice president of System Engineering at ASML.

Credit: Bart van Overbeeke

"By expanding our architecting network outward, we could take better advantage of supplier knowledge," says Roelof Klunder, competence lead for ASML's electrical architects.


Credit: ASML

Ecosystem

In another effort to deal with the increasing system complexity, ASML has started to travel down the route of commonality – reusing identical design elements in multiple places throughout the architecture. "This allows us to do more with the same development effort," says Van Vlerken. "We use a common element unless there's a solid business case for introducing something new – putting the brakes on adopting technology for the sole reason of it being new and fancy. Commonality isn't a goal in itself but a way for us to advance in different areas – not just development effort and cost but also improved learning cycles, risk mitigation and sustainability through reuse."

Commonality is also a topic in ASML's close collaboration with VDL ETG on the wafer handler. "We're discussing how we can introduce common elements in the architecture of that module," Peijnenburg explains. For Van Vlerken, this exercise, as he calls it, has already yielded some valuable observations. "One of the insights we've gained in this case is that it's unwise to keep the module common on the top level, while configurability with commonality on the lower architecture levels can be very useful." To Peijnenburg, the discussions nicely demonstrate the value of good architecting skills. "Here, too, we see the importance of being able to look at the bigger picture – from both sides. Technical assertiveness is playing a similar big role, stimulated by ESI's system architecture training, where a combined team from ASML and VDL used the commonality case to advance their skills."

The wafer handler collaboration is illustrative of the ecosystem's growing involvement in ASML's product realization, making this process more complex and the role of system engineers even more important. "Having external modules like VDL's wafer handler means that our system engineers have to work with suppliers, their architects and engineers and their ways of working," observes Van Vlerken. "This takes the stakeholder game to a whole new level. We need to build bridges to learn to understand each other."

Klunder sees room for improvement there. "Our suppliers have dedicated knowledge that we're not fully utilizing yet. By expanding our architecting network outward, we could take better advantage of that knowledge, for example, to make our product realization simpler, cheaper or more reliable. And instead of handing them a finished design, we could involve them earlier by tapping into their innovative strengths."

Working for a supplier, Peijnenburg acknowledges that there's still something to gain. "We can definitely contribute to our customers' architecting processes. For this to fly, however, both sides need to really get to know and appreciate each other. We have to move even closer together and act as one. After all, in the end, our interests are intertwined: we benefit if ASML sells good systems and ASML benefits if we deliver good modules. Our architects need to become more aware of that."

ESI 20 YEARS

The art of architecting meets software sculpting

In a virtual roundtable, Canon Production Printing, NXP, Philips, Thales, Thermo Fisher Scientific and Vanderlande discuss how the increasing system complexity has impacted the role of the system engineer. Their consensus: software has shuffled the cards.

The high-tech systems engineering landscape has been profoundly reshaped by digitalization. “Our products are no longer confined to the boxes they’re being shipped in,” observes Klaas Wijbrans, fellow architect at the Chief Architect Office of Philips. “They’ve become part of a network, an ecosystem, a system of systems, interacting with the cloud and other IT infrastructures. Our architects now have to think about a system with components at different locations and a business case that’s about continuous value delivery, across the product’s entire lifecycle.”

With new technologies like artificial intelligence and digital twin, systems are increasingly interacting with the real world. “They’re connecting in more ways, collecting more data and also generating more data,” says Clara Otero, senior director of system innovations at NXP Semiconductors. “As a result, our business focus has shifted from just the chip to the entire edge, and the scope of our architects has expanded to include the data needs of our customers. We’re delivering products with more digital functionality and more software.”

“Digitalization is opening up a plethora of new functionalities we can offer to our customers. At the same time, our customers are operating in an increasingly digital environment that they want to connect to our products,” notes Henk Thomassen, department head of platform development and planning at Canon Production Printing. “This has a considerable impact on our systems, which need to be much more flexible to be able to cope with the ever-changing environment they’ve become an integral part of. This calls for a smart architecture that offers that flexibility and accompanying ways of working like Agile and Scaled Agile.”

Maarten Verhoeven, executive director of architecture and integration at Vanderlande, is seeing a similar spillover of the software mindset. “In the past, system engineers had to be extremely careful as the whole supply chain counted on them to deliver a flawless design. Mistakes cost a lot of money. Software has changed that completely, with the Agile mindset of failing fast encouraging them to quickly learn from their mistakes and deploy an improved version. That’s quite a turnabout.”

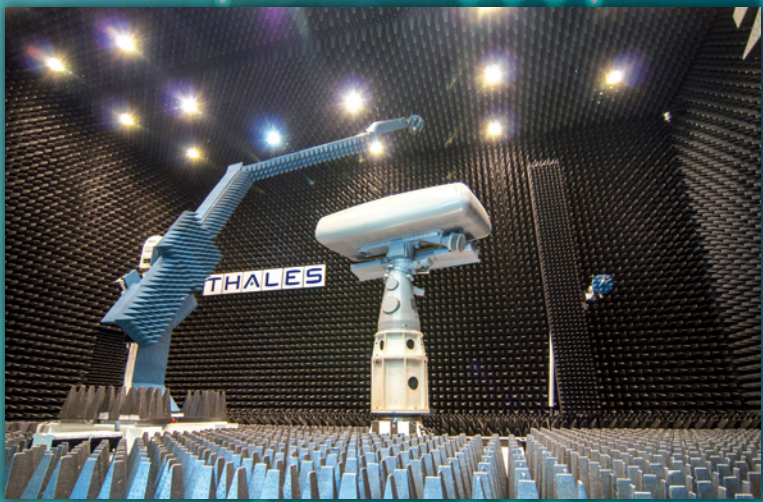
Software has become the dominant discipline and that has upset the relationships on the system level. “System engineers used to throw work over the fence, to put it bluntly, relying on the hardware and software specialists,” says Roel Aalbers, technical director of systems at Thales in the Netherlands. “Now, they need to understand what’s going on there. They need to know their fair share of the digital transformation and the challenges it brings, for example in cybersecurity and the cloud.”

More involvement

At Thales, the Agile mindset is causing a bit of a clash of cultures. “Our system engineers have strict requirements to adhere to, regarding legal matters, safety and security, for example. Those have to be set first, calling for a more traditional waterfall approach. This is at odds with the agility advocated by some software engineering methods,” explains Aalbers.

His colleague and system architecture expert, Jacek Skowronek, points out that because of the short cycles in software development, it’s getting increasingly hard for system engineers to keep up. “Software is already very fluid, but things like Agile and DevOps make it even more so. System engineers want to know exactly when a component will be finished and what features it will have. An agile software team usually is unable to answer those questions. That can feel like a disconnect.”

“In systems engineering, it’s all about keeping the overall overview,” says Vanderlande’s Verhoeven. “I don’t see that as much in the agile software community, where the focus is more on quickly delivering small improvements. The challenge is to keep the system working in this continuous stream of updates. Sometimes, the scale tips too far to the other side. We’re still learning to maintain the balance.”



The director and expert from Thales would like to see more interaction between the system and software side. “Both engineering disciplines should think more about the implications of their work on each other’s level,” Aalbers believes. Skowronek agrees. “In a cloud-based environment, they have access to virtually unlimited resources,” he gives as an example. “In our systems, however, hardware constraints also matter.”

Wijbrans subscribes to the sentiment. “In the past, software people had to think about making their code fit. With the cloud giving them unlimited resources, they don’t have that worry anymore. But they have to realize that they aren’t getting this for free. I’ve seen my share of poorly designed systems result in customers getting huge bills from their cloud provider, especially when data traffic scales.”

Awareness gap

At Thermo Fisher Scientific, they, too, recognize the disconnect. They’ve chosen to partially resolve it by connecting the two worlds on the core functionality only. “We have a hardware update every 2-3 years but a new software release every three months,” illustrates Olivier Rainaut, system architecture R&D manager at the company’s Eindhoven site. “We link the two on the so-called minimum viable product of the system or module and find agreement on the interfaces and requirements there.” This approach has its price, Rainaut admits. “With frequent software updates, lifecycle management becomes a challenge.”

“We make a conscious effort to bring both sides together,” adds his colleague and principal system architect, Jamie McCormack. “In a system reference architecture, for example, we make explicit that each part of the functional system decomposition is a jointly owned exercise between hardware and software. I’m the reference architect for our TEMs, our transmission electron microscopes, and I have a software architect working right next to me on that reference architecture.”

"The challenge is to keep the system working in the continuous stream of software updates," says Maarten Verhoeven, executive director of architecture and integration at Vanderlande.

"We make a conscious effort to bring both the system and software side together," says Jamie McCormack, principal system architect at Thermo Fisher Scientific in Eindhoven.

"Modern system architects need to have sufficient knowledge about software," says Klaas Wijbrans, fellow architect at the Chief Architect Office of Philips.

Wijbrans fully agrees that to bridge the "huge awareness gap," as he calls it, both sides need to better understand each other. "Modern system architects need to have sufficient knowledge about software. They don't have to be able to write it themselves, but they do need to know how it can benefit the product. This background partly comes from training and partly from coaching. At Philips, we're putting together people who are aware of this gap with those who are less aware, for example, as sparring partners or peer reviewers in projects."

The key, Skowronek believes, lies in close collaboration, forced if necessary. "I once participated as an architect in a software scrum team. Now I'm quite senior, but I initially struggled to comprehend their world. Gradually, however, I started to understand it better. Likewise, it can benefit software experts to become more involved in system issues. By stimulating this on both sides of the fence, we can bridge the gap between waterfall and Agile. I'm convinced the solution lies somewhere in the middle."

Lightweight models

Modeling can be very instrumental in bridging the gap, too. "In the past three years, we've completely drawn our system reference architecture in Visio," Rainaut explains. "These models show all the constituent parts and their interfaces. We see that new system engineers quickly get an overview and software people are eager to dive into the Visio files to get to know the system, how it works and how everything is connected."

"It works both ways," says McCormack. "The software guys did Visio drawings as well and when they showed them to me, I started understanding their world better, in the same way that they've come to understand our world better by looking at our models. As our mutual understanding grows, our way of drawing also changes for the better. In our newest models, our decompositions and interface descriptions show much more a meeting of the hardware and software world."

Skowronek feels that such 'lightweight' models are often undervalued. "When many people talk about modeling these days, they're thinking of extensive tools with all kinds of bells and whistles like code generation and validation techniques. It's not about tooling; modeling is a basic technique I think everyone should master without a tool. Every engineer – system and software – should be able to draw a picture of a module and its interfaces using only pen and paper."

"Every engineer – system and software – should be able to draw a picture of a module and its interfaces using only pen and paper," says Jacek Skowronek, system architecture expert at Thales in the Netherlands.

"Virtual prototyping helps us bring our products to the market more quickly and at lower costs," says Henk Thomassen, department head of platform development and planning at Canon Production Printing.

"When we're fully running on models that we can't share, the chain will be blocked," says Olivier Rainaut, system architecture R&D manager at Thermo Fisher Scientific in Eindhoven.

"Whether it's in Visio, Powerpoint or on a piece of paper, simple diagrams can be very powerful, very telling, next to model-based systems engineering," Aalbers finds. "You can hang them on the wall and look at them with a team to get an immediate overview of the system or parts of it. These kinds of simple models are very helpful to share knowledge with new people or in a new project. For example, we have a couple of big new developments where we're using such diagrams to make insightful what the stakeholders want."

The well-known A3 method, named after the size of the paper the diagrams are put on, is based on exactly these principles. It helps to capture relevant architectural aspects for a specific goal, such as an architectural refactoring or handover of responsibilities. Developed by the University of Twente, Philips and ESI's Gerrit Muller, the method has been applied within many companies.

MBSE

For Thermo Fisher's Rainaut, the lightweight modeling in Visio was a great stepping stone to full-fledged model-based systems engineering (MBSE). His people are now using the open-source Capella tool, through the cloud-based offering from Obeo, for system decomposition in a SysML-like way – hardware and software, up to interface mapping. Rainaut sees this catching on quickly. "Especially with our architects who already have some background in other types of modeling, like finite element analysis, or scripting, for example in Matlab."

MBSE is also Vanderlande's answer to the challenge of bridging the gap between systems and software engineering. "Our system architects use models to capture the complete system behavior. Our software engineers contribute to these models from their area of expertise," outlines Verhoeven. "Thus, by modularly expanding the systems engineering model, as it were, they keep the overview together so that they can keep building together."

At Philips, MBSE is of growing importance. "It does create a tension," Wijbrans notes. "A systems engineering model needs to be complete as it specifies the whole design, whereas architecting is about creating an abstraction of the key issues. The challenge we're looking into now is how to keep the two consistent."

Canon Production Printing is increasingly using modeling to bring together the disciplines in virtual prototypes. "We're putting our system architecture and the resulting designs in what we call digital lab models," explains Thomassen. "These are fairly complex, multidisciplinary models that allow us to test our actual software before the hardware is available. Thus, virtual prototyping helps us bring our products to the market more quickly and at lower costs."

Together with partners like ESI, the companies are making strides to get MBSE to really fly in their organization. They're participating in multiple projects focusing on the adoption of methodologies and tools to support the modeling process itself as well as the maintenance of the resulting models. Through training programs and workshops, their architects are cranking up their MBSE skills.

Ecosystem

With MBSE getting more and more established, this creates a whole new challenge – an interfacing conundrum in the ecosystem. "Through co-development, insourcing and outsourcing, we're partnering with a host of suppliers and other companies," elaborates Rainaut. "How do we ensure that they can use our models? That's becoming increasingly difficult as we're collaborating with more and more partners, all with their own tooling. At some point, it's going to impact our efficiency. When we're fully running on models that we can't share with them, the chain will be blocked."

The solution, in the eyes of Rainaut, lies in the interface between the partners. "The ideal situation would be if everybody were to use open-source models that can be easily exported and imported. Unfortunately, that's not going to work because we simply can't impose our choice of tooling on our partners. The only way to ensure an unobstructed exchange of models is to connect both sides through standard interfaces. Otherwise, we'll be left with no other option than sending back and forth Word and Excel documents."

Sometimes that actually is the only option. "In our part of the industry, you'd have a very hard time finding organizations that have adopted MBSE, especially on the customer side," Skowronek from Thales notes. "Our customers almost all have their own environment. I don't see them switching to our tooling anytime soon, let alone to an open-source alternative. Of course, we'd like to have an all-encompassing model-based environment across the value chain, but it would take many, many years to get there. For now, we mostly communicate with them through Word and Excel. I admit that it's not very fashionable, but it works."

Aalbers does see some progress. "Particularly in the hardware department. Systems engineering seems to be a bit lagging. In my view, there's some more bridging work to be done there. It would already be a step in the right direction if we could have a collaborative digital platform where we could share architecture and design information with partners and customers."

Platformization

Digitalization has also added to the growing system complexity. To keep the development manageable, companies have set their sights on platformization and modularization. "Technological progress is accelerating at such a rate that it's become very hard to monetize innovations for one-offs. That's why we've moved to an approach where we base multiple products on the same platform," explains Thomassen. "We're basically cutting up our monolithic software architecture into functional blocks with standardized interfaces and interactions. These blocks can be developed by smaller teams, relatively independently. Like bricks, they can be combined to form complete systems by iteratively building bigger units, each of which has to be testable on its own."

At Vanderlande, they've made a similar move from customer-specific projects to more generic developments. "We're creating modular building blocks, as small and as confined as possible," details Verhoeven. "This approach allows us to have multiple teams simultaneously working together on the same system without the complexity getting out of hand. The big challenge is to keep the interfaces in tune and decoupled, which is where the art of architecting comes in again."

A platform-based architecture, Thomassen maintains, creates synergy in two ways: through the reuse of existing functions and components and by providing flexibility for the future. "We need to look ahead more, to product avenues that aren't very tangible yet but that hold potential. With a flexible platform architecture, we can easily take advantage of those opportunities as they present themselves."

"A collaborative digital platform would be a step in the right direction," says Roel Aalbers, technical director of systems at Thales in the Netherlands.

"In the past, our system engineers 'only' needed to know about our components. Now, they also have to understand how our customers are using them," says Clara Otero, senior director of system innovations at NXP.

Vanderlande has tasked a special group of specialists with ensuring that the platform developments align with the overall systems engineering efforts. "Our architecting guild actively monitors this alignment," says Verhoeven. Canon Production Printing has set up platform reference projects to take care of this, Thomassen points out. "Next to the traditional product projects, we have special teams of architects watching over a technology platform and defining a reference architecture for it."

Knowledge transfer

All the new methodologies and tools are very useful to provide insight and overview to system engineers, but they're no substitute for good 'old-fashioned' knowledge sharing. To support this, training providers like ESI have shifted their educational focus from teaching individual courses to offering (in-company) programs involving multiple stakeholders on multiple levels, sometimes even from multiple companies. Taking a real-life business case in a learning-by-doing approach, these programs explicitly aim to stimulate the cross-fertilization.

"Agile only gets you so far. So does comprehensive tooling," Aalbers contends. "You won't really get anywhere without a solid background in systems engineering. That's partly education but mostly training on the job. You learn the most from experienced colleagues. Sharing knowledge with other high-tech companies, for example through the ESI network, is very important as well. To advance in this complex world, we can learn a lot from one another."

At Philips, too, there's a strong focus on knowledge transfer. "I lead the company's architecture community," Wijbrans clarifies. "This community currently encompasses about 900 people – system, software and solution architects. We bind them together through coaching and an extensive internal program of cross-training. Our system engineers can strengthen their 'digital' side with courses on cloud, connectivity and data, for example. Having such a network is extremely valuable as, in my opinion, people can only gain experience by doing and sharing."

With the increasing digitalization, NXP's system engineers need to raise their level of knowledge to include the customer application. "In the past, they 'only' needed to know about our components. Now, they also have to understand how our customers are using them, what systems they're making with them," says Otero. "That's a big step – a struggle for some – but a necessary one to be able to identify future requirements. For our customers, sharing that knowledge with us can be an equally big step."

Too much customer or product focus can also create local heroes – system engineers who know everything about a very specific development. Having worked in such a culture for a long time, the older generation still tends to be old-school like that. However, in a world where increasingly multidisciplinary groups of architects are taking on the ever-growing system complexity by sharing models and platforms, local heroship doesn't work anymore. At Canon Production Printing, Thomassen is already feeling a new wind blowing. "In our architect community, I notice the younger generation using shared models and teaming up more easily."

ESI Symposium

Once every one and a half years, the ESI Symposium is organized. This event has grown into a valued gathering of the Dutch high-tech equipment network and, in particular, ESI partners. We're proud of the number of participants, which has increased over time to over 400.

The symposium includes parallel sessions and an innovation market. The format is characterized by active participation from the high-tech industry. The overarching theme is carefully chosen each time to match the trends that the industry is experiencing and to reflect the needs of ESI's industrial and academic partners. The innovation market features about ten selected exhibitors with demos from relevant projects and programs.

Challenges

For a few years now, ESI has involved co-organizers in the symposium. From 2017-2021, Holland High Tech helped to actively engage Dutch high-tech SMEs. In 2021, Covid-19 forced an alternative virtual edition, IDEW21 (International Digital Enablement Week). It featured six sessions in one week, three of which were co-organized with partners TNO, SERC (US) and OFFIS (Germany). We're also keen to expand international participation. This year, SERC and DLR (of which OFFIS became part on 1 January 2022) are each again organizing one of the tracks.

Until 2019, we hosted the ESI Symposium in the Auditorium on the campus of Eindhoven University of Technology. In 2020, the entire symposium was online. For 2022, we've relocated to De Schalm in Veldhoven.

Tackling the challenges

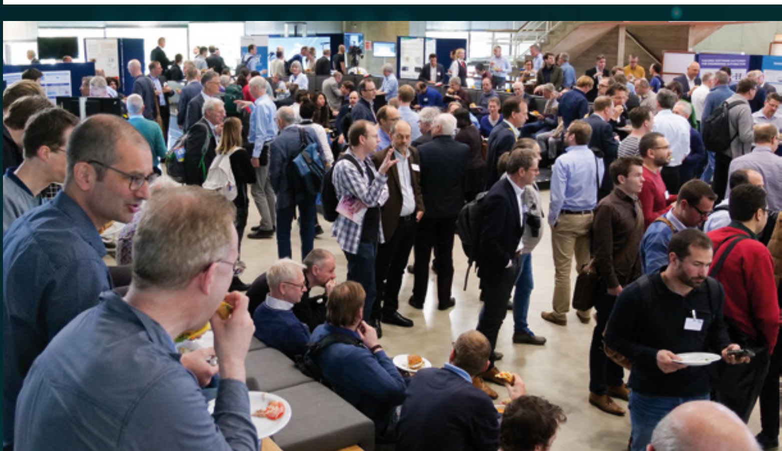
For each of the parallel sessions, a track title is carefully chosen, fitting the theme and linked to a project that's running at one of the partners. A representative, industrial speaker with specialist knowledge will be approached in good time to introduce the sub-theme and lead the session. In terms of content, the presentations for each session track are well aligned. ESI is represented in each track. Special attention is also paid to the added value of human capital. The exact format varies from workshop to parallel session to, this time, embedding the human capital agenda in the various tracks.

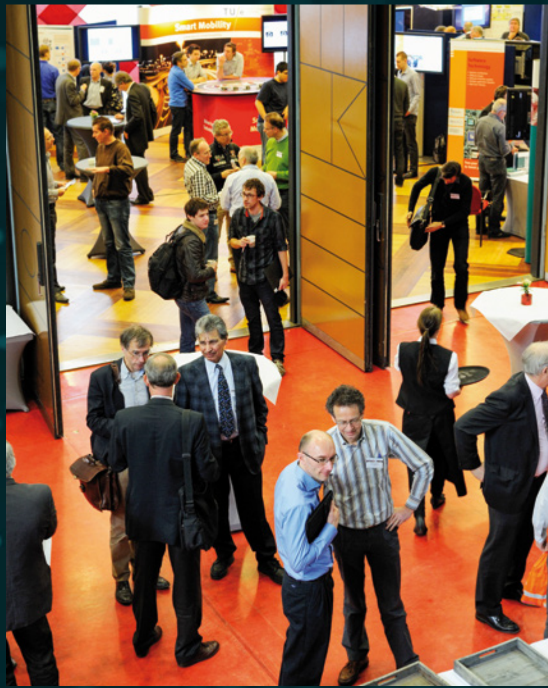
Where are we today?

On 27 September 2022, the symposium's 11th edition is organized in the year that ESI celebrates its 20th anniversary. For the first time, the venue will be De Schalm in Veldhoven. In the run-up to the symposium, the big question is whether measures will be necessary as a result of the corona pandemic. Different scenarios are being prepared: physical, hybrid and completely online. All partners will participate in the innovation and poster market.

Themes

2022	“Integrating systems”
2020	“Digital enablement”
2019	“Intelligence, the next challenge in system complexity”
2017	“Managing complexity in embedded systems engineering”
2016	“A paradigm shift”





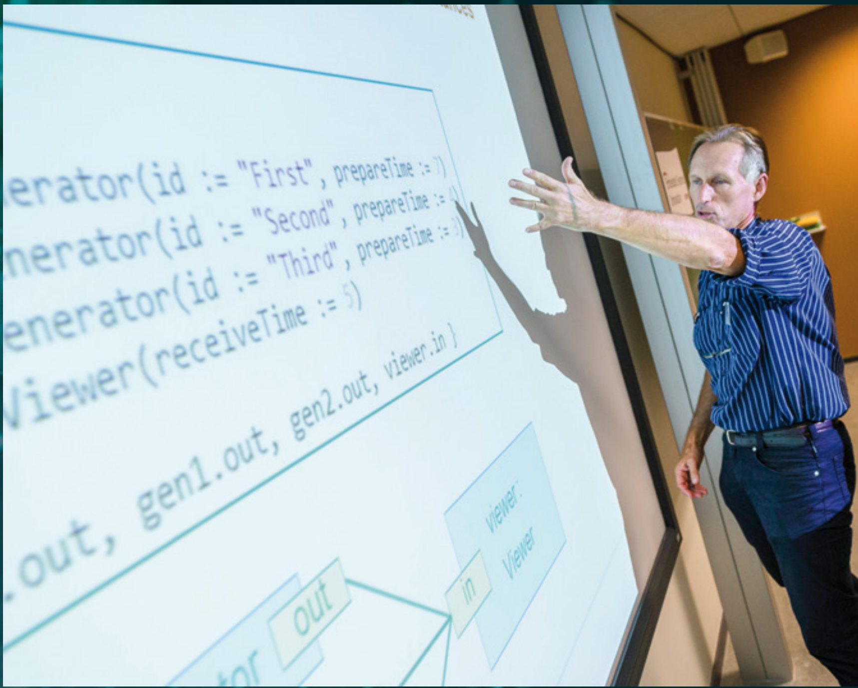
Industrial keynotes

2022	Bernhard Quendt (Thales Group)
2021	Jim Koonmen (ASML)
2019	Henk van Houten (Philips)
2017	Paul Hilkens (Canon Production Printing)
2016	Julian Bartholomeyczik (Bosch)
2014	Kees van der Klauw (Philips)
2013	Jan Mengelers (TNO)
2012	Gerben Edelijn (Thales)
2010	Michiel Peters (Vanderlande)
2009	Harry Borggreve (ASML)
2008	Anton Schaaf (Océ)



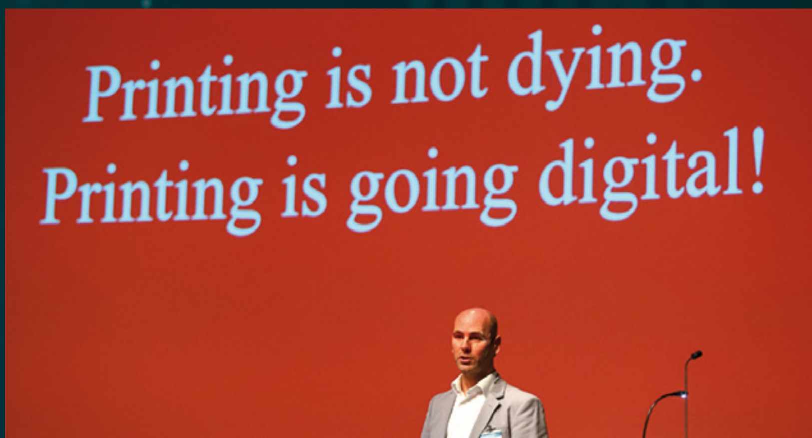


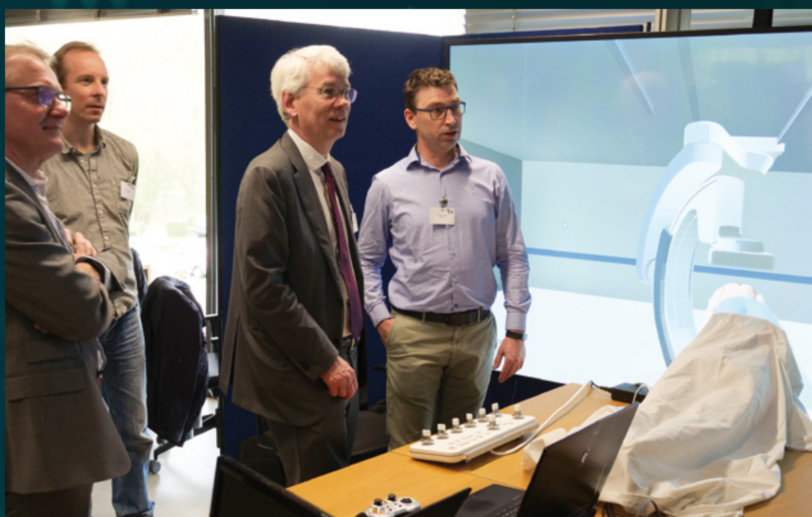




Academic keynotes

2022 Gail Murphy (University of British Columbia)
2021 Jan Bosch (Chalmers University of Technology)
2019 Edward E. Lee (University of California)
2017 Dinesh Verma (Stevens Institute of Technology)
2016 Dieter Rombach (Fraunhofer IESE)
2014 Luca Benini (ETH Zürich)
2013 Jan Madsen (Technical University of Denmark)
2012 Frits Vaandrager (Radboud University)
2010 Rolf Ernst (Technical University of Braunschweig)
2009 Rudy Lauwereins (IMEC)
2008 Lothar Thiele (ETH Zürich)





Future of ESI

High-tech equipment is becoming increasingly cyber-physical. Their core is still physical but the added value is coming more and more from software and the connection to other systems. This has a considerable impact on systems engineering and, therefore, on ESI and its research.

High-tech equipment is also becoming increasingly the product of co-creation within ecosystems. This leads to all kinds of new developments, like supply chains being optimized through digitalization, innovations from third parties being integrated and systems becoming part of systems of systems to maximize their value. These developments have an equally profound effect on ESI's agenda.

Research scope

In early 2022, we conducted in-depth discussions with all our industrial partners to understand their long-term business strategies and the related innovation requirements. This has resulted in a roadmap for key topics we need to address going forward. Many shared system challenges lie ahead that set ambitious goals for ESI research in the coming years. Common themes include model-based and platform-based systems engineering, the role of AI in systems engineering, hyper-automation, knowledge access and systems engineering covering the full lifecycle.

These themes will steer the development of methodologies in our research projects. Model-based systems engineering will gain an ever-stronger foothold in the high-tech industry, requiring not only new ways of working and new modeling tools but also new techniques for bringing the vast amounts of implicit and tribal knowledge into the new environment. This is necessary to lower the knowledge barrier and reduce the learning cycle for new personnel. At the same time, it brings new challenges, like model management and legacy models.

For similar reasons, we're broadening our research focus to expand the systems engineering toolkit in other ways. New methodologies are required to deal with the speed, dynamics and agility of systems in the digital era, with the incorporation into systems of systems, taking into account aspects like security, safety and dependability, and with the design, validation and lifecycle support of embedded AI.

Impact

Over the past two decades, ESI has come to cater to a growing part of the Dutch high-tech equipment industry. We've expanded our partner board and our collaboration with other companies, as well as our university network. We're also starting to embark on advisory and consultancy journeys to apply our expertise to other societal challenges where system complexity and the lack of oversight are hampering progress, such as the energy transition and mobility infrastructure.

Our scope has always extended beyond the large organizations on our partner board. Because of digitalization, more and more companies, of all shapes and sizes and from a wide variety of domains, are calling on us to help them with their system challenges. As a result, we're increasingly working together with SMEs, e.g. through our multi-company system architecting acceleration programs.

We aim to further expand our impact while roughly staying the same size. With 60 people, ESI is sufficiently compact for colleagues to know each other and quickly make cross-expertise and cross-project combinations. Rather than growing the team, we're looking to train more implementation partners in our methods and tools, thus creating a much larger group that can help the industry manage complexity while freeing up our research fellows to focus their attention on new developments. With the help of our implementation partners, we can also offer more SME-oriented services, such as shorter consultancy projects.

In another effort to increase our impact, we're exploring the feasibility of setting up ESI satellite offices in the Netherlands. Smaller companies often operate in local ecosystems and by moving closer to their place of business, we want to lower the threshold for them to seek our assistance. The focus of these collaborations would be more on the valorization of research results than on the research itself. Together with the University of Twente, we're investigating the possibility of establishing the first of these satellites there.

International network

To execute our mission of creating value for the Dutch industry, we need to be world leading in what we do. On a European level, we'll continue our involvement in the Electronic Components and Systems Strategic Research and Innovation Agenda, through projects in the Key Digital Technologies Joint Undertaking. This program is coordinated by the Inside industry association (formerly known as ARTEMIS-IA). ESI is represented on the Inside board.

On a global scale, we plan to intensify our collaboration with our international peers – DLR's Institute of Systems Engineering for Future Mobility (Germany), Fraunhofer IESE (Germany), SERC (US) and KTH's TECoSA (Sweden). We see great added value in combining our roadmaps and results. With Fraunhofer IESE, we're revamping the successful series of cross-border workshops bringing together our mutual industrial networks. With the other centers, we're joining forces at the end of 2022 in a shared issue of INSIGHT, the magazine of the International Council on Systems Engineering (INCOSE).

Facts & Figures

Funding

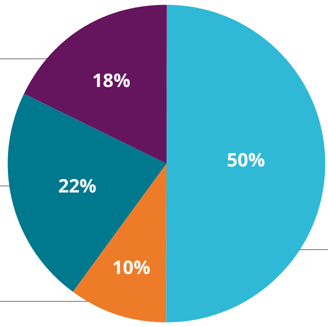
Funding in total
2013-2021
(x EUR million)

73,235

TNO funding

National funding

International funding



Private funding
Contributions partners
Industry

Contribution by Ministry of Economic Affairs (until 2017)
Subsidy of "TKI-toeslag"
EU subsidies (national and H2020)

Partners

16 ▸ 9 | 7

Industry | Academic

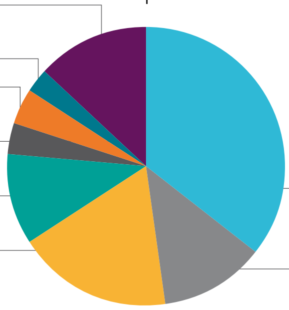
Scientific publications 2003-2021

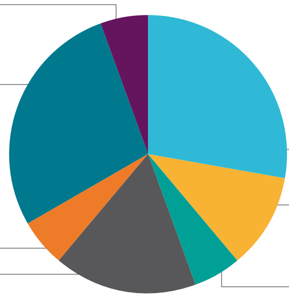
1,047

Projects

Research projects
Completed and
actual

122





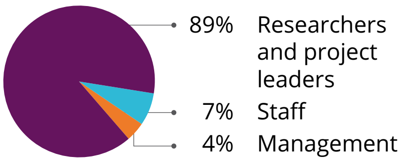
EU projects
Completed and
actual

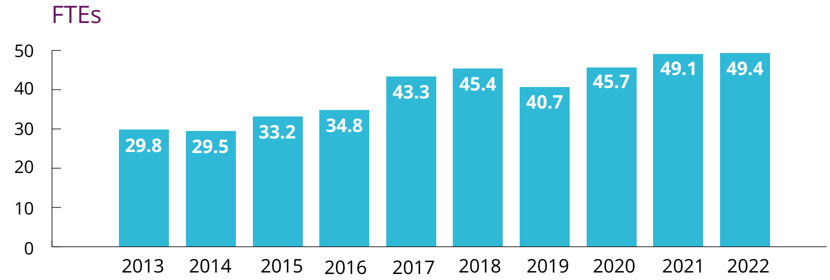
18

ESI

Actual
part-time professors

7





Average age **47**

Competence Development Program

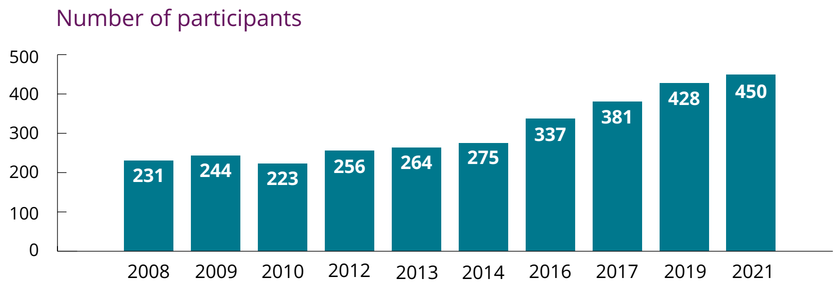
Completed and actual

Programs	**Programs**	**Projects**	**Projects**
Number of educational programs on site at industrial companies	Number of participants	Number of educational projects	Number of participants
69	**1,159**	**91**	**1,274**

Symposium

Number of symposia from 2008-2021

10



Networking

System Architecture Study Group (SASG)

Exchange of experiences with fellow software and system architects. Started in 1997 by Philips CFT, since 2003 organized by ESI.

Members	Meetings on site	Participants
250	**3** per year	**35** per meeting

Special Interest Groups (SIGs)

System Architecture Forum (SAF). Focused on practical system architecting and the application of architectural information and knowledge. Jointly organized by ESI, University of South-East Norway and Stevens Institute of Technology.

Members	Meetings	Participants
>60 worldwide	**2** per year	**20** per meeting

MBSE. Jointly organized with INCOSE NL.

Meetings	Participants
4 per year	**10** per meeting

System and Software Testing (SST). Exchange of knowledge and experiences in various domains.

Meetings	Participants
5 per year	**15** per meeting










TNO
ESI











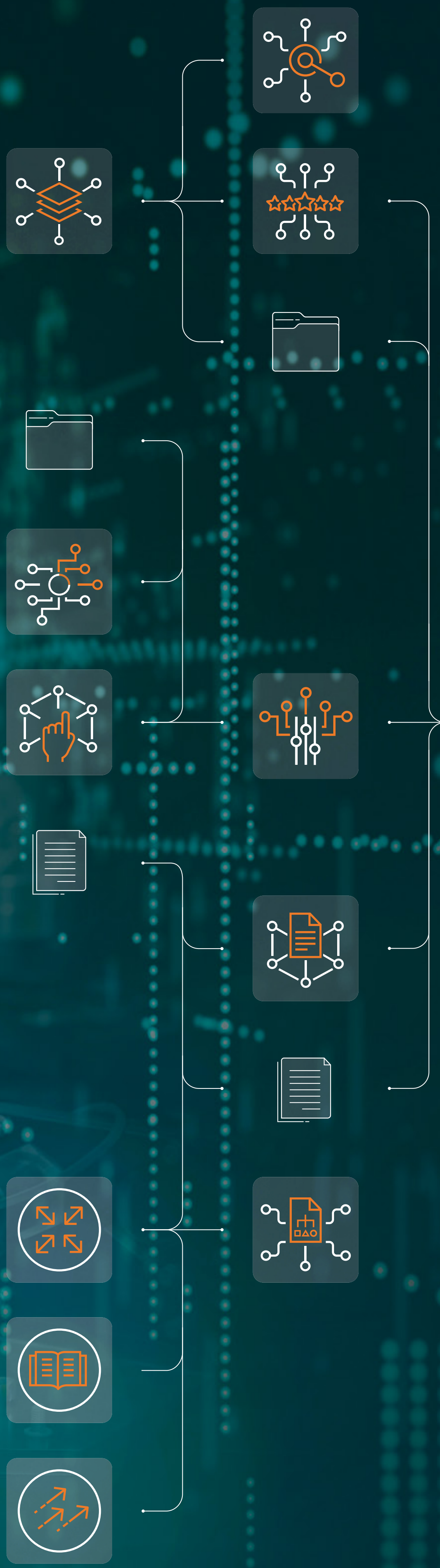












Managing Complexity

Within TNO, ESI is an open and diverse innovation research center with strong partnerships with industry-leading high-tech companies.

Through the development of new methods and techniques for system design and engineering, ESI addresses the ever-increasing complexity facing the high-tech industry in the systems it creates. Its extensive research program aims to advance the high-tech industry by improving the lead times and effectiveness of their product innovation processes, as well as the functionality, quality and societal impact of their products.

We are a leading applied research center for systems design and engineering in the high-tech equipment industry. We work in close collaboration with Dutch industry as well as in strong association with the fundamental research of academia, both nationally and internationally. We contribute to society and the economy by driving advances in high-tech systems technology through a strong shared research program, dedicated innovation support, a focused competence development program and various knowledge- and experience-sharing activities. Through our strategic research projects, we show how to put fundamental knowledge - including the latest insights into model-driven engineering - into practice in the harsh reality of industry.

ESI 20 YEARS

ESI

An initiative of industry,
academia and TNO

ESI.NL