Posters from the Software Center Reporting workshop December 2021

- Accelerating Digitalization Through Data
- Enterprise Scale Continuous Integration and Delivery
- Strategic Ecosystem-Driven R&D Management
- Engineering Knowledge Flows in Large-Scale Agile System Development
- Using Digital Twin to Detect Cyber-Attacks On Cyber-Physical Systems
- Analysis of Timing Properties in System of Cyber-Physical Systems
- Flaky Tests & Software Center: In a Nutshell
- Managing Model Inconsistencies
- Automated Recovery of Data Pipelines
- Noise Handling For Improving Machine Learning-Based Test Case Selection
- Towards Federated Learning
- Challenges in developing and deploying AI in the engineering, procurement and construction industry
- Transforming Automo/ve Architecture with Assistance from AI
- Architectural Design and Verification/Validation of Systems with Machine Learning Components
Accelerating Digitalization Through Data

Research theme: Customer Data- and Ecosystem-Driven Development

Project number: Software Center #5

Partners:
Jeppesen Systems AB
Saab AB
Siemens AB
Volvo Car Corporation
Volvo Truck Corporation
Wärtsilä
Grundfos Holding A/S
Chalmers University of Technology
Malmö University

Summary
In this project, we help companies advance their adoption of data-driven development practices with the overall intention to ensure rapid and continuous delivery and improvement of customer value. We focus our research on 'value design' with the intention to help companies identify and agree on what they optimize for, align metrics at different levels and transition from a qualitative towards a quantitative understanding of customer value.

Value Design: Why and How?
- The purpose of value design is to move from implicit assumptions about the value of a feature, a product, a portfolio, a service offering etc., to explicating those assumptions.
- The process starts from a purely qualitative assessment of value factors but with the intention to translate this into quantitative and/or quantitative measures.

Value Design: What?
- Value design is a process intended to help explicate assumptions and move towards more quantitative assessment of customer value.
- Value design helps organizations ensure customer value and align low-level and high-level business metrics.
- Value design recognizes the multiple stakeholders and their different perspectives; (1) user perspective, (2) technology and product perspective, and (3) business and market perspective.

Contact us:
Helena Holmström Olsson
helena.holmstrom.olsson@mau.se
+46-40 6657319

Jan Bosch
jan.bosch@chalmers.se
+46-31-7725716

www.software-center.se
Enterprise Scale Continuous Integration and Delivery

Project acronym: ESCID
Research theme: Continuous Delivery
Project number: Software Center Project #6

Enable more frequent integration of software

Designing the CI/CD pipeline

Exploratory testing in the CI/CD pipeline

Nine key factors

The ExET model

The MaLET model
Strategic Ecosystem-Driven R&D Management

Research theme: Customer Data- and Ecosystem-Driven Development

Project number: Software Center #9

Partners: Saab AB, Siemens AB, Toyota Material Handling, Volvo Car Corporation, Volvo Truck Corporation, Ericsson AB, Jeppesen Systems AB, DEIF, Chalmers University of Technology, Malmö University

Summary
In this project, we study digitalization and digital transformation of the embedded systems industry and the many ways in which this impacts the business ecosystems in which companies operate. The goal of this project is to provide companies with strategic guidance for how to transition from traditional companies towards digital companies. This involves e.g., the transition towards more service-oriented revenue, continuous monetization of data and the adoption of new and innovative ways-of-working.

Digitalization and Digital Transformation
- Digitalization is transforming industry to an extent we have only seen the beginnings of
- With software, data and AI, companies seek to complement their existing physical offerings with new digital services
- To transition from transactional business models towards continuous (digital) business models is key to survive and stay competitive
- To increase service revenue, companies need to expand the scope of their business models and identify new revenue streams that go beyond their existing sales

Service monetization approaches
- Traditional product sales: Service opportunities and monetization is explored only after the customer has bought the product
- Edge customers: Service opportunities and monetization is explored in collaboration with "edge customers" (smaller customers that might be more open-minded and willing to experiment)
- New customers: Service opportunities and monetization is done outside the primary business and not with the main customer group(s)

Contact us:
Helena Holmström Olsson
helena.holmstrom.olsson@mau.se
+46-40-6657319

Jan Bosch
jan.bosch@chalmers.se
+46-31-7725716

www.software-center.se
Previously: RE for Large-Scale Agile System Development

Engineering Knowledge Flows in Large-Scale Agile System Development

Project acronym: Flow (previously: RE4Agile / MaRK-C)

Research theme: Customer Data- and Ecosystem-Driven Development

Project number: Software Center #27

Partners:
- Axis Communications AB
- Ericsson AB
- Grundfos Holding A/S
- Saab AB
- Siemens AB
- Tetra Pak
- Volvo Car Corporation
- Volvo Truck Corporation
- Zenseact
- Chalmers University of Technology
- University of Gothenburg

Summary

The Flow project (Engineering Knowledge Flows in Large-Scale Agile System Development, formerly known as RE for Large-Scale Agile System Development) provides new approaches to manage knowledge flows in large-scale agile system development. Traditional engineering disciplines and processes cannot keep up with the speed of agile development. Agile frameworks do not sufficiently cover important engineering knowledge. The Flow project specifically investigates challenges, best practices, methods, and tools for building, sharing, and maintaining engineering knowledge at scale. For this, we rely on 5 years of research on managing requirements knowledge in large-scale agile system development and extend it to related knowledge areas such as architecture, design, and release management knowledge.

Traceability in Large-Scale Agile System Engineering

Establish and foster collaborative traceability and construct traceability information models.
- Cross-organizational view on how knowledge connects
- Design and Define Traceability Strategy
- Coordination needs and Traceability Structure as complementing mechanisms to define information models and knowledge flows

Coordination Through Engineering Knowledge

Coordination through requirements, architectural decisions and other engineering knowledge: Improving coordination between methodological (agile) islands based on requirements-related boundary objects. This allows to:
- Consider the needs of developers, product owners, system managers, architects, ...
- Identify knowledge items
- Anticipate governance and shared maintenance

Contact us:

Eric Knauß
Eric.Knauß@ese.gu.se
+46 31 772 10 80

Jennifer Horkoff
Hans-Martin Heyns

Jan-Philipp Steghofer
Jörg Holtmann

www.software-center.se
Using Digital Twin to Detect Cyber-Attacks On Cyber-Physical Systems

Monitoring Cyber-Physical Systems and Detecting Attacks

An IDS using a Digital Twin

Intrusion Detection Systems (IDSs) are deployed in communication networks to detect attacks on the systems. IDSSs are generally equipped with a set of rules to detect the attacks. The rules are written by safety and security engineers.

Digital Twin represents an image of a physical object.

Our IDS monitors input and output of the system and employs a Digital Twin to detect cyber-attacks. The Digital Twin represents legitimate behavior of the system extracted at design time. We build the digital twin based on an actor model of the system, and model checking.

IMPACT
- Eliminate the need to write policies for detecting attacks by safety/security engineers
- Provide concrete tractable reports after a successful detection.
- Low False-Positive Rate (FPR)

NOVELTY
- Employing a Digital Twin to detect cyber-attacks including coordinated attacks.
- The Digital Twin is created based on an executable model and formal methods.

CONTACT
Project contact: Marjan Sirjani
Mälardalen University - Sweden
Email: marjan.sirjani@mdh.se

http://www.rebeca-lang.org/

Project #29 Partners

Feridoun Morad  Marjan Sirjani  Sara Abbaspour
Timing Analysis

- End-to-end delay, Age, ...
- Scheduling and Re-scheduling
- Safety Assurance and Robustness
- Availability

For SCPS such as:

- Automotive Architecture
- Interoperable Medical Devices
- Factories

A generic view of a SCPS:
Multiple systems are communicating through different middleware. Each system handles different triggers from inputs and creates outputs (periodically or event-driven).

We use an Actor Model for various Timing Analysis applying Formal Methods.

DEIF Multi-line 300 Platform

Computing max/min end-to-end delays

Project #29 Partners
Marjan Sirjani (MDH), Jesper Graversen (DEIF), Tobias Christensen (DEIF), Bahman Pourvatan (MDH)

http://www.rebeca-lang.org/
Where to Start – level 0 ---- Capture Perceptions of Flaky Tests Among Team Members [1]

Few Flaky Tests – level 1 ---- Flaky Test Prediction [2]

<table>
<thead>
<tr>
<th>Original code</th>
<th>Replication code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 0.369</td>
<td>Job 0.359</td>
</tr>
<tr>
<td>Git 0.344</td>
<td>Git 0.319</td>
</tr>
<tr>
<td>Id 0.343</td>
<td>Id 0.309</td>
</tr>
<tr>
<td>Action 0.337</td>
<td>Action 0.325</td>
</tr>
<tr>
<td>Action 0.335</td>
<td>Action 0.324</td>
</tr>
<tr>
<td>Custom 0.329</td>
<td>Custom 0.316</td>
</tr>
<tr>
<td>Custom 0.328</td>
<td>Custom 0.315</td>
</tr>
<tr>
<td>Grand 0.327</td>
<td>Grand 0.314</td>
</tr>
<tr>
<td>Grand 0.326</td>
<td>Grand 0.313</td>
</tr>
<tr>
<td>Command 0.325</td>
<td>Command 0.312</td>
</tr>
<tr>
<td>Command 0.324</td>
<td>Command 0.311</td>
</tr>
<tr>
<td>Exec 0.323</td>
<td>Exec 0.310</td>
</tr>
<tr>
<td>Exec 0.322</td>
<td>Exec 0.309</td>
</tr>
<tr>
<td>User 0.321</td>
<td>User 0.308</td>
</tr>
<tr>
<td>User 0.320</td>
<td>User 0.307</td>
</tr>
<tr>
<td>Duration 0.319</td>
<td>Duration 0.306</td>
</tr>
<tr>
<td>Duration 0.318</td>
<td>Duration 0.305</td>
</tr>
<tr>
<td>Output 0.317</td>
<td>Output 0.304</td>
</tr>
<tr>
<td>Output 0.316</td>
<td>Output 0.303</td>
</tr>
<tr>
<td>stdout 0.315</td>
<td>stdout 0.302</td>
</tr>
<tr>
<td>stdout 0.314</td>
<td>stdout 0.301</td>
</tr>
</tbody>
</table>

No Flaky Tests – level 2 ---- Multi-Factor Detection [3]

Multi-Factor approach
1. Number of test smells in the failed test cases (F1)
2. Whether failed test cases executed on latest code changes (F2)
3. Test cases history (how many times, the test cases failed due to unrelated code changes) (F3)
4. Test case size (perceived factors identified in our earlier study) (F4)

Above factors for decision making

Machine Learning
Do these test cases have the characteristics similar to the flaky one?

Output
Graphs and CSV files

Many Flaky Tests – level 3 ---- Divergence Algorithm for Randomness in Flaky Tests

Contact us:
Azeem Ahmad
Azeem.ahmad@liu.se
Kristian Sandahl
Kristian.sandahl@liu.se


CHAMBERS UNIVERSITY OF GOTHENBURG BÅSTAD UNIVERSITY OF VESTERÅS MALMÓ UNIVERSITY LIU JEPPESEN ERICSSON BOSCH GRUNDfos CEVT AXIS SIEMENS TOYOTA
Managing Model Inconsistencies

Project acronym: MMInc
Research theme: Continuous Architecture
Project number: #35

Partners: Mälardalen University Grundfos Holding A/S Saab AB
Associated Partners: Robert Bosch Tetra Pak

Summary
When adopting short development cycles, one of the most challenging tasks is to establish and maintain consistency between diverse development artifacts, for example system models, detailed design models, code, requirements, tests, etc. Complete consistency across all artefacts is usually not required nor desirable, but being notified of inconsistencies to prevent their propagation to other development artefacts is important. **Our project aims to provide a framework for lightweight consistency checks across development artefacts to avoid blockages on manual review tasks and ultimately, to support continuous model-based development.**

Consistency checking settings and solutions
In our most recent study we looked at different industrial settings of model-based development (MBD). We derived four levels of adoption towards continuous MBD and three steps to migrate between the levels.

**MBD adoption levels**
1) Disjunct artefacts + semantics
2) Linked artefacts, manual checks
3) Continuous MBD-ready

**Steps between levels**
0=>1) Model with semantics
1=>2) Bridge models
2=>3) Automated checks

Contact us:
Robbert Jongeling  
Jan Carlson  
Federico Ciccozi  
Antonio Cicchetti

www.software-center.se
Automated Recovery of Data Pipelines

Aiswarya Raj M*, Jan Bosch and Helena Holmström Olsson
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden
aiswarya@chalmers.se

1. Introduction

- Data is the new currency and key to success
- Collecting high-quality data from multiple distributed sources requires much effort
- Data pipelines are implemented in order to increase the overall efficiency of data flow from the source to the destination since it is automated and reduces the human involvement

2. Motivation

- Data pipeline failure due to faults at different stages of data pipelines is a common challenge that eventually leads to significant performance degradation of data-intensive systems

3. Objective

- To describe the faults encountered by the practitioners during the development and maintenance of data pipelines and the possible mitigation strategies
- To validate the data pipeline model by implementing fault detection and mitigation in an industrial data pipeline.

4. Conceptual model of Data pipeline

5. Typical faults and mitigation strategies

<table>
<thead>
<tr>
<th>Stage</th>
<th>Faults</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>Mitigation Strategies</th>
<th>Sending Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Generation</td>
<td>Data source failure</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Set a proxy which never fails</td>
<td>A1</td>
</tr>
<tr>
<td>Data Collection</td>
<td>Inactive data source</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Send notification to restart the source</td>
<td>X</td>
</tr>
<tr>
<td>Data Ingestion</td>
<td>Authorization failure</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Functional user credentials</td>
<td>X</td>
</tr>
<tr>
<td>Data Ingestion</td>
<td>Data sending job failure</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Send notification about failure</td>
<td>X</td>
</tr>
<tr>
<td>Data Ingestion</td>
<td>Unexpected data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Send email to flow guardian</td>
<td>X</td>
</tr>
<tr>
<td>Data Ingestion</td>
<td>Incompatible ingestion methods</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Log the error. Define dedicated ingest module</td>
<td>X</td>
</tr>
<tr>
<td>Data Ingestion</td>
<td>Data Extraction faults</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Conversion to acceptable format. Formulate the data</td>
<td>X</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Change in data format</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Versioning mechanisms</td>
<td>X</td>
</tr>
<tr>
<td>Data Processing</td>
<td>Inefficient storage</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Alert the developer and then to support team</td>
<td>X</td>
</tr>
<tr>
<td>Data Processing</td>
<td>Data duplication</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Use of HDF5</td>
<td>X</td>
</tr>
<tr>
<td>Data Processing</td>
<td>Infrastructure Failure</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Sending alarm to IT support</td>
<td>X</td>
</tr>
<tr>
<td>Data Processing</td>
<td>Transformation faults</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Define heuristics and rejections</td>
<td>X</td>
</tr>
<tr>
<td>Data Sink</td>
<td>Unclear definitions and wrong interpretations</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Contact SMES</td>
<td>X</td>
</tr>
<tr>
<td>Data Sink</td>
<td>Human errors</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Data deletion</td>
<td>X</td>
</tr>
<tr>
<td>Data Sink</td>
<td>Schema errors</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Define common schema and pre-defined language</td>
<td>X</td>
</tr>
<tr>
<td>Data Sink</td>
<td>Empty data</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Statistical methods, Data ingestion techniques</td>
<td>X</td>
</tr>
</tbody>
</table>

6. Automated Recovery

- Included 2 connector level components: Fault detection and Mitigation
- Inadequate bandwidth or insufficient resources
- Failed dump ids were 32,453 over 30 days (37% of total dump ids)
- Failed dumps are automatically resent as small batches along with the new dumps to the third parties for decryption

7. AI-Powered Fault Tolerance

8. References


Towards Federated Learning

Hongyi Zhang, Jan Bosch and Helena Holmström Olsson
Department of Computer Science and Engineering, Chalmers University of Technology, Sweden
hongyiz@chalmers.se

1. Introduction

- Nowadays, billions of devices constantly generate data.
- Data and ML/DL methods enable companies to produce better products and smarter services.
- Federated Learning technique does not require centralized data for model training, which is suitable for edge learning scenarios where nodes have limited data.
- However, despite the fact that Federated Learning has significant benefits, companies struggle with integrating Federated Learning components into their systems.

2. Objective

- To identify the reasons why our case company considers Federated Learning as an applicable technique.
- To summarize the services that a complete Federated Learning system needs to support in industrial scenarios.
- To identify the challenges that industries are attempting to solve when adopting and transitioning to Federated Learning.

3. Case Study

Three different use cases within one of the most well-known ICT (Information and Communication Technology) suppliers to service providers

- Data collection and analysis
- System architecture design and operation
- Machine learning project design, development and operation

4. Benefits

With Federated Learning:
- Data privacy-preserving
- Take advantage of the local computation
- Fast model deployment and evolution (Smarter models)

5. Required Services

- Communication services: A gateway service to handle service routing and expose the API interfaces
- Task registration and management service: Assure the service’s high availability
- Training Service: Meta data management and learning monitoring
- Model management service: Store and manage trained model

6. Challenges

We derive the challenges for industries stepping towards reliable Federated Learning services.

1. Components Failures: Federated Learning system robustness and fault tolerance issues are significantly more prevalent than in traditional distributed system environments.

2. Inefficient Communication: The problems of how to reduce the communication round in real industrial scenarios, how to efficiently utilize network resources while maintaining or even improving model prediction performance still need to be searched and verified.

3. Unstable Model Performance: The problem of how to ensure the model can perform well on all the edge devices in Federated Learning systems are still tricky and need to be verified in different real-world application scenarios.

4. Large Number of End Customers: The mechanism of how to handle device joining in and schedule device utilization still need to be researched and algorithms can be designed.

5. Incomplete System Security: With the increasing attention and focus on AI-powered industrial solutions, security issues are more essential to the service provider as well as commercial Federated Learning systems.

7. References

Software Center: Metrics project

**Noise Handling For Improving Machine Learning-Based Test Case Selection**

**Vision:**
Testing in continuous integration (CI) is a cost-intensive process that requires heavy computational power and long execution time of tests at every CI cycle. We propose a machine learning (ML) based method that uses textual analysis of source code to reduce the size of test suites in CI.

Our vision is to reduce the time required by CI systems to select and execute tests that reveal faults in new code integrations.

**Industrial research partners:**
- Ericsson
- Grundfos
- Axis

**Approach: Method Using Bag of Words for Test Case Selection**
- MeBoTS extracts code changes from the development repository.
- Uses Bag of Words for modelling dependencies between code changes and test case execution outcomes.
- Builds a classifier from BoW to predict which code lines will trigger a test case failure/pass.

---

**Class Noise In Source Code Data**

<table>
<thead>
<tr>
<th>line</th>
<th>literal</th>
<th>int</th>
<th>main</th>
<th>include</th>
<th>( )</th>
<th>quote</th>
<th>std</th>
<th>cout</th>
<th>for</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Lines 8 and 13 are contradictory lines that were extracted from two revisions.
- In the first revision, the test case failed -> class is set to '0'.
- In the second revision, the test case passed -> class is set to '1'.

**Noise handling Approach**
- Select syntactically unique lines.
- Relabel the class value of contradictory entries from 0 to 1.

**Attribute Noise In Source Code Data**

<table>
<thead>
<tr>
<th>line</th>
<th>if</th>
<th>var</th>
<th>( )</th>
<th>&lt; &gt;</th>
<th>and</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The attribute values in line 15 deviate from the values in lines 12, 13, and 14.
INTRODUCTION

- AI application in the engineering, procurement and construction (EPC) industry has not yet proven track records in large-scale projects since
- AI solutions for industrial applications became available only recently, and experience of deployment and lessons learned are still to be built up
- Shortage of in-depth studies that focus on experience from EPC companies with high safety standards like the petrochemical or energy sector

OBJECTIVE

What are the primary challenges the engineering and construction industry is experiencing when selecting, developing, and deploying ML/DL solutions?

INTRODUCTION TO THE AI PROJECTS DEVELOPED WITHIN EPC COMPANY

A. “Defect identification on the engineering drawings”

Can highlight the specific patterns on P&IDs (piping and instrumentation diagrams) identified as defects or, in other words, engineering mistakes.

Users: Process engineers might use the developed tool to review engineering drawings before issuing them to the construction team.

B. “Engineering hours budget prediction tool”

Utilizes historical data for future engineering hours budget prediction

Users: The developed engineering hours prediction tool is used by functional management as a reference when reviewing the hours’ estimate made by discipline lead engineers during the proposal phase of the project.

PROJECT CHALLENGING MAPPING

<table>
<thead>
<tr>
<th>Category</th>
<th>Challenges</th>
<th>Use Case 1</th>
<th>Use Case 2</th>
<th>Use Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Challenges in adequately estimate the business</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architectural</td>
<td>Challenges in identifying the activities in the A/E/MI proposal contracts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Challenges in identifying the resources to be invested in the AI solution</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Heat development challenges</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>High variation of graphical symbol representations</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Challenges in integrating into company software platforms</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usability and efficiency of the software</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Legal constraints</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unreliable databases</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Challenges in gathering data across company</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No standard definition of targeted solutions</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Line of priority analysis</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Poor definition of the collected statistics</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inadequate cost + time</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Challenges in go-data setting</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

We described in detail the primary challenges which EPC industry is experiencing when selecting, developing, and deploying ML/DL solutions.

The findings show:
- Domain experts must gain hands-on experience managing AI projects to overcome business, development and organizational constraints
- Top management commitment is essential to support the start of AI project development and provide the resources to launch the fail-safe projects that are necessary before deploying AI tools on a large scale

CONCLUSION
Transforming Automotive Architecture with Assistance from AI

Vision:
To train a deep-learning model that understands software design and eventually use it to assist software design tasks. The collaboration of AI and human experts on software design has the potential to accelerate software engineering by maintaining high-quality software architecture over time.

Current work:
Programming Language Understanding (PLU) models are typically trained on large corpora of source code. We ask the question – has a PLU model, which has seen millions of coding examples during training, implicitly picked up ideas of design? We test this by setting design-related challenges to a pre-trained model.

metrics theme
Research partners:
- CEVT
- Chalmers
- Volvo AB
- Volvo Cars

STEP 1: Training a PLU model for the C-language

```
// Get BatterySOC
input->batterySOC.status = SIG_OK;
rteRet = Rte_Read_BatterySOC_rqst(&input->batterySOC.data);
if (rteRet == RTE_E_OK)
{
    rangeRet = validateSignal(input->batterySOC.data);
    // Check range
    if (rangeRet == SIG_IN_RANGE)
    {
        // Everything OK, assign signal
        input->batterySOC = SIG_OK;
    }
}
```

STEP 2: Testing/probing the model on AUTOSAR knowledge

```
; + ( ] struct case if switch return signal handler control active
Less design sensitive More design sensitive
And these difficult?
```

First results:
- A C-language PLU model pre-trained on ~100 million non-automotive C-files from GitHub seems to be quite comfortable in handling AUTOSAR-C code
- This is an encouraging sign that this PLU model understands key AUTOSAR design contexts, and can help solving design (and many more) related tasks

Ongoing work – we’re currently testing if the PLU model understands AUTOSAR design relationships. Here again initial results are encouraging

dhasarathy.Parthasarathy@volvo.com
Volvo AB
Architectural Design and Verification/Validation of Systems with Machine Learning Components

Project Background
Due to their stochastic nature, using machine learning (ML) components requires specific approaches related to the architectural design and verification/validation of the vehicle’s electrical system. On the one side, suitable architecture containing both stochastic and traditional deterministic components shall be defined which assures safety, robustness, fault tolerance, and the exchange of data between ML components and the rest of the vehicle. On the other side, the nature of ML algorithms makes them difficult to test and analyze from a safety perspective, which requires implementation of specific verification and validation methods.

Semantic Anomaly Detection (RQ3)
Deep learning (DL) algorithms solving perception tasks in the automotive industry can fail when applied on data which is significantly different from the training data. In order to minimize the risks related to these failures, anomaly detection methods can be used, in particular, anomaly detection approaches based on autoencoders (Fig. 1).

When an autoencoder is trained using only normal data, the reconstruction error calculated as the difference between the output in the input data can be used as anomaly score. In this study, we answer the following RQ: How capable are autoencoders in detecting semantic anomalies in highway driving scenarios? Fig. 2 presents examples of the input, output, and reconstruction error of the autoencoder. The results obtained in this study show that the autoencoders are capable of detecting semantic anomalies to some degree.

General Research Questions
RQ1: How do we make a sustainable architectural design of an automotive system as a combination of individual ML software components and traditional safety critical components?
RQ2: How to assure quality of the architectural design of future car architectures developed in an agile way, where ML and stochastic calculations are part of the safety-critical functions?
RQ3: What are reliable measures of model certainty that can be used as quality indicators by systems architects and how do we infer them from the data and model?

Test Input Prioritization (RQ3)
The goal of test input prioritization is to prioritize test inputs that are more likely to reveal the errors of a ML algorithm faster during the testing. It can also reduce the testing cost by eliminating the need of labelling of unnecessary test inputs. In this study, we compare white-box, data-box, and black-box input prioritization techniques in terms of their effectiveness and efficiency (RQ): How effective and efficient are surprise adequacy, autoencoder-based, and similarity-based input prioritization approaches for testing DL algorithms? Fig. 3 demonstrates the measured effectiveness of different approaches as the area under the curve representing the cumulative number of errors growing during the testing against the test inputs in the prioritized order. The results show that the different techniques are more and less effective and efficient in different cases.

Architectural Patterns for ML (RQ1)
The goal of this case study is to document experiences of which of the emerging ML architectural patterns are used in modern cars already, which are planned for the future and how they are used. We study the software architecture of a modern car product line based on existing patterns and workshops with software architects (RQ: How applicable are the architectural patterns for ML in vehicle software architecture in practice?). The results show that many ML-specific patterns are used or planned to be used in the near future (Fig. 4).

Figure 1. Autoencoder architecture.
Figure 2. Examples of the input, output, and reconstruction error of the autoencoder.
Figure 3. Effectiveness of different test input prioritization techniques.
Figure 4. Applicability of architectural patterns for ML in vehicle software.