ISSN 2186-7437

NII Shonan Meeting Report

No. 2017-18

Model-Based Design for Smart Products and Systems: Advanced Capabilities and Challenging Applications

Fuyuki Ishikawa, National Institute of Informatics, Japan
Peter Gorm Larsen, Aarhus University, Denmark
John S Fitzgerald, Newcastle University, United Kingdom

December 4–7, 2017

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Model-Based Design for Smart Products and Systems: Advanced Capabilities and Challenging Applications

Organizers:
Dr Fuyuki Ishikawa (NII, Japan)
Prof Dr Peter Gorm Larsen (Aarhus University, Denmark)
Prof Dr John Fitzgerald (Newcastle University, UK)

December 4–7, 2017

1 Introduction

This report is an account of activities at the NII Shonan Meeting on Model-based Design for Smart Products and Systems held on December 4-7, 2017 at Shonan International Village, Japan.

The objective of the meeting was to identify opportunities and challenges in developing methods and tools for model-based engineering of systems that are enabled by, and dependent on, networked computing technology. This category of "smart" systems is growing ever wider, and ranges from small autonomous devices to large-scale infrastructure[1]. The availability of data and the growing power to process data flexibly and at scale bring great potential benefits, but also challenge the state of the art in design methods and tools. Some of the most fundamental questions relate to the ways in which engineers work across traditional boundaries between disciplines. Our meeting brought together scientists and engineers from a wide range of backgrounds to appraise the state of the art, develop a shared vision, and prioritise challenges for future work at a range of technology levels.

In this report, we first review the motivation for the meeting, and in particular the need to develop cross-disciplinary methods and tools to address the challenges and leverage the opportunities afforded by smart systems engineering (Section 2). Our meeting was intended to catalyse new collaborations, and so we focussed on discussion and group activities, beginning from two invited presentations: one on the technology of co-simulation of heterogeneous design models, and one on the particular issues raised by Systems of Systems. These are briefly reviewed in Section 3. The participants (who are listed in Section 6) worked in focussed groups to clarify the highest priority challenges and opportunities that they saw in this field (Section 4), and potential collaborative opportunities (Section 5). A common theme in all the discussion groups was the need for research in this field to be driven by the needs of practitioner engineers, and so

---

[1] When we refer to "engineered systems" here, we include interventions in existing systems, such as the integration of new sensors into a building, for example.

collaboration opportunities identified included work on potential case studies. One such study in dynamic system evolution was proposed in some detail during the workshop and is outlined informally in Appendix A. The participants resolved to work to grow the community of interest in model-based methods for engineering smart products and systems, and to build on the work begin in Shonan by working towards a Thematic Issue of the Journal of Software and Systems Modeling. The Call for Papers is also included as Appendix B.

2 Background: the Need for Cross-disciplinary Methods and Tools

The rapidity of technological development has made time-to-market a key to commercially successful innovation. At the same time, growth in communication capabilities has led to new interdependencies between engineered systems and the other systems with which they interact. This means that better integration is required between design disciplines and between life cycle phases. New technology possibilities – for example in electronics, virtual reality and 3D printing – are emerging and may disrupt existing solutions[2] [1]. There is therefore an urgent need to ensure that researchers from diverse disciplinary backgrounds can combine different models into well-founded but heterogeneous collections that describe the key characteristics of these new emerging smart systems [2].

There are different dimensions to the increasing level of smartness in systems. One dimension is the relationship of the engineered system to the overall (eco-)system of which it forms a part. Along this dimension, we envisage components, products, connected products, product systems and systems of systems. Another dimension envisaged in [2] concerns how an individual engineered system evolves to raise its level of smartness from monitoring, control and optimisation to autonomous behaviour (self-coordination with other systems in the environment, self-diagnosis and enhancement).

Currently users expect systems from different suppliers to interoperate seamlessly in ways that may not have been considered when the individual systems were conceived. In particular, in a business-to-business value chain it is important to have a vision for the evolution of interoperability between systems. Artefacts produced in one product lifecycle phase may subsequently be needed activities that were not originally envisaged. For example, CAD drawings might subsequently be used for 3D printing or to support augmented reality views after deployment. There also could be monitoring of deployed products that can feed information back to either the development or production phase based on big data analysis.

For each individual product, different levels of smartness can be achieved, each enabled by particular technologies:

1. **Monitoring:** Sensors and external data sources enable monitoring of aspects of the system and its environment. Currently, the Internet of Things (IoT)[3] is a significant enabling technology for monitoring.

---

[2] Disruptive IoT Innovation, Article No :1274 — July 14, 2014 — by Avi Itzkovitch, UX Magazine, `https://uxmag.com/articles/disruptive-iot-innovation`

[3] See, for example `https://smartanythingeverywhere.eu/`

2. **Control:** Given a monitoring capability, embedded software enables control of system functions, including adaptation to user needs. Currently, a key enabling technology is that of Cyber-Physical Systems (CPSs)[4] coupled with cloud computing to provide elastic resources for storage and processing large volumes of gathered data.
3. **Optimisation:** Given monitoring and control capabilities, it becomes possible to optimise performance by predicting forthcoming system behaviours and environmental conditions. Here too, a key enabling technology is that of big data analysis[5] or Machine Learning (ML)[6].
4. **Autonomy:** In certain domains, a degree of autonomy is sought. In practice, a balance is often sought between such autonomy and the capacity to collaborate and interact with external systems, for example to create systems that optimise performance and carry out self-diagnostics in a safe manner. Expertise in fields such as safety, security and dependability must be involved if the system has an ability to do damage[7].

Model-based technologies for systems engineering have been the subject of research and innovation activities worldwide. Such technologies provide languages, methods and tools for analysing engineered systems and systems-of-systems starting at a very early stage of design [3]. Experience suggests that model-based techniques help manage development risk, for example by reducing the number of prototypes required prior to system release. They also enable analytic approaches to the assurance of key system properties related to dependability and performance. Recent advances allow diverse engineering disciplines to integrate diverse models of cyber and physical system elements in simulation environments. These are beginning to deliver tool-supported analysis of such systems that deliver monitoring and control [4, 5]. However, the technologies of model-based engineering networked, smart systems with learning for optimisation and autonomy are not advanced, partly because of the separation of research and engineering practice in the range of disciplines involved [6]. It is in this context that we proposed our Shonan meeting.

3 Overview of Invited Talks

The meeting began with short personal introductions by each participant. We were grateful to Hans Vangheluwe and Judith Dahmann for giving keynote lectures, each focussing on one area of major technical significance to smart systems engineering. We summarise their talks below.

Co-simulation: Serving Multiple Masters

Hans Vangheluwe, University of Antwerp

It is essential to find new ways of enabling experts in different disciplines to

---

[4] See, for example `http://into-cps.org/`

[5] See, for example `https://en.wikipedia.org/wiki/Big_data`

[6] See, for example `https://en.wikipedia.org/wiki/Machine_learning`

[7] See for example the predictions for autonomous cars in “Autonomous Vehicle Implementation Predictions Implications for Transport Planning”, Todd Litman Victoria Transport Policy Institute, November 2016, `http://www.vtpi.org/avip.pdf`

collaborate more efficiently on the development of complex systems. One potential solution is to use a heterogeneous model-based approach in which different teams can produce models and analyse models using their own established notations and tools, but in addition coupling their models for simulation. Such “co-simulation” permits the study of the system’s global behaviour which arises from the interaction of the constituent modelled systems. Due to its potential, co-simulation is being studied in many different contexts, but with limited sharing of findings. This talk aimed to summarise, bridge and clarify future research in this multidisciplinary area. The main concepts in co-simulation were introduced, and currently open challenges were surveyed [7, 8, 9].

Model-Based Systems of and Smart Systems: State of the Art and Challenges

Judith Dahmann, The MITRE Corporation

A System of Systems (SoS) is a group of systems that arises from the integration of independent and useful systems into a larger system that delivers unique capabilities arising from the interactions between its constituent systems [10, 11, 12, 3]. The engineering of SoSs presents significant challenges. SoSs are particularly susceptible to interactions between constituent systems and with humans who may behave in unpredictable ways, leading to unanticipated emergent behaviour. Viewing smartness in terms of the NIST ALFUS contextual autonomous capabilities model[8], the inclusion into SoSs of smart constituents which introduce new and unexpected behaviours may have unanticipated effects on other systems or on the SoS as a whole, potentially introducing risks to resource utilisation, safety, security and trust.

4 Discussions

As each participant gave their short introductory presentation, all participants were invited to identify trends and drivers affecting the development of smart CPS; needs and requirements for methods and tools for model-based engineering of smart CPS; research themes that have the potential to meet needs and requirements; the technology and capabilities that have the potential to enable the research themes; and the enablers or barriers affecting the delivery of the technology or capabilities. Following collation of these initial ideas, results were clustered into four thematic groups

- Dynamic evolution and model-based systems engineering (MBSE) for smart products
- The role of machine learning in smart systems
- Tool support for MBSE of smart products
- The role of the human in the loop

Participants in each group worked over two days to characterise their views of the state of current practice, and discuss their priorities for ways ahead in each area. These discussions are summarised in Sections 4.1 to 4.4 below.

---

[8] `https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=823618`

4.1 Dynamic Evolution and MBSE for Smart Products

4.1.1 State of the Practice

Evolution means dealing with updates in the components, elements and environment of a smart system. Within the state of practice, we see emerging support for traceability in evolution, recording dependencies between design artefacts, corresponding regression testing, and (to a limited degree) argumentation and proofs. Support (configuration-based) is beginning to appear for product lines. However, interface descriptions are not precisely specified, in particular, in an SoS context. Companies are trying to do incremental dependability analysis when models change, but impact analysis is limited. The abstractions used in formal modelling languages may ease change, compared to, for example, modifying intricate code, although explicit features for evolution management are not generally foundi n formal notations or tools.

There is a general lack of support for managing dependability evidence throughout evolution. Model-based methods and tools in practice provide only limited support for analysing and predicting the impact of change during evolution: one can envisage the use of a publish /subscribe mechanism established so that notices of updates can be released to dependent constituent systems prior to deployment.

4.1.2 Ways Ahead

There is a real opportunity to enhance model management for safety/assurance cases and for change impact analysis. Future work should increase the continuous-integration capabilities of verification tools. Contract based approaches can help mitigate the challenges of managing evolution in the presence of multiple stakeholders. In the context of digital twins, it is important to keep model and system synchronised, and contractual approaches may have some value here. The idea of models at runtime (see [13]) is relevant, as is the investigation of evolution in relation to autonomy of systems that include machine learning.

4.2 Integration of Machine Learning in Smart Systems

This group focused on software that generates an output in an inductive data-driven way (c.f. a classical deductive approach embedded in a conventional program).

4.2.1 State of Practice

The group felt there was little or no practice in model-based design/engineering that includes ML, and that, although there are effectively some code libraries available, and reports on illustrative applications, the predominance of “buzzwords” in this field disguises a lack of well established practice.

4.2.2 Ways Ahead

In model-based engineering for smart systems, we need a contract or interface that defines the ML component in terms of its responsibilities and impact on other system elements. There is the potential to apply formal or deductive approaches to state expectations or bounds on outputs. The consensus was we do not even have typical use cases for ML in a system or SoS, and so a review of the state of knowledge and practice would be highly recommended. We are at the road mapping stage. Work on models and languages for contract/interface specification and bounds/expectations are potentially valuable.

4.3 Tool Support

4.3.1 State of Practice

There is a wide variety of tools for specific contexts such as Matlab/Simulink, SysML tools for architectural modelling, test and code generation, requirements and traceability management, and compliance evidence management. Heterogeneous tool chains are emerging in practice, but the group was unsure whether we have capable tools for reasoning across radically different abstraction levels. The gap between the logic and the presentable layer is very great, with formal techniques “under the hood”.

The group took the view that, broadly speaking the capability of MBSE tools to handle components, products, connected products, product systems and SoSs tends to diminish as one seeks increasing levels of control, optimisation and autonomy (Figure 1).

The experience of the participants was that tools are largely used within disciplinary silos. Given the cross-disciplinary character of SoS-level properties, interoperability was seen as an imperative, from the level of semantics underpinning tools, up to methods and guidelines. For example, a move towards multi-modelling requires solutions to reasoning fully about stability. Even the most comprehensive tool sets have to integrate successfully with other individual tools, and “do it yourself” integration remains necessary, especially when working across abstraction levels. In this context, the closed character of many tools was seen as problematic, although experience in other fields such as image rendering suggests that this may improve over time.

Moving from tools to the processes that they support, the participants felt that current product lifecycle models tend to be document-based, rather than model-based. In a document-based approach, engineering information is dispersed among a wide range of artefacts including plans, analysis reports, requirements descriptions, etc. In a model-based approach a major part of the same information is captured in a set of system models. The models themselves are primary outputs of engineering activities [14].

Skills are inseparable from tools, and the level of skills on semantic foundations and theory required to gain the maximum benefit from advanced tools has to be balanced with the expressiveness and analytic power that they enable. Successful exploitation of tool chains requires discipline on the part of users.

Current tools lack support for the transition between the simulated design environment and the operational environment of the deployed system as built. Support for managing the connection between models and analysis outcome from multiple abstraction levels is also seen as lacking. These factors contribute to the challenge of integrating into environments supporting continuous development.

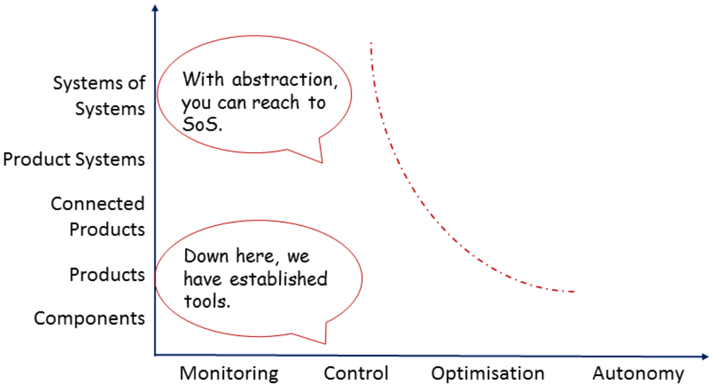


Figure 1: Current MBSE tool capabilities

4.3.2 Ways Ahead

The group identified several key directions towards the achievement of more interoperable cross-disciplinary tool chains:

- Integrated model sensitivity analysis tools would help people inexperienced with model-based analysis to gain value from such approaches.
- Need to bridge the gap between different disciplinary approaches by offering multiple ways of formulating the same outcome (imperative, functional, symbolic, numeric).
- Could one develop "plug-ins" that support analysis of emerging properties like dependability and security?
- Focus on the process of model production and maintenance and not solely on the semantic richness of models. For example, support "round-trip" engineering processes that synchronise diverse models as they change.
- Incremental assurance methods should help address the product line issues.
- There is a need for good tools to support model refactoring. There will be an increasing need to address legacy *models*, rather than just legacy *code*.
- There is a current need for an assisted refinement methodology.
- Could test automation methods be valuable? Differential testing for models, e.g., in dealing with models that describe phenomena at different levels of fidelity, could have consequences for impact analysis.
- There are exciting opportunities to deploy advances in data analytics to evaluate simulation models, but care is required because in some situations there is a lack of control over the data itself (cf. carefully designed experiments used to determine design parameters).

- Experts in different disciplines need to have a collaborative relationship, and this starts at the ontological level in keeping consistency in co-design. You need frequent holistic system analysis for an agile development.
- Pay attention to the need to deliver early representative models for domain experts for validation.
- Tools interoperability standards are going to affect this field very significantly.
- Maintain links to existing practice regardless of how bad it is! Research and innovation in the field can be dominated by scientists talking to and competing with other scientists and setting the standards for success. This needs to be constantly calibrated agains the practitioner base. For example, while static verification performance is the subject of regular international competitions, tools have to bring better value in general (e.g. they need to be usable by practitioners).

4.4 The Role of the Human in the Loop

4.4.1 State of Practice

Participants singled out advances in the aviation, nuclear and automotive sectors. The first two are seen as highly controlled environments. Passing of control is comparatively well understood in these situations, and hard constraints exist on interactions between human and technical system elements. In the automotive sector, there are good basic models of human reactions and system interaction with physical consequences, e.g. in managing acceleration. There is also experience with human-in-the-loop design validation (e.g. around user perceptions and feelings) via simulation, as well as a growing methodology of user-centred design. In spite of this, model-based methods and human aspects of smart (in the sense of increasingly autonomous) systems remain an art.

4.4.2 Ways Ahead

Participants would prioritise research in human and system interaction through a cooperation of cognitive and computer scientists targeting: the understanding of control and responsibility across human/system boundaries, the presentation of system information at a real range of abstraction levels. There are opportunities to develop improved models of individual and collective user behaviour through data analytics and let these influence design and operation via digital twin techniques. The group would recommend an examination of continuous training of technical and human system elements together (as a team) and in non-critical situations look at behavioural adaptation as a means to reduce initial training.

5 Working Groups

Following the discussions on the state of practice and research priorities, the participants selected three areas that could benefit from future collaborative work: evolution, machine learning and MBSE for smart systems. These formed the subjects of working groups over two days.

The evolution working group focussed on identifying and developing a case study that could be a catalyst to future collaborative work. The study concerns the planning and execution of agricultural operations and the description covers a domain model, data model, business logic and implementation which was cut down on an example that was just large enough to illustrate evolution challenges. There was a common interest in examining this study using multiple notations and teams. The case study description is included in this report in Appendix A.

The working group on Machine Learning focussed on the automotive domain and image processing in particular. It is expected that rudimentary image detection requires neural structures with around 3 layers of neural structure with order $10^2 - 10^3$ in the first and second layers, but many structures are possible. It is not possible to show the absence of faults but testing with scenario coverage and checking for misclassification is done. Explainable ML would be potentially necessary to help understand how decisions are made (e.g., in a driving context). There is preliminary work on trying to explain what goes on within a neural net, but it does not currently provide a solution to the level required for the kinds of scenario considered here, and it is not considered that this level will be reached in the next 5 years. Some researchers are trying to understand what happens inside nets by perturbing inputs, and adversarial and robust training methods migt be used. There is work on the verification of neural networks, but there is not yet an established way to certify ML approaches for critical systems.

The working group on MBSE for smart SoS focussed on characterising properties of smart SoS: the dimensions of the smart systems space. Starting from the characterisation in [2] (see Section 2), it was noted that the monitoring-control-optimisation-autonomy dimension would not probably not be a total order (optimisation not necessarily being a prerequisite for autonomy, for example). The group worked on a case study of an urban road management system working at a basic monitoring level: components (sensors) in products (smartphones) collect data about road conditions (such as potholes). These products are connected via a common app that allows analysis of the gathered data to reveal locations and severities of travel disruption caused by damaged roads. This product system can deliver information to a city-wide "dashboard". Across cities, diverse monitoring systems are coordinated into an SoS allowing collaboration over the efficient deployment of shared road mending services. In this largely monitoring system, current MBSE technology is able to make progress at gaining confidence in SoS-level properties (with some effort!). At the other (autonomy) end of the spectrum, the group considered a resilient smart energy grid delivering power from the integration of independently owned and managed product systems formed from connected infrastructure elements. A range of other characteristics of smart systems are worth considering, including the complexity of the task undertaken, the complexity of the environment, the proportion of interaction.

6 List of Participants

- Judith Dahmann, The MITRE Corporation, USA

- Gidon Ernst, National Institute of Informatics, Japan
- John S. Fitzgerald, Newcastle University, United Kingdom (Co-organiser)
- Constance Heitmeyer, Naval Research Laboratory, USA
- Fuyuki Ishikawa, National Institute of Informatics, Japan (Co-organiser)
- Alexandros Iosifidis, Aarhus University, Denmark (remotely connected) participated in discussions on Machine Learning
- Einar Broch Johnsen, University of Oslo, Norway
- Taro Kurita, Sony, Japan
- Peter Gorm Larsen, Aarhus University, Denmark
- Kenneth Lausdahl, Aarhus University, Denmark
- Mark Lawford, McMaster University, Canada
- Zhiming Liu, Southwest University, China
- Mike Nicolai, Siemens, Germany
- René S. Nilsson, Aarhus University, Department of Engineering / AGCO A/S, Denmark
- Tomohiro Oda, Software Research Associates, Inc., Japan
- Holger Pfeifer, fortiss, Germany
- Ken Pierce, Newcastle University, UK
- Sebastian Steinhorst, Technical University of Munich, Germany
- Anders Franz Terkelsen, CustomOffice ApS, Denmark
- Hans Vangheluwe, University of Antwerp/Flanders Make and McGill University, Belgium

7 Meeting Schedule

Check-in Day: December 3rd 2017 (Sun)

- Welcome Banquet

Day1: December 4th 2017 (Mon)

- **Goal:** Getting to know one another, identifying and prioritising trends, drivers, enablers and barriers in model-based Smart Systems Engineering
- **7:30 – 9:00:** Breakfast
- **9:00 – 9:45:** Session 1: Introduction Talks: 3 slides per participant (1 on personal background; 1 on current research; 1 giving the view on the Seminar goal)
- **9:45 – 10:15:** Tea Break
- **10:15 – 11:45:** Session 2: Introduction Talks cont?d
- **11:45 – 13:30:** Lunch
- **13:30 – 15:45:** Session 3: Clustering, prioritising of ideas
- **15:45 – 16:15:** Tea Break
- **16:15 – 17:45:** Session 4: Clustering, prioritising of ideas
- **18:00 – 19:30:** DinnerTalks and Discussions

Day2: December 5th 2017 (Tue)

- **Goal:** Assessment of the trends, drivers, needs, enablers and barriers to progressing the top areas identified in Day 1.
- **7:30 – 9:00:** Breakfast
- **9:00 – 9:45:** Session 1: Opening Plenary (review Day 1; arrangements for Day 2), possibly guest survey presentation
- **9:45 – 10:15:** Tea Break
- **10:15 – 11:45:** Session 2: Parallel Discussion Groups refine the priority topics identified in Day 1
- **11:45 – 13:30:** Lunch
- **13:30 – 15:45:** Session 3: Parallel Discussion Groups refine the priority topics identified in Day 1
- **15:45 – 16:15:** Tea Break
- **16:15 – 17:45:** Session 4: Rapporteurs report back from the group discussions
- **18:00 – 19:30:** Dinner

Day3: December 6th 2017 (Wed)

- **Goal:** Consolidation and identification of potential collaboration opportunities
- **7:30 – 9:00:** Breakfast
- **9:00 – 9:45:** Session 1: possibly guest survey presentation
- **9:45 – 10:15:** Tea Break
- **10:15 – 11:45:** Session 2:
- **11:45 – 13:30:** Lunch
- **13:30 – 18:00):** Excursion
- **18:00 – 19:30):** Banquet

Day4: December 7th 2017 (Thu)

- **Goal:** Plan future reporting and dissemination activities (maybe a book or extended report?)
- **7:30 – 9:00:** Breakfast
- **9:00 – 9:45:** Session 1: possibly guest survey presentation
- **9:45 – 10:15:** Tea Break
- **10:15 – 11:45:** Session 2:
- **11:45 – 14:00:** Lunch

8 Follow-up Activities

A theme issue of the International Journal on Software and Systems Modelling (SoSyM) has been approved on the subject "Model-Based Design for Smart Products and Systems" (see Appendix B). This will be edited by the three organisers of this Shonan event.

References

[1] O. Vermesan and P. Friess, eds., *Internet of Things – From Research and Innovation to Market Deployment.* River Publishers, 2014.

[2] M. Porter and J. Heppelmann, "How Smart, Connected Products are Transforming Competition," *Harvard Business Review*, November 2014.

[3] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Model-based engineering of systems of systems," *ACM Computing Surveys*, vol. 48, September 2015.

[4] J. Fitzgerald, P. G. Larsen, and M. Verhoef, eds., *Collaborative Design for Embedded Systems – Co-modelling and Co-simulation.* Springer, 2014.

[5] P. G. Larsen, J. Fitzgerald, J. Woodcock, P. Fritzson, J. Brauer, C. Kleijn, T. Lecomte, M. Pfeil, O. Green, S. Basagiannis, and A. Sadovykh, "Integrated Tool Chain for Model-based Design of Cyber-Physical Systems: The INTO-CPS Project," in *CPS Data Workshop*, (Vienna, Austria), April 2016.

[6] T. Blender, T. Buchner, B. Fernandez, B. Pichlmaier, and C. Schlegel, "Managing a mobile agricultural robot swarm for a seeding task," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pp. 6879–6886, IEEE, October 2016.

[7] H. L. Vangheluwe, J. de Lara, and P. J. Mosterman, "An introduction to multi-paradigm modelling and simulation," in *Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems)* (F. Barros and N. Giambiasi, eds.), (Lisboa, Portugal), pp. 9–20, April 2002.

[8] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: State of the art," tech. rep., feb 2017.

[9] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a Survey," *ACM Comput. Surv.* Accepted on January 11, 2018 for publication in ACM Computing Surveys.

[10] M. W. Maier, "Architecting Principles for Systems-of-Systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[11] J. Dahmann and K. Baldwin, "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," in *IEEE Systems Conference*, IEEE, April 2008.

[12] J. Dahmann, "Systems of Systems Pain Points," in *INCOSE International Symposium on Systems Engineering 2014*, april 2014.

[13] V. Braberman, N. D'Ippolito, J. Kramer, D. Sykes, and S. Uchitel, "Morph: A reference architecture for configuration and behaviour self-adaptation," pp. 9–16, 2015.

[14] INCOSE, "Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities, Version 4.0.," Tech. Rep. INCOSE-TP-2003-002-04, International Council on Systems Engineering (INCOSE), January 2015.

A Case Study on Dynamic Evolution in an SoS Context

Evolution scenarios in an SoS for Agricultural Contractors

At CustomOffice we help Danish agricultural contractors digitize their daily operations through the integration of various digital data suppliers and tools in to a coherent digital ecosystem that gets neatly presented to the users via customized apps for smartphones, tablets and internet browsers (web apps).

A CustomOffice solution consists of a per customer cloud environment running CustomOffice cloud services responsible for connecting to third party services as well as allowing third party services and devices to connect to it.

The CustomOffice cloud services gathers and processes relevant business and operations information and monitor events, presenting it to different users in different useful ways depending on their roles in the customers company. Information is presented to and gathered from the users via mobile apps; automatically as well as via manual user input. The CustomOffice apps all support offline operation, allowing data to be collected even when there is no internet connection. Once connection to the internet is restored, synchronisation of stored data and events takes place.

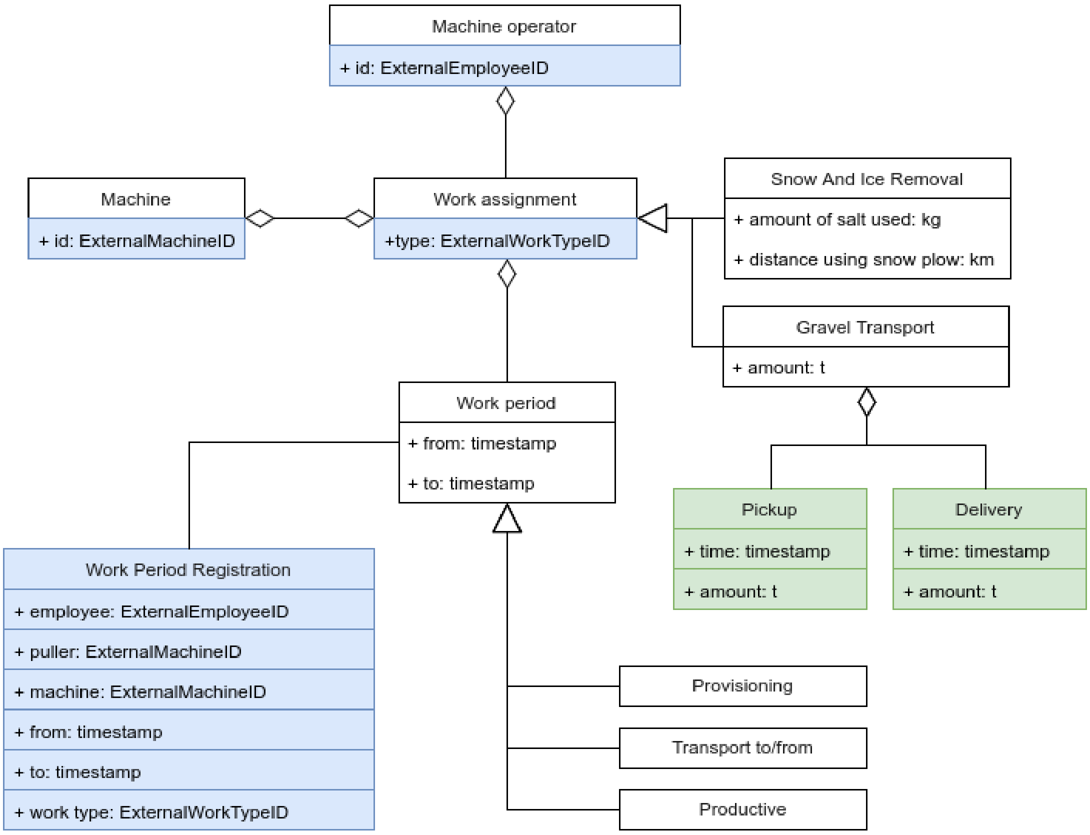
For a machine operator it is quite common to lose connection to internet several times during a work day. A primary concern for the machine operator is the ability continue his work, getting his work time and the results of the work registered correctly. In this domain the delay of data delivery is acceptable but the loss of data is not.

At CustomOffice, working as a digitisation consultant, we are continuously modelling each specific customer domain, business processes, as well as trying to understand and capture the underlying culture our customer's business. Most of our models are often quite informal, but are invaluable tools for communication and digging out all the latent knowledge inside the heads of managers and employees alike.

We see a great potential in bringing true formal modelling tools in to our domain as consultants. It can hopefully allow us to work more with models as our primary tool for exploration and simulation of company process improvements, as we today find ourselves depending solely on our logic reasoning and deductive abilities as humans to spot conflicts, nondeterminism and constraint violations within the semi-formal representations we have of our customers domain.

Example Domain Model

Here follows a minimalistic model, capturing a small subset of the domain of a machine operator working for an agricultural contractor. The coloured classes and attributes represent evolutions. That is, the model started out with only the white classes and attributes. The two evolution scenarios are explained in the following two sections.



Her follows a short description of each class:

- **Machine Operator**
  Instances represent the actual employee, operating the machines on the field.
- **Machine**
  Instances of this represent the various machines available to the machine operator for him to use in order to do his job. Examples: Tractor, plow, lorry, harvester.
- **Work assignment, Snow and Ice Removal, Gravel Transport**
  The actual job the machine operator is sent out to perform. Here we only consider to types of jobs: Snow and ice removal, and gravel transport.
- **Work period, Provisioning, Transport to/from, Productive**
  Each job can be broken down to in to a sequence of work periods, each with its specific purpose. Provisioning consist of gathering and preparing the machine needed for the job, and getting them ready for transportation to the customer. Then

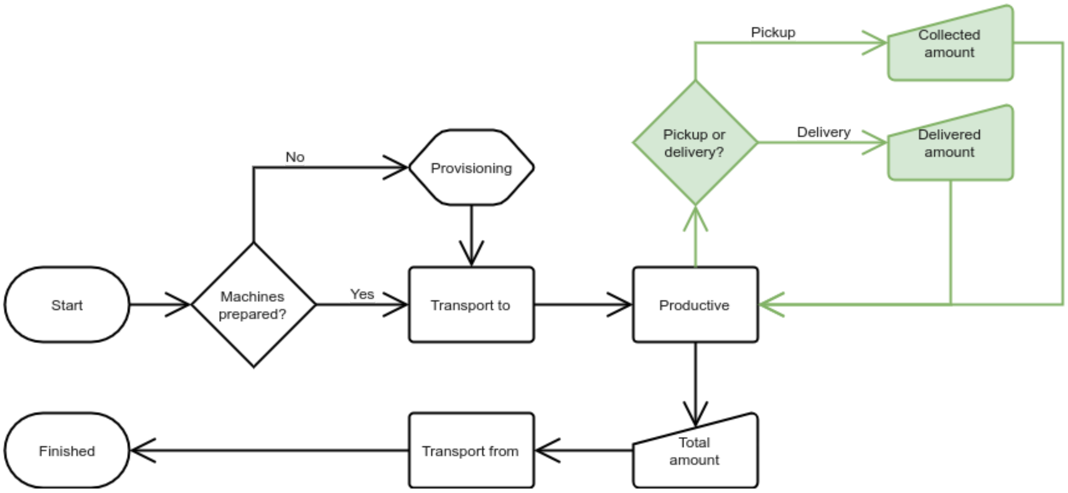
there is the actual transport to/from the work location and finally the period of productive (paid) work performed. Typically, in Denmark, only the actual work is paid. Provisioning and transport to/from is expected to paid indirectly through the hourly wage of the job itself. This is solely due to tradition but obviously makes it very important to minimize provisioning and transport time in order to maximize profits.

- **Work Period Registration**
  Data format required by a 3rd party service for calculating work assignment type, employee and machine cost and revenue based on a specific cost/revenue model used by many agricultural contractors.

Scenario 1 (Green): Simple evolution of capabilities by the addition of new information.

The flowchart show the sequence of steps a machine operator goes through when executing a Gravel Transport work assignment. We have omitted the nodes that represent the registration of work period start and end timestamps that are in fact being stored at the begging and end of each process-node shown.

If we ignore the green nodes/transitions then the flowchart tell us that after performing the productive part of a Gravel Transport work assignment the total amount of gravel moved has to be manually entered by the machine operator before he leaves the work location. This might be the first simple model we choose to digitize.



However, the completion of the work assignment often require many trips, to and from a pickup location and a delivery location. So at a later point in time, we decide together with our customer to support this finer level of detail and thus extend our model and flowchart, and in turn the mobile apps used by the machine operator, with the green nodes you see in both the domain model and flowchart.

Now imagine a more complete model, where constraints are added, like for instance that you should not be able to deliver more than you have picked up, or that the total amount registered before transport from the workplace begins must equal the sum of all delivery amounts. Such constraints may act as part of the requirement specification for the software apps and services being build to digitize this business process.

Challenges:

1. The apps are running in a highly distributed system. What happens if apps implementing the old model has to coexist alongside apps implementing the new?
2. What happens if a machine operator is performing a single load Gravel Transport is on his/her way from pickup to delivery as the app gets updated? Registering the amount delivered, without a sufficiently large registered amount picked up doesn't make much sense, does it? Depending on the data validation implemented it might not be allowed to do this, effectively deadlocking the app.

Scenario 2 (Blue): Capability deprivation caused by third party SoS participants.

AFT: This is a scenario that I didn't mention at Shonan but I find very interesting and the example is very real.

The Snow and Ice Removal work assignment specialisation shown in the domain model represent the task of a machine operator taking a tractor, adding a snow plow to the front and salt dispensing machine to back and then using this machine configuration to remove snow from streets and ice from streets. Snow is removed by lowering the snow plow in front and pushing it to the sides of the street, and the salt being dispensed at the back melts the ice that has formed on the street below the now removed snow.

It seemed natural to make a flexible model that supported all sorts of machine configurations, with various constraints specifying which machines can be combined and how, and which cannot.

However, at some point a 3rd party system used for cost/revenue calculations got introduced to the SoS. The Work Period Registration class in the domain model define the data needed to deliver registered work periods to this system 3rd party system. Notice that each registration only allow two machines to be associated with it. The puller and the machine. The puller, typically a tractor, responsible for dragging a productive machine like a plow, for instance. The reason for this puller-machine relationship in the 3rd party system is due to the cost/revenue model it implements, which isn't important here apart from the fact that it does not support a 3 machine configuration like the one used for Snow and Ice Removal work assignments.

Correct integration with such a 3rd party service might be unavoidable due to political or even legal reasons. However, in order to do so, the puller-machine relationship becomes a constraint that "infects" the entire SoS. It deprives the service implementing are more flexible machine configuration scheme of capabilities it inherently has.

B Thematic Issue of the Software and Systems Journal: Call for Papers

Springer

Call for Papers

Editors

John Fitzgerald
John.Fitzgerald@ncl.ac.uk
Newcastle University, UK

Peter Gorm Larsen
pgl@eng.au.dk
Aarhus University, Denmark

Fuyuki Ishikawa
f-ishikawa@nii.ac.jp
National Institute of Informatics, Japan

Editors-in-Chief

Jeff Gray
University of Alabama

Bernhard Rumpe
RWTH Aachen University

Important Dates

Intent to submit	31 May 2018
Paper submission	30 Sept 2018
Notification	1 Dec 2018

Software and Systems Modeling

Theme Issue:

Model-Based Design for Smart Products and Systems

A new generation of products and services is being enabled by, and is dependent on, networked computing technology. Such "smart" systems (and systems of systems) may link diverse and independent devices and systems using Internet of Things (IoT) technology, coupled with data analytics, machine learning and increased autonomy. Some of the most fundamental challenges relate to the ways in which model-based methods can enable engineers to work across traditional boundaries between disciplines and organizations to deliver analysis of and confidence in holistic global performance at the system or system-of-systems level.

Significant advances are being made in the foundations, methods and tools of collaborative model-based systems engineering, and are beginning to reach technology readiness levels that enable industry deployment. However, the integration of such diverse elements as autonomy, IoT and data analytics within heterogeneous multi-models and tool chains remains an open topic. The aim of this theme issue is to provide a resource that describes the state of knowledge and practice in model-based engineering for smart systems, and to outline a guide to the key challenges in this area.

The ***Journal of Software and Systems Modeling*** (SoSyM) invites original, high-quality submissions for its theme issue on "Model-Based Engineering of Smart Systems" focusing on topics related to the challenges in this field, including:

- **Collaborative model development:** semantic foundations, methods and tools for development of models across diverse groups, formalisms and organizations.
- **Analysis and Co-simulation:** approaches to multi-paradigm model construction and analysis, including design space exploration.
- **Integration of Data Analytics into MBSE:** approaches incorporating data analytics into a modelling context.
- **Experience reports:** project organization; methodologies and guidelines for model-based engineering of smart systems.

General Author Information

- Papers must be written in a scientifically rigorous manner with adequate references to related work.
- Submitted papers must not be simultaneously submitted in an extended form or in a shortened form to other journals or conferences. It is however possible to submit extended versions of previously published work if less than 75% of the content already appeared in a non-journal publication, or less than 40% in a journal publication. Please see the SoSyM Policy Statement on Plagiarism for further conditions.
- Submitted papers do not need to adhere to a particular format or page limit, but should be prepared using font "Times New Roman" with a font size no smaller than 11 pt, and with 1.5 line spacing. Please consult the SoSyM author information for submitting papers.
- Each paper will be reviewed by at least three reviewers.

Making a submission

- Communicate your intent to submit a paper by emailing the theme issue editors the following information before the Intent to Submit deadline: Title, Authors, and an Abstract.
- Possible submission formats are:
    - Word (.doc, without macros)
    - Rich Text Format (.rtf)
    - PostScript (.ps, special fonts must be embedded)
    - PDF (saved as readable in version 5.0 or earlier)
- Submit your work using the online submission system manuscript central:
    - In step 1, select "Theme Section Paper" as the manuscript type.
    - In step 4, select "John Fitzgerald, Peter Larsen, and Fuyuki Ishikawa" (ti-smart-systems@sosym.org) as editor and press "Add Selected Editor(s)".
    - In step 5, make sure field "Cover Letter" includes the line: "Submission for Theme Issue on MBD for Smart Products and Systems".

Further information

If you have any questions or require additional information about this theme issue, please contact the editors.