The Role of Computer Science and Software Technology in Organizing Universities for Industry 4.0 and Beyond

Prof. dr. ir. Mehmet Aksit
m.aksit@utwente.nl
Department of Computer Science, University of Twente, Enschede, The Netherlands,
http://www.utwente.nl/ewi/trese/people/Aksit/
http://fmt.cs.utwente.nl/
Contents

- Making strategic plans of companies reality!

  **Approach:**

  - The company maturation process;

  **Justifications:**

  - Motivation 1: Researchers perspective;
  - Motivation 2: Trends in industry
  - Motivation 3: Re-shaping universities
  - Motivation 4: CS
  - Motivation 5: Software engineering and technology

- Conclusions.
The beginning of all!

The problem statement
Higher-level objectives

Given a set-of (potentially) strategically important companies

- How to determine the strategic objectives of these companies?

- How to turn the strategic objectives into tactical and implementation objectives?

- How to steer the overall process and be sure that it is converging to these objectives?
Possible answers

- How to determine the strategic objectives of these companies?
- How to turn the strategic objectives into tactical and implementation objectives?
- How to steer the overall process and be sure that it is converging to these objectives?

I think companies should know better than me, but they must be aware of the trends! You know software is crucial: every company will be a software company in the future;

- Apply problem solving processes;
- How to steer the overall process and be sure that it is converging to these objectives?

Define a process that: (1) classifies companies according to their maturity levels and identify the current level; (2) define maturation processes; (3) steer the overall process!
The maturation process
The intention is to enhance the maturity of software techniques by guiding the companies along this process.

0. Introduction

1. Transition

2. Description

3. Knowledge generation

4. Adoption

5. Application

6. Leadership

Looks like a feedback control system

Coordinator

Academic research steering committee

Company 1

Company 2

.....

Company N

University 1

University M
The process is described and approved as a report.
(1) companies present their strategic plans to a team of researchers

The intention is to identify the obstacles/challenges in reaching the objectives!
(2) Researchers and company engineers work together extensively to identify the (future) challenges and prospective solutions
(3) Reports are written, discussed, modified and accepted, by describing the definitions of the terminology, the context, the observed research challenges and the related research work.

Terminology, State-of-the-art, Current research, Return on investment
Mapping concerns from the business domain to CS research domain!
Research activities are grouped around the product-lines

<table>
<thead>
<tr>
<th>Product-lines</th>
<th>Required capabilities (Ph.D. studies, etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 1: Product-line 1</td>
<td>Cap.</td>
</tr>
<tr>
<td>Company 1: Product-line 2</td>
<td>Cap.</td>
</tr>
<tr>
<td>Company 2: Product-line 1</td>
<td>Cap.</td>
</tr>
<tr>
<td>Company N: Product-line M</td>
<td>Cap.</td>
</tr>
</tbody>
</table>

- 6. Leadership
- 5. Application
- 4. Adoption
- 3. Knowledge generation
- 2. Description
- 1. Transition
- 0. Introduction
A reactive feedback-control process model is defined; An example: part of the process for level 3

The committee is also responsible for synergy & coordination & scientific progress
The technical results
The Product Domain Categories

Company 1:
– On-board systems (14 Ph.D.’s);
– Simulation systems (10 Ph.D.’s);
– Mission critical enterprise systems (14 Ph.D.’s);
– Mission critical systems of systems (10 Ph.D.’s);

Company 2:
– Avionics systems (15 Ph.D.’s);
– Communication systems of systems (9 Ph.D.’s);
– Mission-critical hybrid control systems (6 Ph.D.’s);
– Sensor fusion systems (3 Ph.D.’s);

Company 3:
– Intelligent platforms (8 Ph.D.’s);
– Distributed simulation (6 Ph.D.’s);
– Intelligent control systems (6 Ph.D.’s);

Company 4:
– Modernization of flight systems (11 Ph.D.’s);
– Satellite systems (14 Ph.D.’s);
– Unmanned aerial systems (17 Ph.D.’s);
– Generic fault tolerant computing (2 Ph.D.’s);
– Generic fault-removal techniques (5 Ph.D.’s);

Including myself, this work has been carried out together with:

Prof. dr. ir. Bedir Tekinerdoğan;
Assoc. prof. Hasan Sözer;
Hakan Faruk Safi;
Meryem Ayas.

These are large, safety-critical systems!
The Research Categories

- Adaptability and reuse (25%)
- Model-based verification (24%)
- Domain-specific architectures (12%)
- Domain-specific languages (10%)
- Design methods (8%)
- System design (6%)
- Model building (6%)
- Data/cloud management (3%), Application frameworks (3%), Architecture business assessment (3%)
Motivation 1:

from

Researcher’s point of view
Needs for advanced research in companies

- Computer science and technology advances rapidly. This makes software developing companies difficult to follow the advancements effectively.

- Tight project schedules and limited budget for research and development make this process even harder.

- As a result, companies have difficulties in achieving the excellence in computer science and software technology to overcome their technical challenges.
Methods used in research granting programs

Current university-industry collaborations are mostly based on governmental grants of university research programs, Joint Ventures (JVs) and Framework Programs (FP6, FP7, etc.).

- The existing university-industry cooperation methods are based on writing grant proposals to financing organizations and passing through strict selection processes. This is, in general, a very inefficient and tedious process to undertake.

Many good ideas may not go through simply because they do not fulfill the necessary procedural requirements. Probability of success is very low.
From the researcher’s perspective

- Researchers do not waste time in writing long and tedious grant proposals;

- Researchers are directly involved in the maturation process of the companies;

- Scientific quality is monitored and guaranteed by the steering committee, but

- the universities & researchers must agree on the way of working!

I must adapt!
Motivation 2: 

from the perspectives of the 

Trends in industry
Trends (i)


2. Focus on owning and managing knowledge and skills and intellectual property rights instead of focusing on labor/resource intensive manufacturing processes.

3. Dynamically managed and optimized, multi-asset portfolio instead of fixed/ad-hoc, single-asset portfolio.

Trends (ii)

5. Proactive self-organizing companies instead of inflexible hierarchically-organized companies.

6. End-to-end alignment and optimization of (manufacturing) processes instead of focusing only on the improvement of individual phases.

7. Multi-disciplinary usage of teamed personnel instead of working with solely operating individuals.

8. Organizing businesses/enterprises globally through networks instead of isolated and/or localized organizations.
Trends (iii)

9. Improved time-to-market instead of long sequences of research, design, manufacturing and marketing phases.

10. Intensive use of state-of-art CS-ST as the “main enabler” of modern businesses instead of considering CS-ST just like any other technical skill.

11. Strong cooperation with universities for the purpose of innovation instead of considering universities mainly as theoretical institutions that educate people.

We cannot debug Reality! So,…
What about Industry 4.0?

- **Interoperability** meaning that sensors, devices, machines, and people can connect and exchange information with each other.

- **Information transparency** meaning that a rich set of data can be gathered from various sources.

- **Technical assistance** meaning that machines, systems, processes, human beings, etc. can be intelligently and effectively assisted to monitor, control and optimize the overall manufacturing process.

- **Decentralized decisions** meaning that subsystems can autonomously take decisions where possible.
What they said about Industry 4.0

- With Industry 4.0, it is expected that machines and systems will become more self-aware and self-learning so that their effectiveness and maintenance can be improved.

- Due to networked data gathering and intelligent and autonomous process control, the manufacturing processes will be much more efficient and effective than traditional manufacturing processes.

- It is claimed that Industry 4.0 is the 4th industrial revolution in the history of manufacturing.
Our assessment of Industry 4.0

We think that the concepts relevant to Industry 4.0 must be defined and understood in the process of on-going transition from traditional to modern manufacturing processes. It is important to stress that such transitions are not abrupt in nature but gradual, depending on the characteristics of manufacturing, technological and societal progresses.

- It is clear from “the list of trends” in this presentation that Industry 4.0 refers to (or a new name of) a part of a natural transition in manufacturing processes which has been taking place since several decades. We therefore term “the list of trends” as “a list of attributes of manufacturing processes for Industry 4.0 and beyond”.
Motivation 3:

for

Re-shaping universities for the future
Traditional universities

- **Research**, where academic personnel of the university are expected to be expert in certain fields. The selection of the topic of a field is not necessarily derived from industrial and societal needs; it can be ad hoc. The expertise is quite specific and theoretical. The excellence is measured according to number of publications in certain pre-classified journals.

- **Education**, where academic personnel of the university are expected to give lectures in their fields of expertise and examine the students by oral and written tests. In addition, students are expected to be supervised in writing their theses.
How universities should be! (i)

1. The specializations **must be derived from the needs of the targeted society and industry** instead of ad-hoc selection of topics;

2. The academic personnel **must learn to work together in multi-disciplinary teams.**

3. The academic personnel **must be proactive in forming networks** to cooperate with national and international institutions and colleagues not only from his/her own discipline but also from other disciplines.

4. The academic personnel **must be flexible enough to adapt themselves in changing demands from industry and society.**
How universities should be! (ii)

5. The education process must be tailored to answer the mid-term and long-term needs of industry and society:

- It must focus on the core concepts instead of hypes.
- It must focus on gaining analytical skills, critical thinking and reasoning instead of memorizing what are in the books.
- It must aim at teaching problem solving/synthesis instead of gaining knowledge which cannot be utilized for solving actual problems.
- It must emphasize working in multi-disciplinary projects instead of only focusing on mono-disciplinary exercises.
- It must enhance communication skills, such as oral and written presentation and argumentation skills instead of educating students with non-communicative and introvert attitude.
- It must aim at increasing consciousness of students in ethical concerns instead of educating students with irresponsible and/or indifferent attitude.
How universities should be! (iii)

6. The university **must create suitable organizational structures** to enable the academic personnel efficiently and effectively fulfil the objectives listed above. These include:

- **Proactive and self-adaptive organization** to support the objectives of the university in dynamically changing contexts.
- **Organization to set-up and carry-out multi-disciplinary projects** for industry and society.
- **Organization with an award system** to motivate the academic personnel and students along the objectives of the university.
- **Organization which emphasizes CS-ST** since it is the “main enabler” of all disciplines at the university.

**University industry, hand in hand..**
Important quality attributes (i)

- **Relevancy**: The university must be highly relevant in addressing technical and social needs.

- **Alignment with the current state-of-the-art research**: The research and education activities to be carried out must advance the state-of-the-art so that the companies and businesses can be matured to be the leaders in their context.

- **Cross-fertilization**: Different university research and education activities can benefit from each other.
Important quality attributes (ii)

- **Industry-as-laboratory**: To identify the relevant problems and to test the proposed solutions, it is important that the principle investigators and the affiliated (Ph.D. and/or M.Sc., etc.) students visit the companies regularly and carry out experiments within industrial and societal context.

- **Academic research steering committee**: To coordinate the activities effectively and efficiently, it is important to monitor the progress of research and education activities and evaluate them with respect to the desired objectives.
Motivation 4

for

Computer Science!
Cannot be without CS

CS-ST is the main force in almost all industries; it creates added value for products and businesses.

There is almost no product in the market which does not contain software or is not produced by a process controlled by software.

To accomplish the objectives of Industry 4.0 and beyond, advanced CS-ST is needed.

I said, You cannot debug reality!
Recent developments in CS

1. Large infrastructures, service-oriented architectures, cloud computing, systems-of-systems.
2. Sensors, Internet of Things (IOT), and pervasive computing.
3. Big data and big data analytics.
5. Cyber-physical systems.
6. Artificial intelligence and related topics including computational intelligence, machine learning and multi-agent systems.
7. Graphical processing, and visualization including virtual reality.
8. High performance, and/or multi-core/parallel architectures including parallel programming.
9. Theoretical and practical work on algorithms and/or constraint-based “solvers” to address a large category of mathematical problems. In general algorithms/solvers are applied to every category of computer science specializations listed in this section.
10. Software (engineering) methods and techniques to fulfil the functional and qualitative requirements of software systems. The concepts of software engineering can be applied to every computer science specialization listed in this section.
Motivation 5

from the perspectives of

The role of software engineering and technology
The importance of software engineering and technology comes from the economics..

- Economical, sustainable and robust software systems which fulfill functional and qualitative requirements are essential for all software systems.

- To accomplish the requirements of Industry 4.0 and beyond, software engineering methods and techniques are crucial.

- No matter how intelligent a software solution is, if it cannot be realized with the desired quality attributes, one cannot expect an economical value out of it.

- As such software engineering methods and techniques can be defined as crosscutting (meta-level) concerns that relate to all developments within CS-ST.
The trends in software engineering and technology
Product-lines

Product-line instead of product design. Most products are developed and manufactured by specialized companies, which market families of products. It is not economical to develop each product from scratch.
Systems of systems instead of systems perspective. Software systems for Industry 4.0 are generally adopted in large distributed settings. Scale-ability and interoperability of systems are essential. Systems of systems architectures, are therefore the natural candidates of the platforms of Industry 4.0 architectures.

Company 1:
- On-board systems (14 Ph.D.’s);
- Simulation systems (10 Ph.D.’s);
- Mission critical enterprise systems (14 Ph.D.’s);
- Mission critical systems of systems (10 Ph.D.’s);

Company 2:
- Avionics systems (15 Ph.D.’s);
- Communication systems of systems (9 Ph.D.’s);
- Mission-critical hybrid control systems (6 Ph.D.’s);
- Sensor fusion systems (3 Ph.D.’s);

Company 3:
- Intelligent platforms (8 Ph.D.’s);
- Distributed simulation (6 Ph.D.’s);
- Intelligent control systems (6 Ph.D.’s);

Company 4:
- Modernization of flight systems (11 Ph.D.’s);
- Satellite systems (14 Ph.D.’s);
- Unmanned aerial systems (17 Ph.D.’s);
- Generic fault tolerant computing (2 Ph.D.’s);
- Generic fault-removal techniques (5 Ph.D.’s);
Ecosystems

Ecosystem design instead of platform design. Software ecosystems are an effective and economical way to construct large software systems for Industry 4.0 on top of a software platform by adding up software modules developed by different actors. In ecosystem design software engineering is spread outside the traditional borders of software companies to a group of companies and private persons.

Company 1:
- On-board systems (14 Ph.D.’s);
- Simulation systems (10 Ph.D.’s);
- Mission critical enterprise systems (14 Ph.D.’s);
- Mission critical systems of systems (10 Ph.D.’s);

Company 2:
- Avionics systems (15 Ph.D.’s);
- Communication systems of systems (9 Ph.D.’s);
- Mission-critical hybrid control systems (6 Ph.D.’s);
- Sensor fusion systems (3 Ph.D.’s);

Company 3:
- Intelligent platforms (8 Ph.D.’s);
- Distributed simulation (6 Ph.D.’s);
- Intelligent control systems (6 Ph.D.’s);

Company 4:
- Modernization of flight systems (11 Ph.D.’s);
- Satellite systems (14 Ph.D.’s);
- Unmanned aerial systems (17 Ph.D.’s);
- Generic fault tolerant computing (2 Ph.D.’s);
- Generic fault-removal techniques (5 Ph.D.’s);
Auto-adaptive control architectures

instead of architectures without any control mechanisms. To realize the monitoring and controlling activities in Industry 4.0 and to cope with the changing requirements and context, software systems are expected to be more reactive and self-adaptive. This generally requires built-in feedback control mechanisms in software. Self-adaptation can be realized at system level, subsystem level and/or at component-level. In addition, different styles can be adopted, such as single, master-slave, hierarchical and/or peer-to-peer control architectures.
Distributed problem solving including distributed algorithms, coordinating systems and multi-agent architectures instead of centralized problem solving with monolithic and/or localized architectures. Since computer systems for Industry 4.0 are distributed, to reduce complexity and enhance reliability/availability, algorithms and intelligence in systems must be distributed as well. Accordingly, programming languages and techniques must adequately support distributed programming efforts by offering expressive and flexible abstractions.
Model-based development instead of straight-forward programming. Since more and more companies are specialized in certain product categories and in manufacturing processes, deriving software architecture from relevant domain models can help in reducing complexity, enhancing reuse and testability/verifiability of software systems. Model-based development has been adopted in various approaches such as product-line engineering (SPL), model-driven engineering (MDE), domain specific architectures (DSA) and domain-specific programming languages (DSL), model-based verification (MBV).
Multi-objective optimization instead of ad-hoc hand-crafted and/or single objective optimization. Along the line of model-based development, various algorithmic techniques and search-based methods have been introduced to compute the “optimal” architectural decomposition with respect to certain quality attributes. In addition, various run-time optimization techniques can be adopted in computing optimal control strategies and scheduling processes.
Modularization of semantic concerns instead of traditional abstraction mechanisms based on implementation concerns such as data or function. As a consequence of model-based development, software abstractions more and more correspond to the concerns of models. The concerns of a model are naturally based on the semantics of the model, and these cannot always be effectively represented as a data or function. Moreover, concerns in Industry 4.0 systems can be emerging meaning that they may appear or disappear dynamically. As such, programming languages and techniques must adequately support programming efforts by offering expressive and flexible abstractions for emergent semantic concerns.

Abstract
From the early days of computers, researchers have been trying to invent effective and efficient means for expressing software systems through the introduction of new programming languages. In the early days, due to the limitations of the technology, the abstractions of the programming languages were conceptually close to the abstractions of the von Neumann based realization platforms. With the advancement of the technology, computers have been increasingly applied for complex problems in different application domains. This required the challenge of designing programming languages that resemble more the semantics of software rather than the concepts of underlying machines. To this aim, various new language concepts, such as object-oriented, aspect-oriented, and event-based languages have been introduced. While these languages were successful in expressing the expression power of languages towards more semantic concerns of application domains, they fail in short in representing emergent behavioral patterns of software effectively. We outline a set of requirements to overcome these shortcomings, and explain the concept of event-based modularization as a possible solution.

Categories and Subject Descriptors D.3.3 [Programming Languages]: Language Constructs and Features—Modules, packages

General Terms Languages, Design

Keywords Emergent behavior, modularity, von Neumann architecture

1. Introduction
Although there are many facets of software engineering, its main objective is to create software systems that execute on hardware/software platforms [1, p.6-12]. From the early days of computers, researchers have been trying to invent effective and efficient means for expressing software systems through the introduction of new programming languages.
A rich set of composition mechanisms instead of a fixed set of language constructs for hierarchical organization of programs (such as class-inheritance). To support flexibility in control strategies and to cope with various evolution schemes, languages must offer generic and/or domain specific composition mechanisms to express, for example, object, aspect and event compositions and transformational techniques in a uniform manner. The languages must maintain their closure property in compositions so that scale-ability of systems can be provided.
Uniform integration of verification techniques instead of independent tool- and technique-specific verification approaches. There are various model-based verification and testing approaches available. Examples are model-checking, static and dynamic analysis, run-time verification, model-based testing, adopting model-specific verification (simultaneously) based on continuous and/or discrete models, etc. Most of these techniques are complementary and as such combined usage of these may help in finding faults with less false-positive and false-negative cases.
Conclusions
Conclusions

- The trends in industry cannot be controlled! Therefore they must be understood and managed, as much as possible;

- We have presented a unique set of features of these trends and position these w.r.t. Industry 4.0;

- Define a company maturation process to help companies to deal with the trends;

- We have applied this process to 4 large companies;

- We have evaluated these w.r.t. to the trends in CS and present the expected challenges/topics in software engineering and technology.