



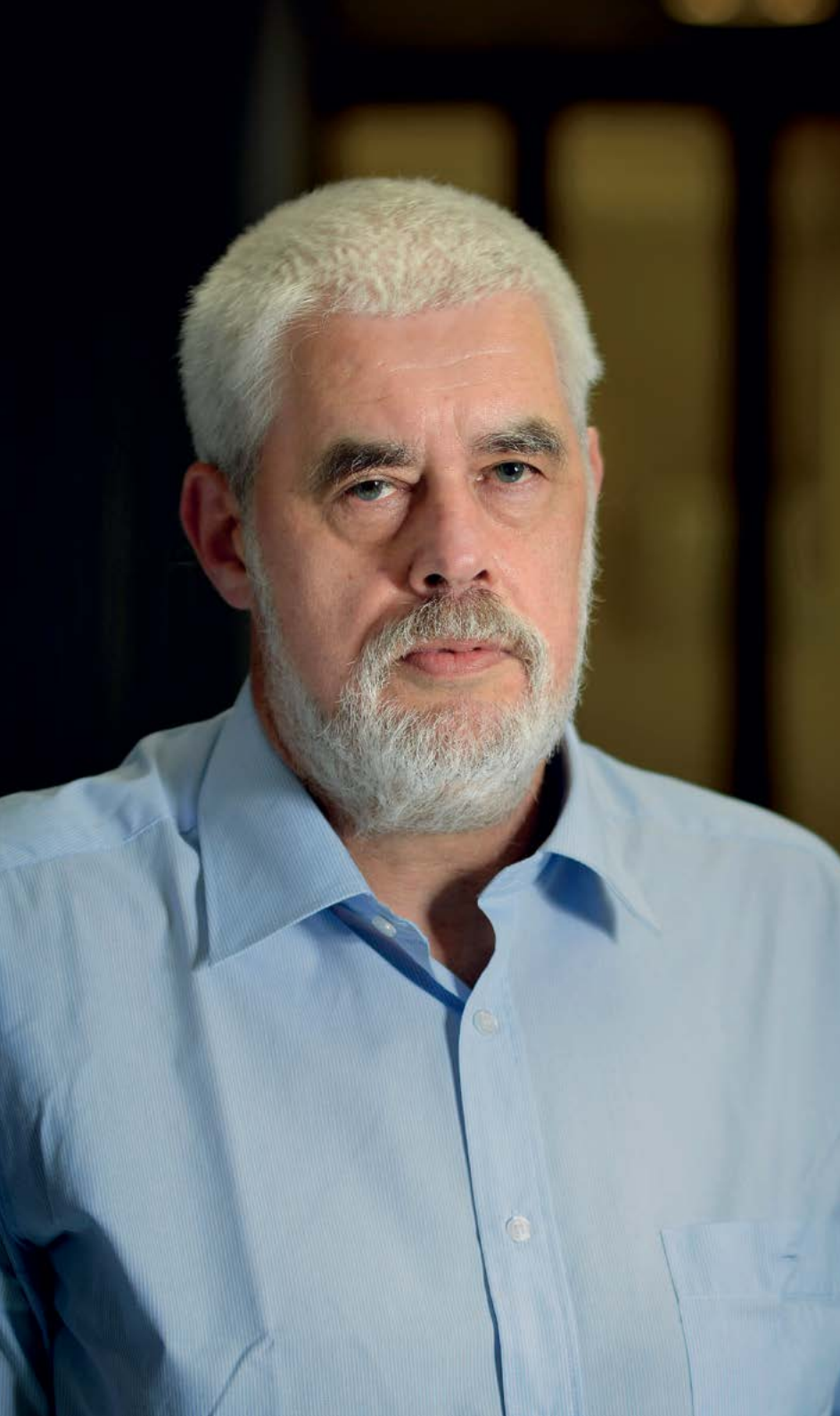
SOFTWARE TECHNOLOGY

PDEng projects 2016

4TU.School for Technological Design,
Stan Ackermans Institute offers twenty
two-year postgraduate technological designer
programmes and tracks. This institute is a joint
initiative of the four technological universities of
the Netherlands: Delft University of Technology,
Eindhoven University of Technology, University
of Twente and University of Wageningen.
For more information please visit: www.4tu.nl/en/

SOFTWARE TECHNOLOGY - PDENG PROJECTS 2016

- 4 **Filip Anastasovski PDEng;** System for Code Generation for Testing Amplifiers at NXP Mobile Audio Department
- 6 **Sarwan Dar PDEng;** A Simulator for Spark
- 8 **Dongqi Hu PDEng;** Analysis Toolset for Calibration, Performance and Diagnostic (CPD) Applications
- 10 **Ana Kostadinovska PDEng;** Data Linkage Architecture for Population Health
- 12 **Aldo Daniel Martínez Márquez PDEng;** MBIF: An Extensible Framework for Automatic Generation of Test Cases and Wrappers
- 14 **Vladimir Mikovski Iotov PDEng;** Performance Analysis Framework for a Complex Financial Data Warehouse
- 16 **Irina Nikeshkina PDEng;** ASML Simulation Development Environment
- 18 **Jeroen Redegeld PDEng;** A Framework to Stimulate Innovation Based on Automotive Data Collection
- 20 **Spyridon Skoumpakis PDEng;** Hue Bridge - Lighting Automation Engine
- 22 **Luc de Smet PDEng;** Impromptu: Reusable, Accessible, and Flexible Data Analysis
- 24 **Habtamu Aboma Tolera PDEng;** Model-based Testing Applied to Foreign Component Acceptance and Formal Components
- 26 **Tamir Tsedenjav PDEng;** CardioNotifier - A Smartphone-based Notification System for the Cardiologist
- 28 **Juan Villa PDEng;** Domain Specific Languages in Software Architecture
- 30 **Yi Xiao PDEng;** The Design and Implementation of a Large ZigBee Network Analyzer
- 32 **Ronald van Zon PDEng;** Towards a Modular Software Architecture for YieldStar's Pattern Recognition Functionality



Prof.dr. J.J. Lukkien

Automation and the Software Development Process

As the intricacies of dealing with large software and its development process become better understood, new opportunities arise to speed up parts of the development process and enhance its quality. Models have been used for a while now to generate functional code. This has improved the quality of the code base at the unit level.

A next step is the validation of the behavior of such code units when they interact with other units, either generated or handcrafted. These units as a rule are not all available at the same time. This is a common situation in a systems environment where hardware and software components are developed concurrently. In order to facilitate early testing the information captured in the models and in particular the interface descriptions can be used to generate simulators and tests that help to probe the available units in interaction with their environment. This year's batch of design projects show some of examples of these and other model-based approaches.

In 2016 we have also seen some other approaches for dealing with the complexity arising from large quantities of various kinds. Examples are the automatic linking of large data sets, the generation of cases for testing in combinatorically complex situations, and the development of tooling and languages to support the management of complexity.

You will find examples of these in this booklet. It again presents an overview of technological developments in the Brainport region and beyond, as well as the challenges that our trainees face in their graduation projects and resolve with high quality designs. Therefore I say to the trainees whose work is represented in this booklet: I congratulate you with the results achieved and I wish you a successful and satisfying career.

Johan Lukkien
Dean of Department Mathematics and Computer Science
Eindhoven University of Technology



“An ambitious project to generate optimized source code from a specification that takes all electrical constraints into account. Both technical as well as organizational challenges (team acceptance) played a role. Filip built a solid, high quality architecture and proved the concept, while carefully cooperating with the team. We now have a prototype to be extended with more functionality. We will build in all best practices, improving quality, test cost as well as test development cost.”

ir. Roger Habets MBA
Director Operations BL SMC
NXP Semiconductors

Challenges

The main challenge of the project was to perform reverse-engineering in order to define format and structure of the code inside the test files that the test engineers manually create. This was performed by conversations with the test engineers and analysis of the code. Additionally, the requirements for the design to have structure that will be extensible and reusable in the future, in order to adapt to new technologies and methods, increased the complexity of the project.

Results

The result was a prototype of a code generator that can be used to create test files containing code for testing amplifiers. The input of the code generator was an excel file containing parameters needed for the test files. The generated code was later compiled and executed on a test environment, used by the test engineers to execute and debug the code.

Benefits

The prototype of the code generator will help improve the development process of the NXP Mobile Audio Test Department in the future. First of all, the time it takes to develop the test files will be significantly reduced as the test files will be generated. Second of all, the test files themselves will have optimized code, which will reduce the execution time of the test files.

Filip Anastasovski PDEng

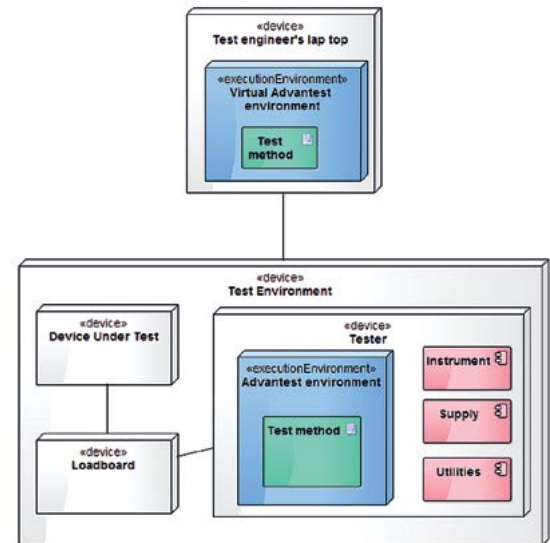
System for Code Generation for Testing Amplifiers at NXP Mobile Audio Department

The test engineering group at Mobile Audio Test Department at NXP is working on testing integrated circuits such as smart amplifiers for smart phones. Testing amplifiers for world known clients (Samsung, Sony, and Nokia) is a complicated and delicate task. That is why the quality of the amplifiers is of high importance. The goal of the project was to design and implement a solution for improving development of the tests, by reducing complexity on the software aspects of the test development. The main aspects that the solution should address were faster development time, less error inside code and less amount of code.

The solution was to automate the development process by generating the tests that the test engineers write manually in the current development approach. The automation will reduce the development time by abstracting the programming of the tests, reduce the errors created by human factors and will remove the redundant boilerplate code created by the test engineers. Additional time can be saved by automatically compiling and executing the generated code.

A code generator prototype was designed and developed as part of the project. The code generator receives test parameters defined by the test engineers as input and generates source files for testing the amplifiers. Even more, the generator compiles and executes the generated source files on the test environment. The prototype of the code generator proves that test methods can be fully generated by describing the tests in an abstract manner, and that can be of great importance to the test engineers.

deployment Deployment diagram



Challenges

The performance of an application in Spark, depends on many configuration parameters of Spark. In order to find an appropriate configuration for an application, several experiments need to be performed. These experiments are very expensive and time consuming. The main challenge of the project was to design a modular simulator for the scheduling in the Spark Framework. The simulator must abstract from the unnecessary details but provide enough functionality so that the clients can perform their experiments.

Results

The results of the project include an extensible design and a working prototype of the design. The prototype constructs a representation of physical machines, components of Spark, model of an application and the ability to simulate application scheduling in Spark. The prototype has been designed to easily add new scheduling policies and multiple cluster managers. The prototype was compared with the real Spark and the results were close to the real Spark.

Benefits

The simulator will give an insight on the important application scheduling issues in Spark. It will save time and money by providing the appropriate simulation behavior required for experiments. The simulator will also contribute to the existing research on resource scheduling. Researchers can use this simulator to perform experiments easily without the need to spend money on clusters or wait for the machines.



“Sarwan had to study the source code of Spark in detail, which is not an easy thing, given the lack of proper documentation. He has managed to do so independently and successfully. To conclude, we are looking forward to putting Sarwan’s simulator into use to further our research in cluster resource management.”

Dr. Rudolf Mak
Eindhoven University of Technology

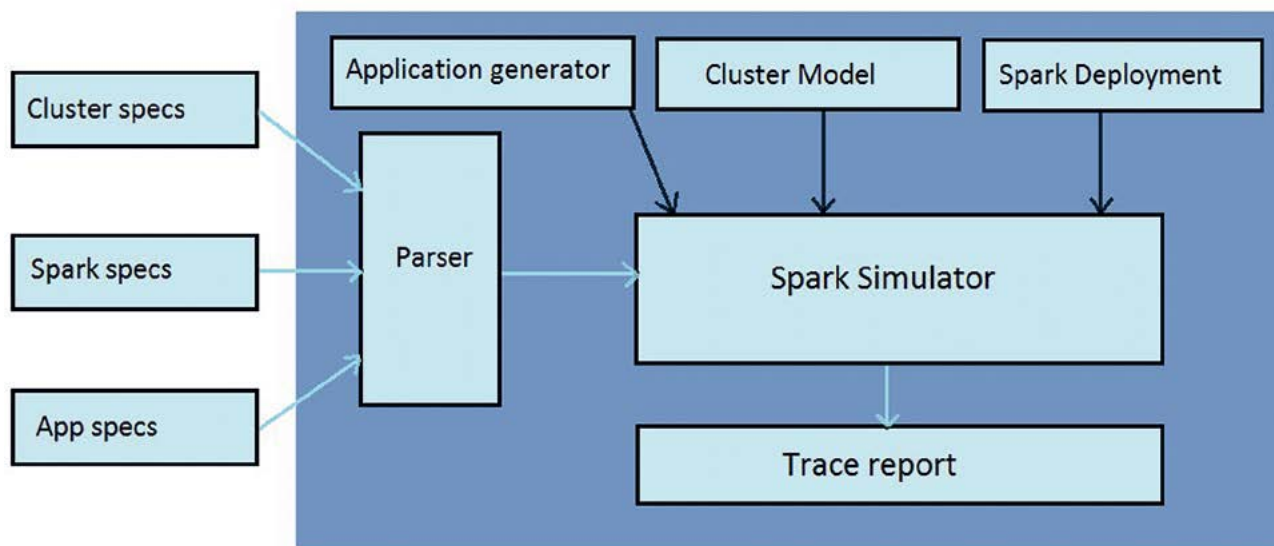
Sarwan Dar PDEng

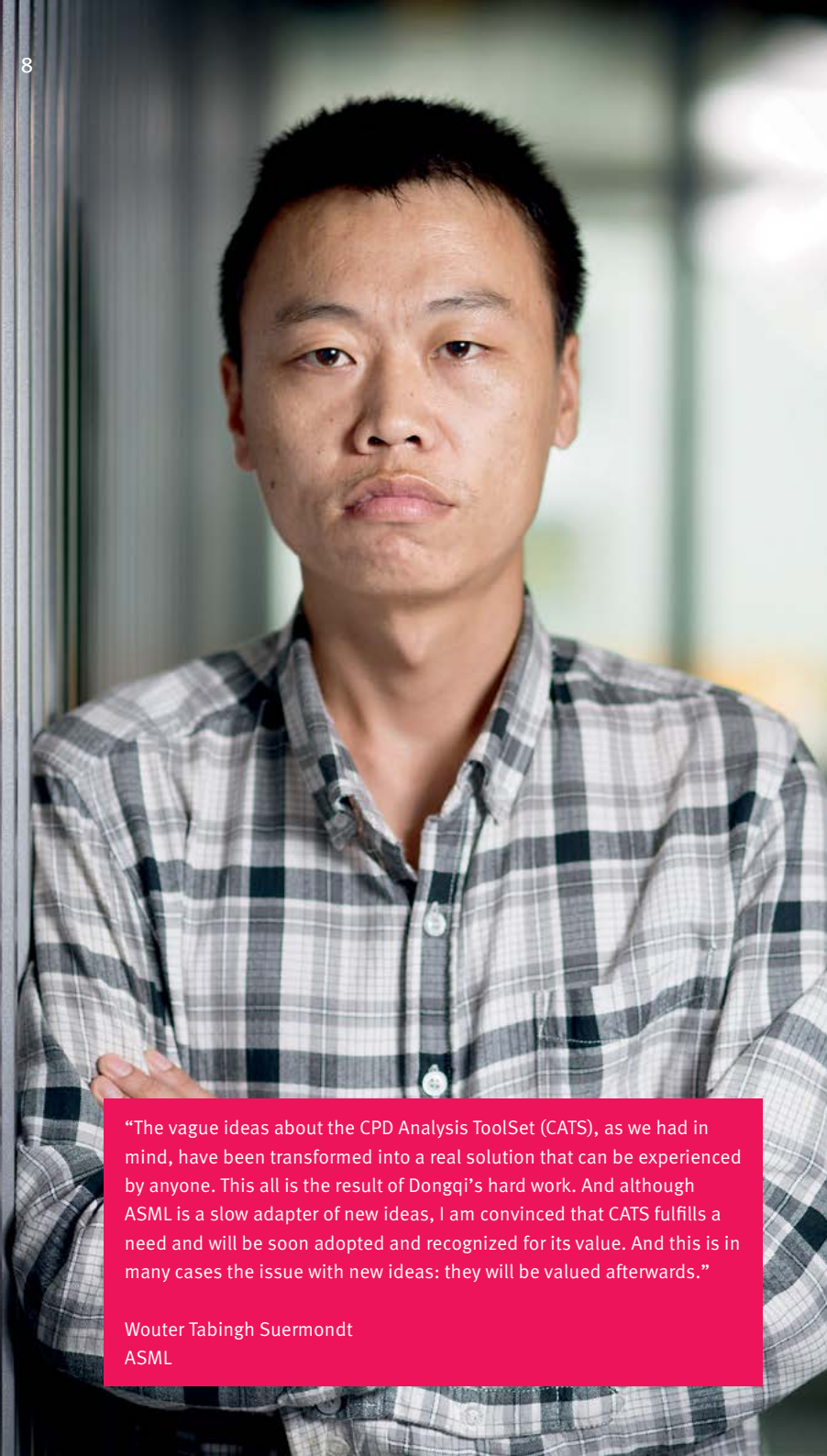
A Simulator for Spark

The advent of social media and the advancement in technology has contributed significantly to data production. The amount of data produced by applications such as Facebook and WhatsApp is growing every day. This large amounts of data is known as 'Big data'. This data is analyzed to find hidden patterns, get useful information and sometimes even to predict the future. Spark is one of the most popular big data analytics framework that helps in analyzing big data.

Spark has a lot of different configurations and using the appropriate configuration results in a good turn-around time of applications and efficient use of resources. Therefore, finding the best configurations for applications is an important research topic. However, performing experiments on real Spark is expensive and time consuming.

The project focused on developing a simulator for Spark scheduler. The simulator allows to perform experiments that are expensive and time consuming for the real Spark. The Spark simulator constructs a representation of the physical machines, components of Spark such as Master, Workers, and Applications and simulates the application scheduling in standalone mode. The simulator was compared with the real Spark and the results look quite promising.





"The vague ideas about the CPD Analysis ToolSet (CATS), as we had in mind, have been transformed into a real solution that can be experienced by anyone. This all is the result of Dongqi's hard work. And although ASML is a slow adapter of new ideas, I am convinced that CATS fulfills a need and will be soon adopted and recognized for its value. And this is in many cases the issue with new ideas: they will be valued afterwards."

Wouter Tabingh Suermondt
ASML

Challenges

The challenge in this project was to study the domain of CPD (Calibration, Performance and Diagnostic) applications and software archive in ASML. To acquire the necessary knowledge I needed to identify and consult many people who could provide it, including my company supervisor, CPD competence team and some colleagues.

Results

There are three main results:

1. collect an initial list of CPD properties, which can be used to analyze the CPD applications,
2. design and implement the toolset that is able to get the CPD properties automatically, and
3. use the toolset to perform analysis on the population of CPD applications.

Benefits

The toolset provides an automated mean to gain insight in CPD applications. By using the toolset, the stakeholders can easily trace the development and modification of CPD applications. It can also benefit the CIDT (CPD Integrated Development Toolkit) project by providing the current CPD landscape and an initial analysis of interface stability.

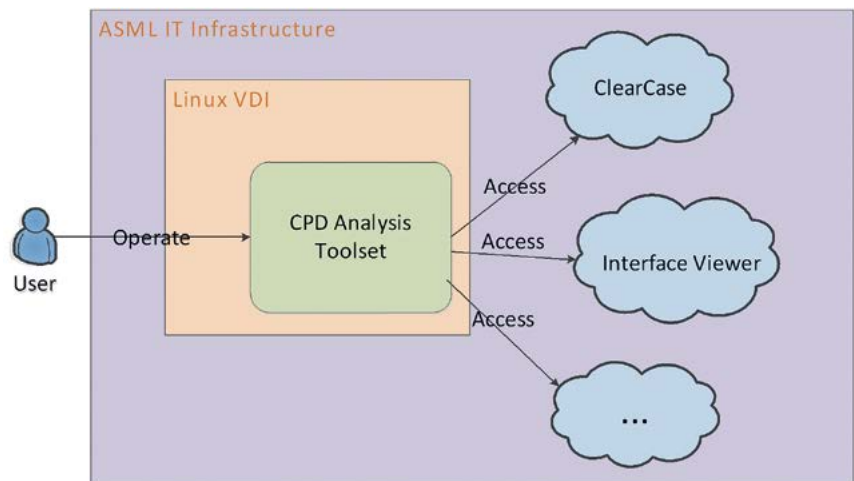
Dongqi Hu PDEng

Analysis Toolset for Calibration, Performance and Diagnostic (CPD) Applications

CPD applications are a group of test applications in ASML aimed to perform calibration, diagnostics and performance measurement on the machine. In order to optimize the development process of CPD applications, a new toolkit CIDD (CPD Integrated Development Toolkit) is under development. Substantial knowledge of CPD applications is needed for the development of CIDD. ASML has developed about 1500 unique CPD applications and around 1000 of them are still in use today. Moreover, most of them are still modified to varying degrees and with varying frequency. A toolset is needed to perform analysis on the CPD applications automatically.

This project is about the development of a configurable toolset that can get the selected properties of CPD applications. The toolset consists of 1) a database that stores the processed data, 2) a front-end that enables the user to analyze the CPD data by means of SQL queries, and 3) a converter that converts raw CPD data into a CPD-data model and stores it in the database. The toolset is designed to be extendible that new properties can be obtained by adding new adapters, which are designed to deal with the diversity of data source. A prototype is implemented and several adapters are implemented to demonstrate the capability and extensibility of the toolset.

The toolset was transferred to the stakeholders in ASML and an adapter has been successfully implemented by the stakeholders.



Challenges

The main challenge of the project was to model and develop a system that contains all algorithms necessary for bringing data together for a same patient, collected from various sources that is customizable to the needs and requirements of Philips data linkage use cases. In order to do so, understanding the current usage, challenges and needs of the data linkage in Philips across the health continuum was of great importance.

Results

The main result of this project is a developed data linkage system that offers a set of data linkage routines for coupling data sets imported by flat files. Using a suite of deterministic and probabilistic data linkage algorithms, the system is capable of bringing together records that contain uniquely identifiable information if present, otherwise less uniquely identifiable information or user-selected identifiers. The system is robust for missing and erroneous data due to administrative errors, coding or misspelling, it is easily adjustable to the end user purpose and it provides evaluation of the data linkage process. The system is deployable on the HSDP, hence the data linkage routines are available for everyone in Philips.

Benefits

Using the data linkage system, data of individuals from various sources can be brought together and new insights can be discovered that were never learned when data stayed in silos. Linking health data of individuals from different sources can be extremely insightful; more information can be pulled from linked data than from every data source separately. Using the data linkage system, the performance of the risk prediction models that are developed by the researchers can be improved as individual data and models can be combined.



“Ana Kostadinovska immersed herself in the field of data linkage and revealed the corresponding needs and challenges for Philips in its innovation drive across the health continuum. She has convincingly demonstrated proof of delivery by providing us a package entailing all algorithms required to bring data together for the same patient from various sources on the Philips HealthSuite Digital Platform.”

Prof. dr. Steffen Pauws PDEng
Philips Research

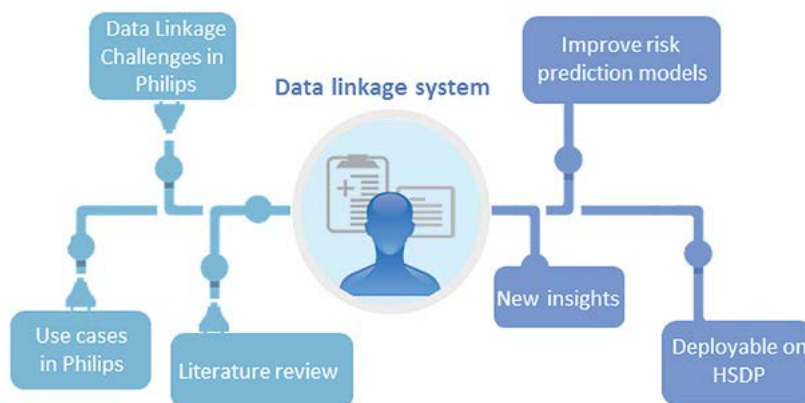
Ana Kostadinovska PDEng

Data Linkage Architecture for Population Health

Linking data of individuals from various sources can be very insightful. Linked data can provide more information than the information contained in every data source separately. Research scientists in Philips are trying to perform data linkage in order to combine data of patients from different sources. By linking data from different sources, the researchers aim to improve the prediction accuracy of the developed prediction or risk models as well as perform analysis with access to complete data.

Various use cases of data linkage reside within Philips Research but no unified approach in data linkage exists. Every project develops its own solution that is mainly used in the project only. The data linkage cannot be performed by the researchers since they are not allowed to access identifiable information of patients. Hence, developing a solution requires laborious manual effort since it requires technological skills which might not be present with the people from business units, who have access to the data and are performing the data linkage. The goal of this project was to deliver a system that provides the data linkage functionality and is a solution for the challenges and needs of the Philips data linkage use cases. In order to be used as a data linkage solution in various projects within Philips, the system should be accessible for everyone in Philips. Being deployable on the Philips HealthSuite Digital Platform (HSDP) was another goal of the project as a way to provide the data linkage system to the end users.

A data linkage system was developed that provides the data linkage functionality and can be used in every Philips data linkage use case. The data linkage system is compliant with the HSDP, which makes it accessible for every project in Philips that has a need of data linkage.





“Aldo has created a language for modelling interfaces, capturing both their semantics and behavior. Further, he created language extensions targeting the specifics of testing and wrapper generation. With the increased productivity and increased test coverage it provides, we believe it will quickly be adopted by our teams, which will be able to spend more time on features and to guarantee the quality of their interfaces.”

Dr. Andrei Rădulescu
Thermo Fisher Scientific (Formerly FEI)

Challenges

The model-based technology was the first challenge, selecting appropriate technology that matches the initial requirements, but offering enough infrastructure and extensibility for the future. The second challenge was building a simple interface modeling language based on the existing legacy code. Finally, the main challenge was the development of an automation process able to produce a complete test suite for ensuring the interface behavior described by the modeling language.

Results

The project produced is an extendable framework featuring a modeling language, a test case generator, and a wrapper generator. The language models the interface syntax and semantics and serves as interface documentation. Generators used these models to automate the creation of specific artifacts (test cases and wrappers). The framework is designed to guarantee the separation of concerns and to enforce the extensibility for future features.

Benefits

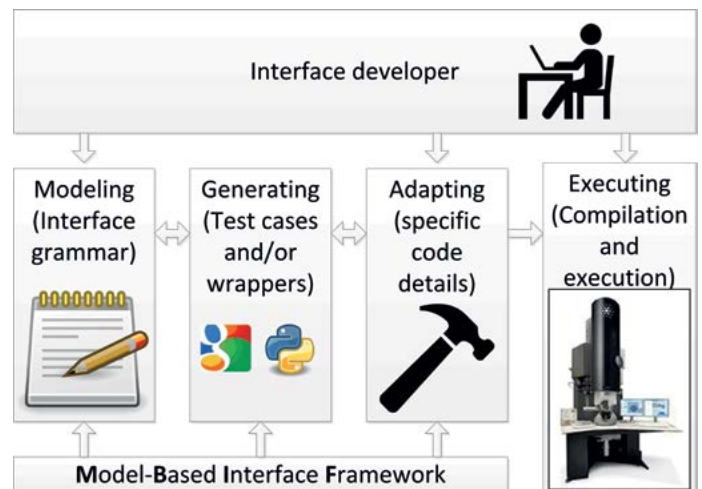
As an automation project the solution helps FEI by removing repetitive and tedious activities while guaranteeing the quality of generated artifacts from those activities. Avoiding these activities developers will be able to spend more time in productive activities, such as, developing new features. Finally, having a modular and extendable architecture permits the inclusion of new automation features and simplifies future maintenance.

Aldo Daniel Martínez Márquez PDEng

MBIF: An Extensible Framework for Automatic Generation of Test Cases and Wrappers

Repetitive activities tend to be tedious, mechanical, and error prone. Additionally, these activities consume significant time, decreasing development efficiency. Automation and model-based technologies are a solution for decreasing development time and ensuring the quality of the artifacts resulted from these activities. The Model-Based Interface Framework (MBIF) uses interfaces models for automating test and wrapper generation.

A modeling language has been designed to capture the main elements of software interfaces, including/ describing their behavior through state machines. The language is descriptive, simple, and concise, and can thus also serve as interface documentation. These models are the input for the artifact generators which are the components responsible for creating specific artifacts. The test case generator uses an algorithm that creates an interface test suite based on the state machine behavior (states and transitions). By design, the algorithm has a full coverage of the modeled behavior. The wrapper generators automate technology translations, but also other interface features, such as, threading and exception handling.



The Model-Based Interface Framework is extendable, easy to use, and flexible. This is achieved by using a component-based architecture with a modular separation of concerns keeping a clear distinction between the modeling language and the code generators. The MBIF is a first step for creating a FEI automation framework. Currently, it covers testing and wrapping. However, in the future, it can be easily extended to cover additional features.

Challenges

The main challenge of this project was to design a methodological approach to performance analysis that can be used as part of the testing process. It has to be flexible to adapt its implementation to a complex data warehouse. Another challenge was to analyze the complex data warehouse selected, which was in constant change since it was continuously being enhanced.

Results

The results are the design of a framework and its prototype implementation that supports the structured approach for performance analysis of Rabobank's data warehouse. The prototype provides performance information that can be visualized and interpreted in an interactive way. This information is used to explore and understand the throughput and behavior of the data warehouse analyzed.

Benefits

The prototype proves that the framework designed can be used to provide new insights on the system's performance and to identify possible bottlenecks. Additionally, the structured approach to performance analysis is a step towards the standardization in Rabobank of executing performance testing of complex systems.



"Vladimir's work already led to identifying some of the bottlenecks, as well as the technical debt in one of our live data warehouses whilst he was building the framework. With those findings, Vladimir already demonstrated the benefits of a thorough, structured approach to performance analysis and testing."

Drs. Jurgen Krosse
Rabobank

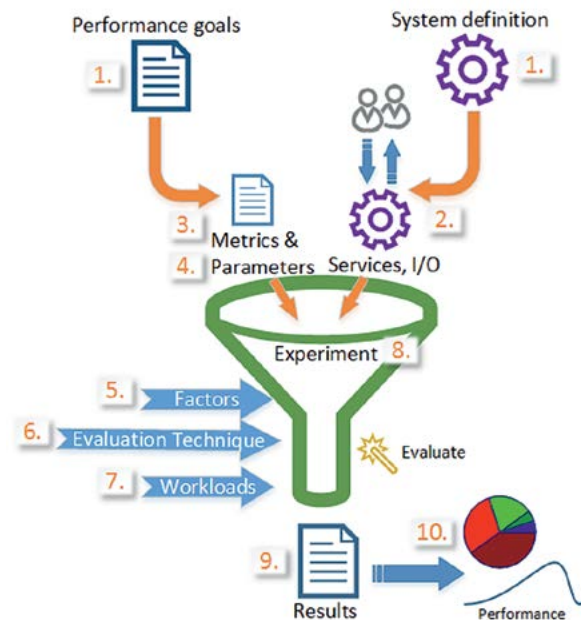
Vladimir Mikovski Iotov PDEng

Performance Analysis Framework for a Complex Financial Data Warehouse

Rabobank is a multinational banking and financial services company and a global leader in agro-financing and sustainability-oriented banking. As a consequence of the international financial crisis in 2007-08, the requirements for liquidity risk management have increased and the existing Rabobank's solution needs to be upgraded and extended. The main focus of this project is the data warehouse used for liquidity risk data aggregation in Rabobank.

Currently, the data warehouse is being enhanced by Rabobank's engineers to extend and improve its capabilities. These improvements could be measured with a continuous analysis on its performance. Furthermore, this performance evaluation should become part of the testing of the system before its release to production. The goal of this project is to apply a structured and methodological approach to performance analysis that can be adopted in the testing phase.

A solution as a framework was designed and implemented to support the performance analysis of the data warehouse. The design of the performance analysis framework emphasizes the usability and genericity attributes of the solution. An implemented prototype demonstrated the benefits of using the framework as part of the testing procedure, bringing valuable new insights into the system's performance and its behavior.





"Irina designed a prototype of a simulator development environment which enables us to create basic simulators with minimal effort. We were pleasantly surprised by the ease with which she absorbed the relevant knowledge; it took her a few weeks to learn what a typical new ASML software employee takes a few months. I admire her ability to adjust to the changes in requirements, which is very typical for a dynamic ASML project, and to align the different interests of the core team members she interacted with."

Ing. P.C. Koper, BSc
ASML

Challenges

One of the main challenges of the current project was to align interests, knowledge, and priorities of stakeholders from different departments and having different experience in simulation. Another challenge was to find an approach for modeling simulator structure and behavior that is suitable for both software and functional developers.

Results

As the result, the generic approach for designing simulators was defined. It consists of four modules that allow configuring a simulator, modeling its structure and behavior, deploying Matlab models directly in the simulator design, and generating code from this design. Together, these modules form the Asml Simulator development EnviRonment called ASTER.

Benefits

ASTER allows simplifying the simulator design and integration process by enforcing the unified simulator architecture and reducing the engineering effort via code generation. ASTER provides the simulator modeling module that raises the abstraction of high-level languages by introducing the simulator domain specific modeling engine. Therefore, a simulator can be designed by both software and functional developers familiar with the simulator domain.

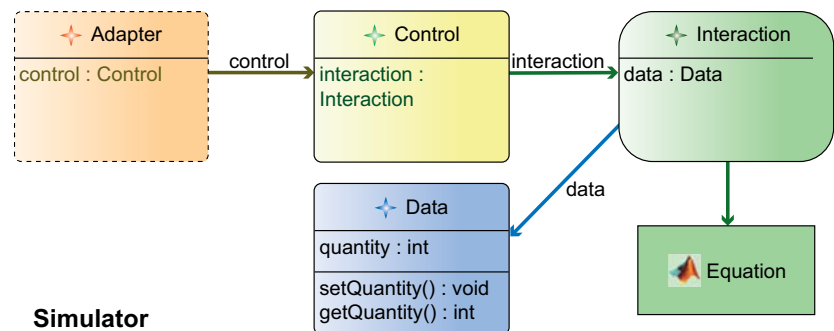
Irina Nikeshkina PDEng

ASML Simulation Development Environment

ASML is the world leading provider of photolithography machines for the semiconductor industry. Lithography is a complex process that requires a reliable control system. The biggest challenge of designing such a system is the realization of an effective and efficient software and hardware testing process. When the testing involves electronic or mechanical equipment, the process becomes more expensive and time consuming due to low availability of this hardware compared to testing on pure software platforms. However, integration and system testing can hardly verify software quality without this hardware involved. Therefore, hardware simulation approaches are widely used in ASML for testing purposes.

A number of simulators have already been designed for different subsystems of the machine, in different ASML departments, by different developers, implemented using different programming languages, and as a consequence with different architectures. Such simulator variability exists due to a lack of clear guidelines that enforce a unified architecture and guard the process of a simulator design. Moreover, the knowledge of the simulated hardware is generally outside of the software domain. At the same time, each simulator design involves repetitive work implemented manually. Together, these make the modelling, design and integration of simulators a complex process.

To solve the described problem the following project goal was established:
Design a framework for creating simulators that enforces unified simulator architecture, reduces the engineering effort by code generation, and can be used by both software and functional engineers for simulator design.



Challenges

Accessing in-vehicle data is one of the challenges that needs to be solved when developing applications ranging from a smartphone app that keeps track of fuel or electricity usage, load prediction of the electricity grid, traffic-jam and weather-forecast apps, pothole detection, and co-operative driving applications. Currently, data access solutions are application specific, which results in a solution that is not easily sharable between applications.

Data users want to access new sensors that are installed in future vehicles, so it should be easy to integrate new sensors. Data owners want transparent and easy configurable data sharing setting implemented by security and privacy mechanisms that prevent unauthorized data access. Most stakeholders benefit from a plugin mentality that enables installing, updating, and removing third party functionality. The challenge of this project is to define an independent framework that offers a generic data access strategy that is satisfactory for all involved stakeholders.

Results

The project starts with the observation that current ways of accessing data in cars is insufficient. Vehicles contain more in-vehicle data that is accessible by means of standardized interfaces such as the OBD2 (On-Board Diagnostics, version two) standards. A stakeholder analysis shows the different concerns of stakeholders that are involved in a data access strategy. Furthermore, a framework design is proposed that addresses data access, data preprocessing, and enables storing and using of this data by third party databases and applications. An Android based prototype is used to test and demonstrate the feasibility of those features on a smartphone.

Benefits

The project resulted in a framework proposal that enables independent development, by third party developers, of apps that require access to in-vehicle data. Developers do not directly depend on agreements with manufacturers, but can use a standard interface to access in-vehicle data instead. The result is an extendable, flexible, and cost-effective solution that allows implantation of different privacy schemas.



“Automotive is a domain with a complex interplay of stakeholders. The system and software architecture of the respective systems reflects this complexity. Jeroen’s project represents a piece of the puzzle that brings together the ongoing standardization in automotive by IEEE and ETSI and the world of smartphone applications and quick innovations.”

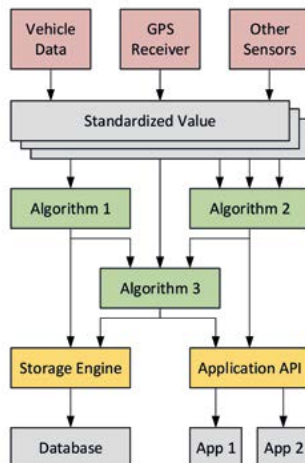
Prof.dr. J.J. Lukkien
Eindhoven University of Technology

Jeroen Redegeld PDEng

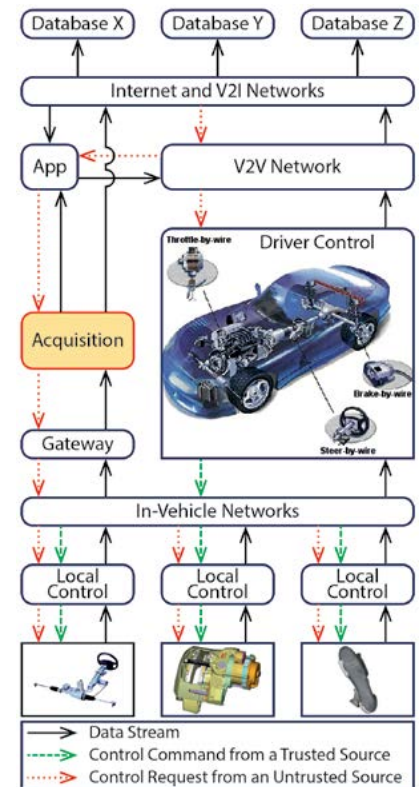
A Framework to Stimulate Innovation Based on Automotive Data Collection


Waterways and Public Works Agency, or Rijkswaterstaat (RWS) in Dutch, is a governmental agency that is responsible for the design, construction, management, and maintenance of the main infrastructure facilities in the Netherlands. Main infrastructure facilities in the Netherlands consist of the main road systems, the main waterway network, and the main water systems. The focus of this project, as suggested by the title, is on the main road systems.

RWS uses real-time data to inform and guide road users in an optimal way to improve traffic flow, lower traffic jams, reduce emissions of harmful substances, and improve safety. Furthermore, data that reflect road conditions is required to plan road maintenance and winter services. Innovations in Intelligent Transport Systems (ITS) – the usage on Information and Communication Technologies (ICT) in the field of mobility – introduces alternative data sources that can be used by RWS for their tasks.



A trend in the automotive industry is that complexity of software based in-vehicle systems increases. Modern cars contain up to 70 to 100 Electronic Control Units (ECUs), tens of millions of lines of code in vehicles, an increase in the number of sensors installed in vehicles, and at least 5 different in-vehicle networks to exchange data between ECUs. As a result, more data about vehicles and their surroundings becomes available. Accessing this data enables the development of new, innovative applications. This project focuses on a generic in-vehicle data access framework that can be used by third party developers, including RWS, to gain access to in-vehicle data.





“Spyridon stepped into the emerging world of IoT which is on a journey discovering where it can offer the best user added value. With endless questions he provided an overview of requirements and techniques of lighting automation engines, as well as a proposed direction. This will be used in determining the next steps on our journey.”

W.Slegers
Philips Lighting

Challenges

Working on a brownfield project with limited domain knowledge was the biggest challenge of all. At the beginning, embedded world seemed like uncharted territory. Consequently, extra effort was needed at early stages in order to acquire the sufficient level of understanding. Additionally, the learning curve was steep because the project had a lot of dependencies on a complex, existing system.

Results

First, through a thorough investigation of existing open source automation engines good practices and invaluable design insights were extracted. These findings in combination with elicited requirements led to the high level design of a scripting engine. Second, abstraction layers were created on top of the proposed engine. These layers describe ways to expose configurable metadata to third-party applications hiding the complexity of the underlying engine.

Benefits

Philips Lighting is now able to automatically describe valuable information to external developers in order to create apps interfacing with Hue and broaden the openness of the system. Hue is also enabled to easily update and extend the new specification, two features inherently supported by the technology choice. Last but not least, the project suggests a direction on how Philips Hue system can improve its current lighting automation based on good practices widely used in other similar systems.

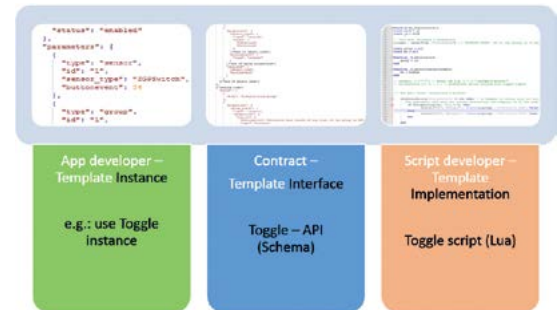
Spyridon Skoumpakis PDEng

Hue Bridge – Lighting Automation Engine

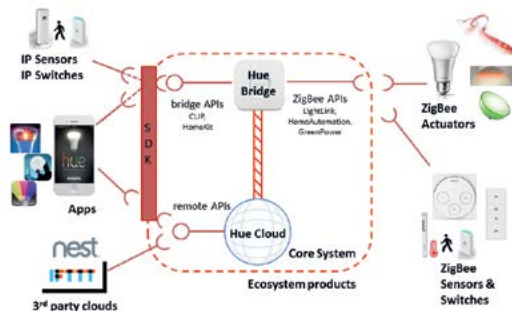
Investigation of lighting control alternatives for the existing rule-engine

Philips Hue is an internet connected lighting system designed to transform how users experience light inside their homes. It is one of the leading and most installed IoT products in the world. Philips Hue enables color tunable lights to be controlled from smartphones, web services or other control logic and devices running in the system.

The brain of Hue is an embedded device called Hue bridge. It controls and monitors ZigBee lights, sensors, and switches. Moreover, it uses a rule engine, which receives switch, sensor, or timer triggers and then sets a specific lighting scene as result. This engine is a software module responsible for the automation logic of the bridge.



Improving home lighting experience is of great importance to Philips Hue. With an increasing complexity of home lighting control use cases the need of exploring more sophisticated automation engines is imperative. In the context of this project, an investigation of alternative engines is conducted having a comparison table as main output. Furthermore, a formal



specification for a future Lighting Automation Engine is developed based on the findings of the alternatives research. The specification describes ways to transcend the limitations of the existing engine introducing the power of abstraction layers and scripting. The abstraction layers expose important information to different user categories and at the same time hide the unnecessary complexity of the underlying scripting proposal. A proposal that allows more flexibility to handle the needs of the emerging IoT world while preventing a loss of control. All in all, this project gives a direction for Philips Hue on how to improve and proceed with its current lighting automation.

Challenges

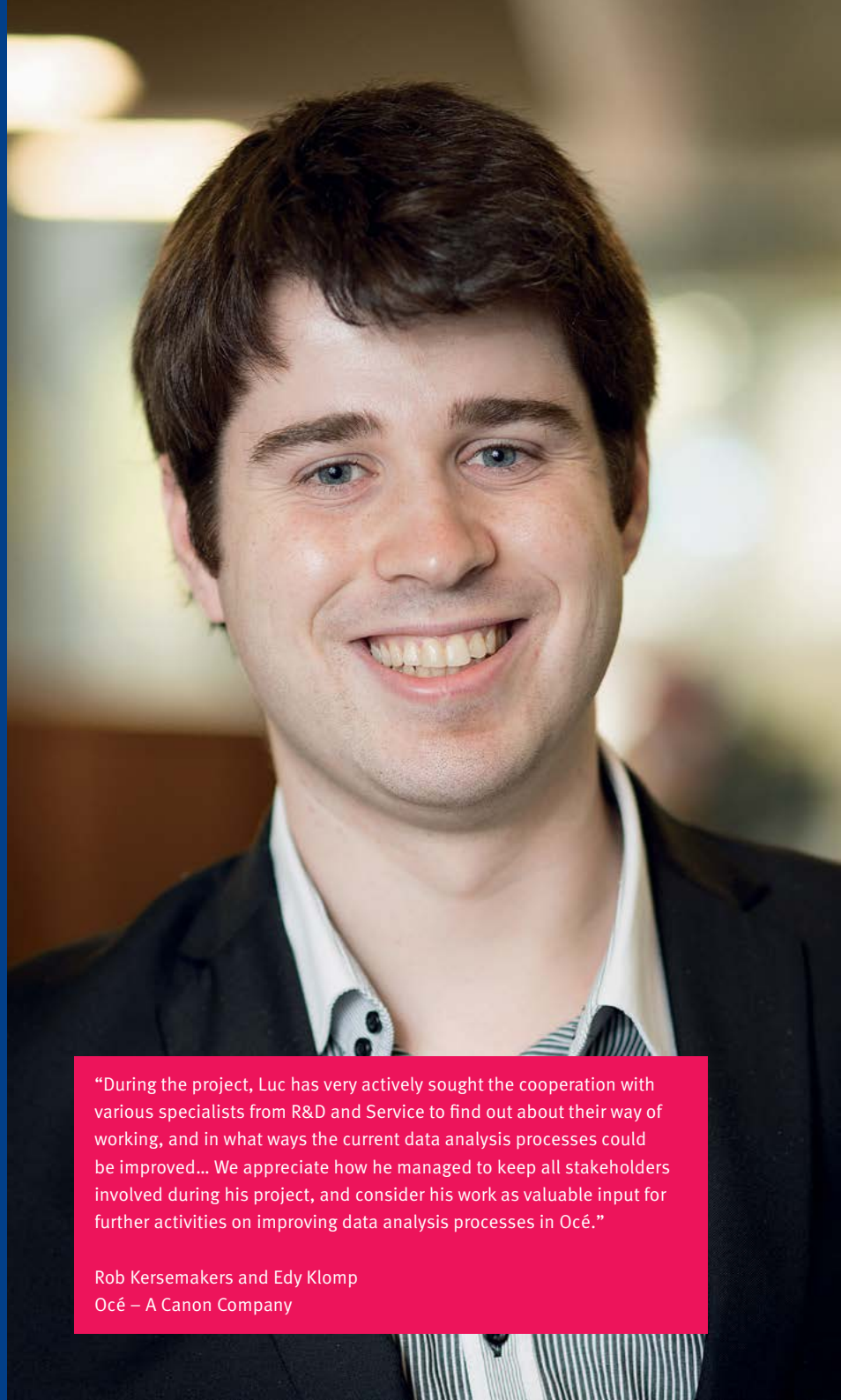
The main challenge of this project was to offer data analysis techniques in such a way that the average machine module designer at the company can use and adapt these techniques. This challenge puts restrictions on the result in terms of ease of use and flexibility for specific domains. In addition, it was challenging to scope towards these stakeholders given the broad project definition.

Results

The result is Impromptu: A framework that allows creating, sharing, and adapting generic data analysis scripts. These generic scripts can easily be applied in domain-specific scripts. In addition, the framework was populated with a number of plugins. These plugins concern preprocessing, anomaly detection, correlation analysis, and visualization. Finally, Impromptu offers a prototype for easily storing and viewing data-related annotations to allow sharing of domain knowledge and insight gained through analysis.

Benefits

Impromptu offers designers a more effective way of analysis by 1) Making it easier to reuse analysis scripts, 2) Enabling anomaly detection and correlation analysis, and 3) Storing domain knowledge and insight to be retrieved later. This saves time for the designer performing analysis, money because of more effective service visits, and time until product release. These Improvements should be quantified to gauge the true benefit.



“During the project, Luc has very actively sought the cooperation with various specialists from R&D and Service to find out about their way of working, and in what ways the current data analysis processes could be improved... We appreciate how he managed to keep all stakeholders involved during his project, and consider his work as valuable input for further activities on improving data analysis processes in Océ.”

Rob Kersemakers and Edy Klomp
Océ – A Canon Company

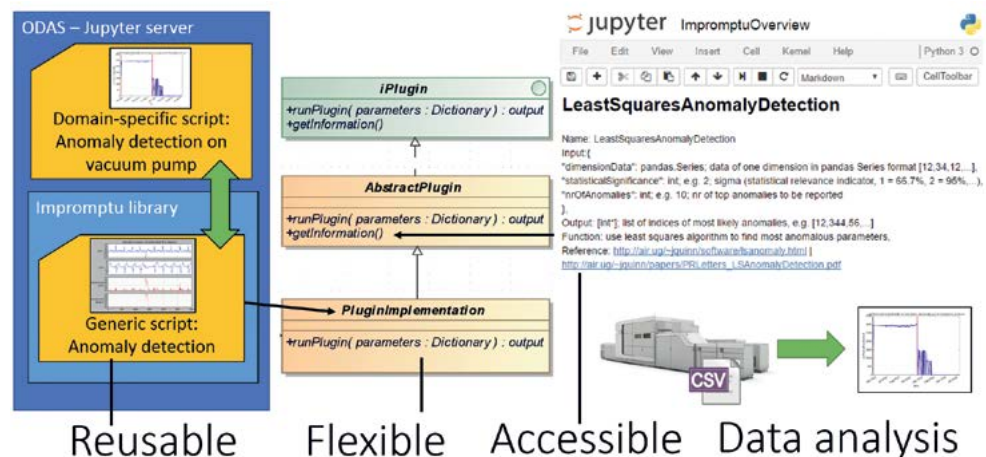
Luc de Smet PDEng


Impromptu: Reusable, Accessible, and Flexible Data Analysis

Océ – A Canon Company creates industrial printers which are capital goods for their customers. To validate and verify product design, these complex machines are released to early adopters in early stage of development. In this stage, it is important to learn from the machines in the field and further improve machine design and service strategies. The goal of this project is to improve learning from machines in the field by providing a set of reusable, flexible, and accessible data analysis tools for machine behavior logs.

Machine behavior is extracted from sensor values logs for particular machine modules. In general, machine module designers are responsible for creating the logging software and using the resulting logs to analyze machine behavior. In addition, analysis is part of the incident handling for product service specialists. An opportunity existed to improve the current toolset these stakeholders use for analysis: To make analysis scripts reusable, while keeping the flexibility and the accessibility of current tools, such as Excel, Matlab, or the Diagnostic Framework (an internally developed tool).

Impromptu is a Python-based framework that implements these requirements by using a plugin-based architecture and integrating it in a data analysis platform under development, called ODAS. Impromptu allows access to plugins through a standardized interface and automatically updates an overview of all available scripts. In addition, Impromptu was populated with plugins for data preprocessing, anomaly detection, correlation analysis, and visualization. Stakeholders recognize the value of Impromptu, which will be integrated into ODAS.



A portrait of a man with dark, curly hair and a light beard, wearing a light blue button-down shirt. He is looking directly at the camera with a slight smile. The background is a blurred indoor setting.

“When the software for a machine is developed by more than a thousand programmers, testing of that software requires an elaborate toolchain. It starts with the development of component models for the different parts of that software, at the right abstraction level. From these models, a set of abstract tests may be automatically be derived, tailored to the type of errors a test-engineer is looking for. That set of abstract tests has then to be converted to tests that can actually be executed on a real, or partial, implementation of the software in a reasonable amount of time. And preferably, as a final step, the execution of those tests needs to be automatized, and their interpretation communicated back in an efficient manner. The challenge for Habtamu was to make the automated testing solutions proposed by academia and spin-off companies practical.”

dr.ir. P.J.L. Cuijpers
Eindhoven University of Technology

Challenges

One of the main challenges of the project is to select a model-based testing tool, from the many available, that best suits the requirements of the stakeholders. The second challenge is to generate test cases from non-deterministic models, which is not supported by most model-based testing tools.

Results

The project resulted in steps and guidelines to apply Model-based testing to software components, both foreign and formal components. The concrete designs that follow from these guidelines demonstrate the feasibility of the approach. The approach can be integrated with the software development approach in ASML. Using the prototype implementation of the design, test cases were generated based on different model characteristics with little manual effort.

Benefits

Applying the described steps and guidelines of Model-based testing to software components leads to improved software development process. The result is reliable software with a reduced development cycle time.

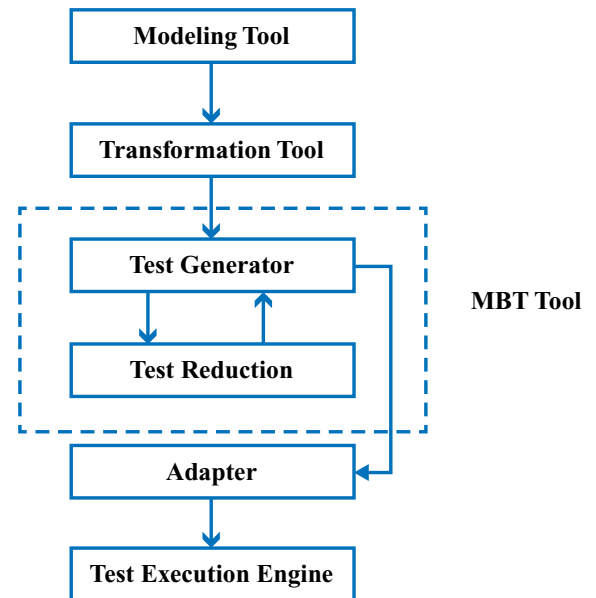
Habtamu Aboma Tolera PDEng

Model-based Testing Applied to Foreign Component Acceptance and Formal Components

ASML is the world's leading provider of lithography systems for the semiconductor industry. These systems are critical for the production of integrated circuits or chips. The lithography machines require large and complex software systems running on it. Because of the size and complexity of the software, ensuring software quality (absence of defects, on time delivery, reliability and meeting expectations) is a significant challenge.

Current testing method rely on manual process for test development. Because of this, testing software component takes long time. By applying model-based testing to software components, test development and software development time can be reduced.

Model-based testing was demonstrated by generating test cases from the interface models of the software components. Interfaces of the software components, which are modeled for development purpose, are used for test generation. From those interface models, test cases are generated with Model-based testing tools. Using the existing interface models reduces test development time and improve test quality by generating as many test cases as needed based on model characteristics.



Challenges

The challenge in this project was to investigate the possibility to support the cardiologist's workflow propose with help of a smartphone application, as well as to discover which functionality. In addition to that, the challenge was not familiar domain, medical cardiology and organizing enough discussions with a clinical expert is needed.

Results

As a result of the project, the CardioNotifier, a smartphone-based notification system with integrated FHIR server for the cardiologist was implemented and demonstrated as a support of the concept. For this, a research about cardiologist's workflow was conducted to find out how smartphone can support cardiologist's daily work. Also, various technical investigations took place to achieve most optimum design solutions.

Benefits

The prototype proves that smartphone-based notification system can be used to provide feedback that will directly or indirectly improve. In this project we selected FHIR as a standard to work with, since it is an emerging and modern standard, endorsed by HL7. Therefore, the project also covered the detailed investigation and possibilities of FHIR standard usage in mobile application solution.



"Starting from essentially a one-sentence project proposal 'how can smartphones be used to support cardiologists in their work?', Tamir worked with domain experts to understand and identify the most suitable use cases. He then created a series of prototypes, each with increased functionality. He leaves us with a demonstrator to approach the end-users and gain feedback to further develop and refine the application."

Dr. Charles Sio
Philips Research, Professional Healthcare

Tamir Tsedenjav PDEng

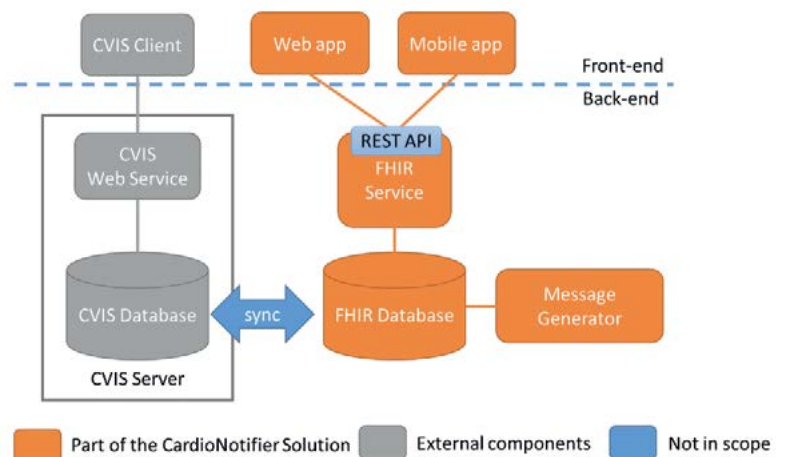
CardioNotifier

A Smartphone-based Notification System for the Cardiologist

Philips provides cardiology application solutions in the Cardiovascular Information Management System (CVIS) domain, such as Xcelera and IntelliSpace CardioVascular (ISCV).

More recently, with the introduction of ISCV, Philips is shifting towards web-based solutions that can be used anytime and anywhere. However, without a notification system a cardiologist is required to stay close to workstation to not miss any important information update. Staying at one place is not possible for a cardiologist: they have to be at many places, including those where there is no access to the workstations, throughout the day. All hospitals have efficient systems implemented for relaying emergency information, but for sub-emergency information a balance must be found between speed of relaying updates and interruption of daily activities. Because of that, crucial time may be lost by not having fast access to important information at any location in the hospital. Therefore, a mobile solution that can notify the cardiologist would be very helpful.

As a solution for above problem, in the CardioNotifier project, a smartphone-based notification system for the cardiologist, was developed and demonstrated. For this, a research about cardiologist's workflow was conducted to find out how a smartphone can support the cardiologist's daily work. Also, various technical investigations were done to achieve most optimal design solutions.





“Juan demonstrated various interesting technologies that are worthwhile investigating in our Model Driven Engineering projects. The editor for the Domain Data Services he realized will improve the acceptance of this technology within ASML.”

Wilbert Alberts
Software Architect
ASML

Challenges

Introducing Model-driven engineering in an organization is a difficult enterprise. Aligning the expectations from metrologists, developers and software architects, and explaining the potential benefits from using a DSL proved to be the most difficult challenge.

Results

An extension to ASML’s model-driven environment, which supports developers in implementing applications in line with ASML’s reference architecture. The approach followed has been documented generically and can be reused in different scenarios to extend a language workbench.

Benefits

ASML metrology now has a proof of concept that estimates the amount of effort required to extend ASOME with a DSL to express domain-data services. On a more general note, this project has brought to ASML the knowledge needed to evolve DSLs that only express structure to more complex ones that integrate structure with behavior.

Juan Villa PDEng

Domain Specific Languages in Software Architecture

In 2011, the metrology department at ASML started a pilot project to improve the designs of software modules that were to be refactored. The result was DCA, an in-house architectural style (inspired by the design methodology known as Domain Driven Design DDD and the Onion-layered architectural pattern) composed of three aspects -Data, Control and Actions-.

Fast forward to 2016 and the *software architecture group* is now building ASOME, a model-driven environment for supporting metrologists and developers in implementing systems in line with the DCA architecture pattern. ASOME provides developers with standard development environments, i.e., textual, graphical and combined editors, (code) generators, pathways to verifiers, analyzers and simulators among others, to best fit the development of components on each layer of the DCA architecture.

This report evaluates the feasibility of extending ASOME's domain specific languages with a new language. The goal: support developers and metrologists in implementing data-domain services for the data aspect in DCA. A domain model in the context of DCA captures metrology concepts, their attributes, relationships, as well as domain constraints and invariants. Domain-Data Services (DDS) enrich domain models with operations when an operation is not the natural responsibility of an entity.



Challenges


The challenge in this project was to provide a ZigBee analyzer with easy-to-use and reliable features that help to identify the problems quickly. Meantime, the design of the analyzer should be understandable for Hue light engineers and testers, which means the implementation should be extendable and maintainable. Besides, the setup of a large ZigBee network for the verification of the analyzer was time-consuming and demanding.

Results

A large ZigBee network analyzer was built up for this project. It enables visualizing the topology and data transmission of a large ZigBee network, which helps to investigate the performance and reliability of Hue devices. Besides, a large ZigBee network was set up and a preliminary analysis of the large ZigBee network performance was given based on the network.

Benefits

This project designed and implemented a custom fit large ZigBee network analyzer. This analyzer provides a solution to continue measurements for large ZigBee networks. Besides, it can be used for the envisioned target of enabling easy analysis, debugging and helping to create insights in the performance of large ZigBee networks.

A portrait of a man with short dark hair, wearing a blue and white plaid button-down shirt over a white t-shirt. He is smiling slightly and looking towards the camera. The background is a blurred indoor setting with warm lighting.

“Yi Xiao very quickly mastered the intricacies of the ZigBee network and our system, was working independently and needing very little guidance. He performed well and surprised us with his abilities to very quickly create the initial prototypes and with the speed with which he could implement new features. Thanks to his efforts we now have custom fit large ZigBee network analyzer. ”

Frank van Leeuwen & Luud Woltjer
Philips Lighting

Yi Xiao PDEng

The Design and Implementation of a Large ZigBee Network Analyzer

Philips Hue is the world's smartest home lighting system, which provides a new way of experiencing light at home. It allows users to create and control the lights via smartphones or tablets. Moreover, there are endless possibilities to help to personalize lighting for different lifestyles. Most important of all, Hue applications enable users to control their lighting whenever and wherever possible.

Hue devices are connected via a ZigBee mesh network. Currently, the Hue system supports up to 50 nodes that form the network. However, the performance and reliability of the system become uncertain in large networks. Therefore, a need for investigation how the data is transmitted within the network and how the performance and reliability change when the network is larger is required.

Therefore, the solution centered on the design and implementation of a large ZigBee analyzer, which aims to assist Hue developers and testers to investigate related issues. This analyzer visualizes the topology of a network by processing captured packets that are saved in .pcap files. Most importantly, this analyzer enables visualizing the multi-hopping dynamic routing behavior, which helps to analyze how packets are transmitted in a large ZigBee network.





“With strong determination Ronald led us on the road towards a modular design. Along the way he clearly recognized the difficult patterns in the current design and we learned valuable lessons for future projects. Thanks to Ronald we definitely reached our destination.”

Ir. Albert-Jan Nijsten
ASML

Challenges

Performing analysis and restructuring software in such a large system were the main challenges of this project. Not all software dependencies can be analyzed, so various abstraction levels had to be used. Restructuring parts of the existing code base proved challenging as well, especially because the code base is rapidly evolving.

Results

An overview of dependencies on the library was created. From that overview, the user interface was chosen as a case study. A replacement for previously used user interface control was designed, implemented, and integrated in various parts of the user interface. Further, a prototype for further decoupling the user interface from pattern recognition functionality was designed and implemented. In line of that effort, handles were given for the creation of a pattern recognition module.

Benefits

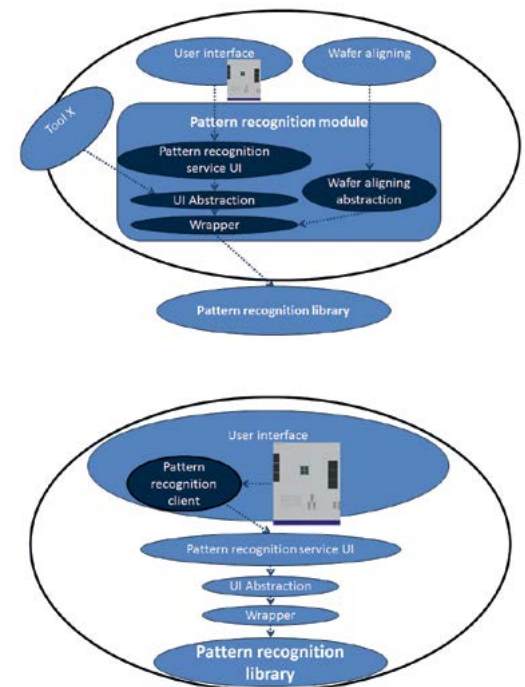
The analysis gives a clear view on the size of software parts depending on the library. With this analysis and the various designs, an approach is given how to continue to decouple the user interface from pattern recognition. With this, the software becomes more maintainable. It also serves as a case study for later redesign projects. Moreover, by decoupling the user interface from the library, also organizational dependencies have reduced.

Ronald van Zon PDEng

Towards a Modular Software Architecture for YieldStar's Pattern Recognition Functionality

The YieldStar machine performs quality measurements on a wafer. From these measurements, feedback is obtained with which the production process in a lithography machine can be improved. The YieldStar machine can be considered a young product, as it is still in the growth phase of the product lifecycle. Due to its market conditions, the product evolved rapidly from a prototype to a mature product. This has had impact on the maintainability of the software architecture. Improving the maintainability is one of the goals of this project. The YieldStar machine performs the measurements partly by using pattern recognition techniques. These techniques are provided by a third-party library. This library is used since the YieldStar was a prototype. Therefore, many parts of the software architecture are intertwined with logic of that library or with logic of pattern recognition in general.

An analysis was performed to visualize dependencies on the library. Upon that analysis a goal and scope was defined. A user interface control that was used directly from the library is replaced, such that the control is extendable. Further, the need for pattern recognition logic in the user interface is partly removed. This is realized by designing a web service that is deployed outside the user interface. The replacement for the user interface control was integrated with this web service as an example for other parts of the user interface that can be decoupled.



Ultimately, modularizing pattern recognition will improve maintainability greatly. Some handles for performing this step are given. Moreover, measuring the maintainability will give insights in trends and impact of future changes.

www.tue.nl/softwaretechnology

Follow us:



www.linkedin.com/company/tue_pdengst



www.facebook.com/TUePDEngST



www.twitter.com/tue_pdengst

Credits

Edited by: A.T.M. Aerts, Y. Dajsuren and M.L.L. van Oorschot,
Software Technology Program, Eindhoven University of Technology

Text: PDEng graduates of the Software Technology programme

Photography: Rien Meulman

Production: Communication Expertise Centre,
Eindhoven University of Technology

Design: Grefo Prepress

Printing: Drukkerij Snep

Filip Anastasovski PDEng; System for code generation for testing amplifiers at NXP Mobile Audio Department // **Sarwan Dar PDEng;** A Simulator for Spark // **Dongqi Hu PDEng;** Analysis Toolset for Calibration, Performance and Diagnostic (CPD) Applications // **Ana Kostadinovska PDEng;** Data Linkage Architecture for Population Health // **Aldo Daniel Martínez Márquez PDEng;** MBIF: An extensible framework for automatic generation of test cases and wrappers // **Vladimir Mikovski Iotov PDEng;** Performance Analysis Framework for a Complex Financial Data Warehouse // **Irina Nikeshkina PDEng;** ASML Simulation Development Environment // **Jeroen Redegeld PDEng;** A Framework to Stimulate Innovation Based on Automotive Data Collection // **Spyridon Skoumpakis PDEng;** Hue Bridge - Lighting Automation Engine // **Luc de Smet PDEng;** Impromptu: Reusable, accessible, and flexible data analysis // **Habtamu Aboma Tolera PDEng;** Model-based Testing Applied to Foreign Component Acceptance and Formal Components // **Tamir Tsedenjav PDEng;** CardioNotifier - A smartphone-based notification system for the cardiologist // **Juan Villa PDEng;** Domain-specific languages in Software Architecture // **Yi Xiao PDEng;** The Design and Implementation of a Large ZigBee Network Analyzer // **Ronald van Zon PDEng;** Towards a modular software architecture for YieldStar's pattern recognition functionality

The Software Technology PDEng (Professional Doctorate in Engineering) degree programme is an accredited and challenging two-year doctorate-level engineering degree programme.

During this programme trainees focus on strengthening their technical and non-technical competencies related to the effective and efficient design and development of software for resource constrained software intensive systems in an industrial setting. During the programme our PDEng trainees focus on systems architecting and designing software for software intensive systems in multiple application domains for the High Tech Industry.

The programme is provided by the Department of Mathematics and Computer Science of Eindhoven University of Technology in the context of the 4TU.School for Technological Design, Stan Ackermans Institute.

For more information, visit the website:

www.tue.nl/softwaretechnology