A Framework for Cyber-physical System Tool-chain Development

A Service-oriented and Model-based Systems Engineering Approach

JINZHI LU

Doctoral Thesis in Machine Design
KTH Royal Institute of Technology, 2019
SE-100 44 Stockholm, Sweden

TRITA 2019:19
ISBN 978-91-7873-226-5

KTH School of Industrial Engineering and
Management, SE-10044 Stockholm, Sweden

Academic thesis, which with the approval of the Royal Institute of Technology, will be presented for public review in fulfillment of the requirements for a Doctorate of Technology in Machine Design. The public review is held in Gladan, Brinellvägen 85, Stockholm on 7th June 2019 at 13:00.

Print: US-AB, Stockholm

Abstract

The development of complex Cyber-physical Systems (CPS) requires tight interactions of projects, system components, stakeholders, data and models. These models and data support component design, which are implemented by various engineering tools used by stakeholders. Effective tool integration thus relies on development of tool-chains. Five common challenges have been identified as part of this work concerning the development of tool-chains to support Model-based System Engineering (MBSE) for complex CPS:

1. Models from different domains are represented by different syntaxes and seemingly similar syntax may have completely distinct semantics. Moreover, such models and their views have completely different purposes. A unified formalism, therefore, does not exist to accommodate all system artifacts from all constituent models.
2. Modeling tools are developed by different tool suppliers and hence the interoperability of these tools may be limited, because the interfaces may not be fully open.
3. The interoperability limitation is especially prohibitive for co-simulations across multiple simulation tools.
4. Tool-chains must support design automation of product development workflows as adopted by different business units.
5. Users and developers of the MBSE tool-chain must have evaluation criteria to judge the effectiveness of different tool-chains and workflows.

To alleviate these challenges, this thesis proposes a novel Model-based System Engineering (MBSE) framework, called SPIRIT, to support model and data integration and tool-chain development. The framework contributes to four main phases for CPS tool-chain development, namely concept, design, early evaluation and operation phases. For the concept phase, the framework exploits the systems thinking approach to develop novel concepts of MBSE tool-chains. For the design phase, the framework adopts a service-oriented approach to construct tool-chains from the perspectives of social networks, process, information-service-infrastructure, and technology. For the early evaluation phase, quantitative metrics are defined to measure, (i) the MBSE capabilities of tools within a tool-chain, and (ii) the interoperability of the tool-chain. For the operation phase, several MBSE tool-chain prototypes are developed to support product development. An advantage of the new framework is to support tool-chain development using systems thinking and considering integrations of several open standards, including: 1) a domain-specific modeling (DSM) approach based on the Graph-Object-Property-Point-Role-Relationship meta-meta model; 2) ontology design based on the Web Ontology Language; 3) co-simulation using the High-Level Architecture and the Functional Mock-up Interface; 4) model-driven process management using BPMN; 5) tool-integration based on Open Services for Lifecycle Collaboration; and 6) value selections of design parameters based on an automated decision-making algorithm.

The effectiveness of the novel MBSE framework is investigated and verified by three case studies. Three main contributions are concluded from this thesis:

1. Benefits and challenges of MBSE tool-chains in industry are identified through a questionnaire survey and literature review. The results and the use of a systems thinking approach led to the development of a conceptual architectural model aiming to support MBSE tool-chain formalisms.
2. The SPIRIT framework is defined to provide support for MBSE tool-chain development. The framework has the following properties: architecting by DSM, integrating tools and models via open standards, enabling automatic co-simulations, and supporting design automation.
3. A DSM approach supporting visualization and Bayesian network analysis is presented to support MBSE tool-chain assessment. Quantitative metrics are defined to evaluate the effectiveness of MBSE tool-chains.

Keywords: Cyber-physical system development; Model-based systems engineering; Tool-chain; Domain-specific modeling; Tool-integration; Co-simulation; Systems thinking

Sammanfattning

Utvecklingen av komplexa Cyber-Fysiska System (CPS) kräver nära interaktion mellan projekt, systemkomponenter, intressenter, data och modeller. Dessa modeller och data stödjer komponentdesign, vilken skapas av olika intressenter med hjälp av olika ingenjörsverktyg. Effektiv integrering av verktyg är därför beroende av utveckling av verktygskedjor. Fem vanliga utmaningar identifieras i den här avhandlingen avseende utvecklingen av verktygskedjor som stöd för modellbaserad systemutveckling (MBSE) för komplexa CPS:

1. Modeller från olika domäner använder olika syntax och även om dessa kan likna varandra så kan de ha helt distinkt semantik. Dessutom kan sådana modeller och deras perspektiv tjäna vitt skilda syften. En enhetlig formalism som tillgodoser alla systemartefakter från alla ingående systemmodeller existerar därför inte.
2. Modelleringsverktyg utvecklas av olika verktygsutvecklare och därför är interoperabiliteten mellan dessa verktyg ofta begränsad och deras gränssnitt oklara.
3. Begränsningen i interoperabilitet är speciellt problematisk för samsimulering som involverar flera olika simuleringsverktyg.
4. Verktygskedjor måste stödja automation av produktutvecklingens arbetsflöde så som den implementerats av olika delar av företag.
5. Användaree och utvecklaree av MBSE-verktygskedjor måste ha utvärderings-kriterier for att bedöma effektiviteten hos olika verktygskedjor och arbetsflöden.

Denna avhandling föreslår ett nytt MBSE-ramverk, kallat SPIRIT, för att hantera dessa utmaningar genom att stödja modell och dataintegration, och utveckling av verktygskedjor. Ramverket bidrar till fyra huvudfaser för utveckling av CPS verktygskedjor, nämligen faserna för **koncept, design, tidig utvärdering** samt **användning**. Under konceptfasen utnyttjar ramverket *systemtänkande* (Eng: ”Systems Thinking”) till utvecklingen av nya koncept för MBSE-verktygskedjor. Under designfasen så tillämpar ramverket en tjänsteorienterad ansats för konstruktion av verktygskedjor baserat på perspektiven runt sociala nätverk, processer, information - tjänster – infrastruktur, och teknologi. Under den tidiga utvärderingsfasen så definieras kvantitativa mätetal för att mäta, (i) MBSE funktionaliteter av verktyg inom en verktygskedja och, (ii) verktygskedjans interoperabilitet. Under användningsfasen så har flera prototyper av MBSE-verktygskedjor utvecklats för att stödja produktutveckling. En fördel med det nya ramverket är att det stödjer utvecklingen av verktygskedjor genom användningen av systemtänkande och med beaktande av integrationen av flera öppna standarder, inkluderande: 1) domän-specifik modellering (DSM) baserad på ”*Graph-Object-Property-Point-Role-Relationship meta-meta model*”; 2) ontologi-design baserad på ”*Web Ontology Language*”; 3) samsimulering genom användningen av ”*High-Level Architecture*” och ”*Functional Mock-up Interface*”; 4) modell-driven process-styrning genom ”*BPMN*”; 5) verktygsintegration baserad på ”*Open Services for Lifecycle collaboration*”; och 6) val av värden för designparametrar baserat på en algoritm för automatiserat beslutstagande.

Effektiviteten av detta nya MBSE-ramverk undersöks och verifieras av tre fallstudier. Sammanfattningsvis omfattar denna avhandling tre huvudbidrag:

1. Fördelar och utmaningar med MBSE-verktygskedjor i industrin identifieras genom en enkätundersökning och litteraturstudie. Resultaten och tillämpningen av systemtänkande har lett till utvecklingen av en konceptuell arkitektur-modell för att stödja formalisering av MBSE-verktygskedjor.
2. SPIRIT-ramverket är framtaget för att tillhandahålla stöd för utvecklingen av MBSE verktygs-kedjor. Ramverket har följande egenskaper: arkitekturutveckling genom användande av DSM, integration av verktyg och modeller genom öppna standarder, möjliggörande av automatisk samsimulering, och stöd för design-automation.
3. Ett DSM-tillvägagångssätt som stödjer visualisering och Bayesiansk nätverksanalys presenteras som ett stöd för utvärdering av MBSE verktygs-kedjor. Kvantitativa mätetal definieras för utvärderingen av effektiviteten hos MBSE verktygskedjor.

Terminology

Bayesian network models - BN Models: Probabilistic graphical models that represent a set of variables with their conditional dependencies in directed acyclic graphs [1].

Co-simulation: An enabling simulation technique, where global simulations of a coupled system can be achieved by composing the simulations of its parts [2].

Consistency: An absence of contradictions or when something at one place contradicts with something at another place [3].

Commercial Off-The-Shelf - COTS: Products that are commercially available and can be bought "as is" [4].

Complexity: A system characteristic making it difficult and sometimes even impossible, to accurately predict behavior over time, particularly, in terms of understanding all relevant interactions among system components in the defined system boundary [5].

Cyber-physical systems - CPS: Systems integrating cyber systems with physical processes which are constructed by embedded computers, physical plants, computations and networks [6].

Domain-specific Modeling - DSM: A modeling methodology that the various facets of systems involved with systematic uses of higher-level formalisms for developing systems and software [7].

Domain-specific Modeling Language - DSML: A set of higher-level abstractions which requires less efforts and fewer low-level details to specify a given system using models, compared with general-purpose modeling languages [7].

Dependency: The ability processing the dependent relationships of technical recourses, system information, system development processes and social networks [8].

Dependability: The ability to deliver a system function or software service that can justifiably be trusted [9].

Design automation: The use of computers to automate the design process [10].

Design parameter: System properties determining system performance, cost and risk trade-offs in the development (**Paper F**).

Design space of parameters: A set of selected parameter values which satisfy the defined requirements (**Paper F**).

Efficiency: The degree to which a system or component performs its designated functions with minimum consumption of time and technical resources [11].

Functional Mock-up Interface - FMI: A tool independent standard to support model exchange and co-simulation between dynamic models in different simulation tools using a combination of XML and executed files (either compiled in DLL/shared libraries or in source code) [12].

Functional Mock-up Unit - FMU: A zip file distributed in term of a model with a simulation slave or the coupling part of a tool [12].

Graph, Object, Port, Property, Relationship, Role - GOPPRR: A meta-meta model language including 6 key meta-meta models: Graph, Object, Port, Property, Relationship, Role [13].

High Level Architecture – HLA: A standard for distributed simulation, used when building a simulation for a larger purpose by combining (federating) several simulations [14].

Levels of Information Systems Interoperability – LISI: A maturity model for systems which should logically progress, or "mature," in order to improve their capabilities to interoperate [15].

Maintainability: The ability to undergo modifications and repairs [9].

Model-based Systems Engineering - MBSE: The formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases [16].

MBSE capability: Capabilities of tools supporting MBSE [17].

Middleware: A technique used in systems to enable heterogeneous processors, Operating Systems (OSs), software languages, and architectures to inter-operate in a transparent fashion [18].

Reliability: Continuity of correct system functions or software services [9].

Reusability: The capability using a technical resource (e.g., models, data, tools) by additional modules or works [11].

Integrity: Absence of improper system function or software service alterations [11].

Internet of Things – IoT: A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving internet and network developments. It will offer specific object identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability [19].

Interoperability: The capability of two or more components in tool-chains, e.g. tools, models can be exchanged and the capability to use the exchanged information in a heterogeneous network [11].

Key Performance Indicators – KPIs: Measurable values used to evaluate the success of an organization or employee in achieving performance objectives [20].

Lifecycle of CPS: Evolution of a CPS entity from conception through retirement [21].

Open Services for Lifecycle Collaboration - OSLC: An open community which defines a set of specifications enabling integration of software development using linked data [22].

Ontology: Formalisms of things to support complex system development in an MBSE toolchain (**Paper H**).

Quality: The degree to which a system, component, or process meets customer or user needs or expectations [10].

Run-time Infrastructure – RTI: A middleware that is required when implementing the High Level Architecture (HLA) [23].

Stakeholders: An individual, team, or organization (or classes thereof) with interests in, or concerns related to a system [21].

Service: The service concepts representing technical resources and activities that cause a state transformation of a technical resource, such as changing a parameter in a model [24].

Service orchestration: The operational and functional execution processes involved in delivering end-to-end services for supporting CPS development (**Paper H**).

Service-oriented framework: An environment supporting run-time infrastructure software implementing web services of software components [25].

Service Orchestration Templates - SoTs: Operational templates supporting service distribution and operation (**Paper H**).

Social factor: Major influences of the surrounding environments on an individual's behavior, such as culture and organization [26].

Simple protocol and RDF query Language - SPARQLS: One query language for defining a data access protocol and a standard query language used with the RDF data [27].

Semantic Web Rule Language - SWRL: One combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language [28].

System artifacts: Information about system characteristics and development generated during system development, which are used to describe development information, such as requirements, functions and architectures [26].

Systems thinking: A holistic approach to focus on the way that a system's constituent parts interrelate and how systems work over time and within the context of larger systems [29].

Tool operation service concepts: A collection of services for tool operations in a web-based process management system (**Paper H**).

Tool Maturity Capability Level - TMCL: A defined item refers to capabilities of MBSE capabilities and Levels of Information Systems Interoperability (LISI) in a tool (**Paper B)**.

Traceability: The capability establishing a relationship between technical recourses, system information, development processes and social networks [11].

Validation: A set of actions for checking the compliance of any element (a system element, a system, a document, a service, a task, a system requirement, etc.) with its purpose and functions [21].

View: A representation of a system from the perspective of a viewpoint [30].

Viewpoint: A specification of the conventions and rules for constructing and using a view for the purpose addressing a set of stakeholder concerns [30].

Verification: A set of activities that compares a system or system element against the required characteristics. This includes, but is not limited to, specified requirements, design description and the system itself. It refers to a situation if the system was built right [21].

Web Ontology Language – OWL: A family of knowledge representation languages for authoring ontologies [31].

Acknowledgment

This thesis and related work would not have been finished without the support of the people at KTH, I would thank as follows:

- Martin Törngren for providing the opportunity to work at KTH for me. He provides advice, feedback, ideas, encouragement and enthusiasm throughout the whole five-year Ph.D. study.
- Our team leader, Martin G. and his help for my five-year ph.d study.
- Dejiu Chen and Lei Feng for the advices during the Ph.D. study.
- Xinhai Zhang, Fredrik Asplund, Frederic Loiret, Andrii Berezovskyi, Naveen Mohan, Lifei Tang, Elias Flening, Yang Wang, Jad El-Khoury and other colleagues for the great working environment and discussions.
- Tong Liu, Didem Gürdür and Xin Tao for the collaborations with related works.
- Most thanks to Vicki Derbyshire for checking the English writing.

In this thesis, I would thank Dr. Fredrik Asplund for his wise comments. With his help, I understood a lot about how to think as a researcher. Moreover, my supervisor Prof. Martin Törngren, my co-supervisor associates Prof. Dejiu Chen and associate Prof. Lei Feng also gave me a lot of good comments for improving the thesis. In addition, Dr. Didem Gürdür, Elias Flening and Dr. Xinhai Zhang also gave me feedbacks about the thesis. What's more, thanks to my thesis quality reviewer, associate Prof. Ulf Sellgren, he gave me a lot of comments which promote my thesis a lot. Finally, I would say thanks to Vicki again for the language improvement. Her patience was a very important motivation for me to improve my thesis.

At the same time, thanks to my Chinese partners to support my research:

- Associate Prof. Guoxin Wang at Beijing Institute of Technology and Prof. Huisheng Zhang at Shanghai Jiao Tong University for collaborations with MBSE tool-chains.
- Associate Prof. Jian Wang at University of Electronic Science and Technology of China and Associate Prof. Jiqiang Wang at Nanjing Aeronautics and Astronautics University for collaborations in case studies.
- Thanks to staffs in Z.K. Feng Chao .Ltd to realize my ideas into industrial practices. Without their support, we cannot understand more about the natural of MBSE.

Thanks to the SESC-PSW Popular science workstation, Systems Engineering Society of China (CCOSE, China Council on Systems Engineering) for providing feedbacks about my research.

The most important is to thank to my dad and mum. Thank you for your patience and efforts on my five-year Ph.D. study（感谢感恩我的父母，在他们的支持和鼓励下，我努力完成我的学业）. Moreover, I must thank to CSC scholarship to support my 4-year Ph.D. study.

I hope to make use of my research to contribute to INCOSE and industry!

My best wishes are that MBSE can make industry better and greener.

Contents

Appended papers

Thesis is based on the following 8 papers, which are appended to this thesis.

Paper A J. Lu, Y. Wen, Q. Liu, D. Gürdür, and M. Törngren (2018), "MBSE Applicability Analysis in Chinese Industry", 28th Annual INCOSE International Symposium, 28: 1037-1051.

This paper is appended to understand the current MBSE applicability in Chinese industry which provides important clues about MBSE tool-chain development. This paper contributes to identify the industrial needs, usages, advantages, barriers, concerns and trends of MBSE and MBSE tool-chains using a questionnaire survey.

M. Törngren and D. Gürdür provided feedback. Q. Liu and Y. Wen supported investigation in the Chinese industry. J. Lu developed the questionnaire, analyzed the results and wrote the paper.

Paper B J. Lu, D. Chen, J. Wang and M. Törngren (2018), "Towards A Service-oriented Framework for MBSE Tool-chain Development", 13th IEEE Annual Conference on System of Systems Engineering (SoSE), Paris, 2018, pp. 568-575.

This paper is appended to clarify the SPIRIT framework for MBSE tool-chain development. This framework enables tool-chain developers to consider their viewpoints for tool-chain development using a systems thinking approach. Moreover, it provides one solution for constructing CPS tool-chains based on a service-oriented and Model-based Systems Engineering approach.

M. Törngren and J. Wang provided feedback. D. Chen provided some suggestions about this framework. J. Lu developed this framework and wrote the paper.

Paper C J. Lu, X. Wang, X. Tao, J. Wang and M. Törngren (under review), "A Domain-specific Modeling Approach Supporting Tool-chain Analysis", journal article

This paper is appended to clarify how one domain-specific modeling (DSM) approach to formalize and assess MBSE tool-chains for early evaluations before prototyping. It proposes a DSM approach to formalize and assess data workflows, structures and tool capabilities of MBSE tool-chains. Then the tool capabilities are evaluated using qualitative and quantitative analyses.

X. Wang, J. Wang and M. Törngren provided feedback. X. Tao contributed to the mathematic descriptions on Bayesian network models. J. Lu developed the models, assessed MBSE tool-chains and wrote the paper.

Paper D J. Lu, J. Wang, D. Chen, J. Wang, M. Törngren (2018), "A Service-oriented Tool-chain for Model-based System Engineering of Aero-engines", IEEE ACCESS (IF:3.557).

This paper is appended to clarify the first MBSE tool-chain developed based on the SPIRIT framework. The tool-chain is used to support co-design of aero-engine performance analysis for integrated verification.

M. Törngren, D. Chen and J. Wang provided feedback. J. Wang provided aero-engine simulation models for this case study. Jinzhi Lu developed the tool-chain prototype and wrote the paper.

Paper E J. Lu, D. Chen, J. Wang and M. Törngren (under review), "An MBSE Tool-chain Supporting Automated Co-simulation of Cyber-physical Systems", journal article

This paper is appended to clarify how the second MBSE tool-chain developed based on the SPIRIT framework supports automated co-simulations based on HLA and FMI. The MBSE tool-chain support automated code generations for manipulating co-simulations between FMUs through HLA RTI.

D. Chen, J. Wang and M. Törngren provided feedback. J. Lu developed the prototype and wrote the paper.

Paper F J. Lu, D. Chen, X. Tao, G. Wang, M. Törngren (under review), "A Model-based Systems Engineering Tool-chain Supporting Internet of Things Development", journal article

This paper is appended to introduce the third MBSE tool-chain developed based on the SPIRIT framework supporting automated value selections of design parameters for verifying an auto-braking system. The automated value selections enable to define the parameter spaces for co-simulation in the later phases based on the simulation results obtained from simplified models in the early phases.

G. Wang, M. Törngren and X. Tao provided feedback. D. Chen provided one use case from his SAFER project. J. Lu developed the models, the tool-chain prototype and wrote the paper.

Paper G J. Lu, G. Wang, J. Guo, H. Wang, H. Zhang, M. Törngren (2019), "General Modeling Language Supporting Model-based Systems Engineering Formalisms", 29th Annual INCOSE International Symposium.

This paper is appended to introduce a DSM approach based on GOPPRR meta-meta models. It provides one textual language based on GOPPRR meta-meta models for supporting MBSE formalisms.

J. Lu, J. Guo and H.Wang contributed to the ideas about the *Karma* language, G. Wang and M. Törngren provided feedback. H. Zhang led the development team for tool prototype development.

Paper H J. Lu, G. Wang and M. Törngren (2019), "Design Ontology in a Case Study for Co-simulation in a Model-based Systems Engineering Tool-chain", IEEE Systems Journal.

This paper is appended to introduce one ontology-based approach for service orchestration development. The approach supports to definitions of service orchestrations for automated co-simulations in one MBSE tool-chain based on Web Ontology Language (OWL).

M. Törngren provided feedback. G. Wang provided several key concepts related to ontology. Jinzhi designed the ontology and wrote the paper.

Additional Related Papers

1. Jinzhi Lu (2019), “State-of-the-art of MBSE tool-chains”, technical report, KTH.
2. J. Chen, Z. Hu, J. Lu*, H. Zhang*, M. Törngren (2019), “An Open Source Lifecycle Collaboration Approach Supporting Internet of Things System Development”, 14th Anuual System of Systems Engineering Conference.
3. J. Guo, G. Wang, J. Lu *, S. Sun and H. Zhang (2019), “A General Modeling Language Supporting Architecture driven and Code Generation (Part 2)”, 29th Annual INCOSE International Symposium.
4. H. Wang, G. Wang, J. Lu*, C. Ma (2019), “Ontology supporting Model-based Systems Engineering based on a GOPPRR approach”, WorldCist'19 - 7th World Conference on Information Systems and Technologies.
5. M. Lian, J. Wang, J. Lu (2018), “A New Hardware Logic Circuit for Evaluating Chip Multi-Processor Security”, Conference on Instrumentation & Measurement, Computer, Communication and Control.
6. B. Hao, J. Lu*, J. Li, G. Wang, X. Lan, J. Chen, X. Wu (2019), “Domain-specific Modeling Approach Supporting Model-based Systems Engineering in Aero-engine Development”, MBSE Special Issue, Science & Technology Review
7. J. Wang*, J. Lu*, Z. Chen (2018), “A Fault Propagation Model for Un-reconvergence Path Netlist in FPGA”, conference paper, 5th International Conference on Information Science and Control Engineering (ICISCE).
8. J. Wang*, J. Lu*, Z. Chen, S. Guo and Y. Li (2018), “A Thermal-balance-oriented Task Mapping Algorithm for CMPs”, The 8th International Conference on Information Communication and Management.
9. Z. Liu*, M. Yang*, Q. Chen, Q. Du, N. Xia, J. Lu* (2018), “An MBSE Tool to Support Architecture Design for Spacecraft Electrical Power System”, 28th Annual INCOSE International Symposium.
10. J. Wang*, H. Li*, J. Lu*, K. Li, H. Li, L. Yang and Y. Li (2017), “A New PSO-based Layout Method for GNSS Pseudolite System”, IEEE ICIT, International Conference on Industrial Technology (best presentation).
11. J. Lu, D. Gürdür, D. Chen, J. Wang and M. Törngren (2018), “Empirical-Evolution of Frameworks Supporting Co-simulation Tool-Chain Development”, WorldCist'18 - 6th World Conference on Information Systems and Technologies. https://link.springer.com/chapter/10.1007/978-3-319-77703-0_80.
12. J. Lu, D. Chen, M. Törngren, F. Loiret (2016), “A Model-driven and Tool-integration Framework for Whole Vehicle Co-simulation Environments”, 8th European Congress on Embedded Real Time Software and Systems, Jan 2016, Toulouse, France. https://hal.archives-ouvertes.fr/hal-01280473/.
13. J. Lu, D. Chen, J. Wang and M. Törngren (2017), “A Tool Integration Language to Formalize Co-simulation Tool-chains for Cyber-physical System (CPS)”, Workshop on Formal Co-Simulation of Cyber-Physical Systems in SEFM2017.

14. J. Lu, D. Chen, D. Gürdür, and M. Törngren (2017), "An Investigation of Functionalities of Future Tool-chain for Aerospace Industry", INCOSE International Symposium, 27: 1408–1422.
15. J. Lu, J. Ding, F. Zhou, X. Gong (2016), "Research of Tool-Coupling Based Electro-hydraulic System Development Method", In: Proceedings of the 6th International Asia Conference on Industrial Engineering and Management Innovation. Atlantis Press, Paris
16. D. Zhang, J. Lu, L. Wang, J. Li (2016), "Research of Model-based Aero-engine Control System Design Structure and Workflow", In Procedia Engineering, Volume 99, 2015, Pages 788-794, ISSN 1877-7058.
17. J. Lu, D. Chen, J. Wang, W. Li (2016), "An Investigation of Model-Based Design Framework for Aero-Engine Control Systems", In: Proceedings of the 2015 Chinese Intelligent Systems Conference. Lecture Notes in Electrical Engineering. Springer, Berlin, Heidelberg

1 Introduction

This thesis proposes and evaluates a framework for concept, design, early evaluation and operation of Cyber-physical Systems (CPS) tool-chain development using a service-oriented and Model-based Systems Engineering (MBSE) approach. MBSE is one advanced technology to support systems engineering using models (details introduced in Section 1.1.2). CPS development is typically supported by a large number of isolated design, modeling and simulation tools which can lead to difficulties in creating a seamless engineering environment linking development processes to system artifacts. MBSE tool-chains are one solution for supporting system engineering with an engineering development environment, for instance including formalisms for system artifacts and development (a description of something using models), tool integration, co-simulation and design automation.

This chapter aims to provide an overview of MBSE and MBSE tool-chains to motivate the research presented later in this thesis. The details are structured as follows: the background of CPS development is briefly introduced (Section 1.1); the challenges of CPS development and MBSE tool-chain development are illustrated and the stakeholders related to MBSE tool-chains are identified and analyzed (Section 1.2); the goals of the thesis are elicited from the challenges to CPS development and MBSE tool-chain development and a number of associated research questions defined (Section 1.3); delimitations and the scope of the thesis are defined (Section 1.4); finally the outline and reading guide of the thesis is described (Section 1.5).

1.1 Background

This section briefly introduces basic important concepts of CPS development, MBSE, and MBSE tool-chains. This prepares for the next section, which outlines associated challenges. Specifically, Section 1.1.2 describes objectives associated with the future of MBSE that this thesis focuses on.

1.1.1 CPS Development

CPS refers to integration of computation with physical processes [6]. This following definition was part of the CPS initiative: *"The integration of physical systems and processes with networked computing has led to the emergence of a new generation of engineered systems: Cyber-Physical Systems (CPS). Such systems use computation and communication deeply embedded in and interacting with physical processes to add new capabilities to physical systems. These CPS range from minuscule (pace makers) to large-scale (the national power-grid)."* [32].

CPS consists of elements from several domains, such as software, hardware and mechanical systems. During CPS development various views are frequently used to capture this heterogeneous nature from a system perspective. Each view expresses the system from one perspective identifying any concerns with respect to its governing viewpoints [30]. The view can include models for formalizing systems, system development, project and organizations. Such models are supported by various tools for design, simulation, analysis and testing. This is frequently based on engineers from different domains using their own specific tools. Each tool provides data and models to formalize related views for system development.

As shown in ISO 15288 [21], CPS development is typically implemented based on the standardized development processes, such as the V model. When tailoring, defining and reusing the entire development process, it is challenging for system engineers and developers to formalize and manage these processes. A particularly challenging aspect is the tool integration required to understand all relevant interactions among system components within the defined system boundary [5]. Especially as the complexity of CPS increases the involvement of many models, tools and data structures makes such integration difficult.

CPS development is often implemented through co-design and collaborative processes across domains. Thus, the consequences of such complex processes impact the efficiency of CPS development [33]. Change and configuration managements of technical resources, as well as traceability, improve support for large-scale concurrent engineering by contributing towards efficient communication among people and teams [34]. Furthermore, process control and monitoring implemented on IT systems are necessary to manage processes and stakeholders in large-scale concurrent engineering.

During CPS development, system parameters are designed and verified by simulations or prototype experiments, for instance parameters of Artificial Intelligence (AI) algorithms for cyber compositions [35]. Traditionally prototypes are used, but simulations can create knowledge and provide data for verification with less R&D cost. However, when considering all simulations required during CPS development, the total simulation execution time can be impractical as many parameters and all their permutations are needed to verify system performance. In order to support design automation of CPS, optimization based on simulations is a solution to deliver more efficient simulation-based analysis and verification [36]. Furthermore, automated implementations of repeated simulations are expected to save time during the optimization process by automated reviewing of results and decision-makings, which should promote design automation during CPS development.

1.1.2 Model-based Systems Engineering

A definition of Model-based Engineering (MBE) is proposed by National Defense Industrial Association (NDIA), "*Model-Based Engineering refers to an engineering approach that uses models as an integral part of the technical baseline including the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition lifecycle.*" [37]. The International Council on Systems Engineering (INCOSE) defines MBSE as "*is the formalized application of modeling to support system requirements, design, analysis, V&V activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases.*" [38]. MBSE includes MBE aspects associated with systems engineering, including requirement analysis, functional analysis, behavioral analysis, system architecture and testing.

MBSE formalizes system characteristics and system development using models [39]. The modeling techniques support structural analysis and design, data flow diagramming, process modeling, CAD, etc. When using such techniques in an appropriate way, the efficiency of CPS is promoted from several aspects using MBSE [38]: 1) communication between stakeholders, teams and organizations; 2) requirement formalisms, system design, integrity and consistency management of them; 3) change, reusability, configuration management; and 4) cost estimation and risk reduction when V&V efforts are defined in an early stage.

Current MBSE engineering practices have demonstrated significant value to stakeholders of CPS [40]. This thesis focuses on several general objectives which are associated with the future of MBSE:

- Formalization of CPS and CPS development from various views in a system perspective.
- Integration of MBSE tools with CPS development processes
- Data, model and tool interoperability
- Integrated verification
- CPS design automation

1.1.3 MBSE Tool-chains

Compared with other tool-chains, MBSE tool-chains are relatively new concepts to support MBSE for CPS development. One initial definition of **MBSE tool-chains** [40] is that they include two or more modeling, simulation and design tools that, when combined, can support and construct a system engineering workflow with the following features:

1). The tools in the workflow support system formalisms in the activities of product development, such as system requirements, design, analysis and V&V.
2). The tools in the workflow support the formalisms of non-functional aspects, such as process management, dependability and so on.

The previous two features aim to deal with one objective of MBSE referring to formalisms of CPS and CPS development from a system perspective.

3). In the workflow, the tools are required to support heterogeneous model integrations of integrated verification.

This feature is expected to support integrated verification based on simulation techniques across domains, such as co-simulation and Modelica [41].

4). The ontology in the tool-chain supports workflows and system engineering formalisms.
5). The tools are required to support data, knowledge, and information exchange and model transformation for integrating tools with CPS development process.
6). Support integration of existing engineering tools and system development platforms for project management and process management.

Feature 4 aims to integrate MBSE tools and development processes. Features 5-6 aim to promote model & tool interoperabilities and design automation when using MBSE. Considering MBSE tool-chains, tool integrations are always implemented by code generation and model transformation. In order to investigate these types of tool-chains, a framework, called SPIT is defined [42], with Social, Process, Information and Technical layers, inspired by H. Sillitto [43]. As shown in the literature reviews (**ARPaper 1,** short for **A**dditional **R**elated **P**aper) and questionnaire survey (**Paper A**), the tool-chains under SPIT are already widely used in industry and academia [36], and several of their limitations are summarized in the next section.

1.2 Problem Formulation

This section describes the basic challenges when using MBSE, particularly MBSE tool-chains under SPIT, to support CPS development (Section 1.2.1); typical stakeholders and their concerns related to MBSE tool-chain are analyzed (Section1.2.2); and the challenges of MBSE tool-chain development tool-chains are identified (Section 1.2.3). The challenges described and analyzed in Section 1.2.1 and 1.2.3 are structured according to the general objectives of MBSE described in Section 1.1.2.

1.2.1 Challenges of MBSE Beyond State-of-the-Art

During CPS development, tools are often dedicated to different engineering domains and used in specific tasks in the development process. However, such tasks are connected with each other, particularly during the iteration and recursion of processes. In order to support all the development activities, an engineering environment is needed to allow for complete integration of all related tools. Based on the objectives of MBSE enumerated in Section 1.1.2, some specific problems still need to be considered: 1) MBSE formalisms, particularly, domain-specific formalisms; 2) process control and monitoring with well managed traceability and configuration; 3) tool integration between heterogeneous tools; 4) co-simulation between heterogeneous models; 5) design automation to promote design efficiency. Each of these issues is described in a separate paragraph below.

The core of MBSE is the **use of models for supporting system development and system artifacts**. Different languages formalize CPS from different views in system modeling tools, such as Systems Modeling Language (SysML) and Unified Modeling Language (UML) [44]. However, domain-specific views are still challenging to formalize. Firstly, concrete syntax and abstract syntax of meta-models based on such modeling languages are limited - they cannot describe system artifacts of specific domains in a general way. For example, circuit elements are described by their own domain specifications, where SysML or UML is not able to describe them using the concrete syntax of electrical symbols [45]. Secondly, integrating meta-models is challenging during model development, which results in different views being described by isolated domain specific models [46]. Thirdly, traditional general modeling language specifications are difficult to learn resulting in strong resistance by users in industrial practice [47].

Process control and monitoring are not considered by current MBSE solutions, as there are typically gaps between real implementations and models for formalizing development processes [48]. Models based on languages, such as UML (activity diagram) and BPMN (Business Process Model and Notation), are used to formalize the development processes [49]. However, most of the real development processes are managed in Product Lifecycle Management (PLM) systems or in independent process management systems [50]. Because of such gaps, integration between MBSE models and the process management systems is required to promote the consistency of process definitions and real implementations. Furthermore, process control and monitoring in traditional process management systems do not allow stakeholders to access design information and technical resources related to MBSE, except for developing the configurations for defining traceabilities between the process management system and technical resources.

When a development process is implemented, **traceability and configuration management** of the technical resources (models, data and tools) are important to promote efficiency of CPS development. Traceability management refers to relationship management established among development process, design information and technical resources [51]. It supports stakeholders' access to their required technical resources through their tasks in the process management system. Configuration management refers to a systems engineering process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information [21]. Consistency between system models and technical resources is difficult to manage using traditional MBSE solutions, since there is not a flexible integration solution based on a unified ontology.

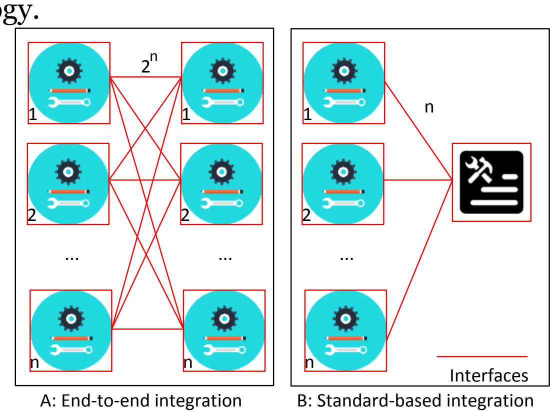

Fig. 1.1 Tool-integration in the MBSE tool-chains

During **tool integration**, different syntax and semantics of meta-models or meta-data make models and data accessibility difficult. Using a traditional approach, it takes a long time to develop interfaces between both tools. Therefore, the number of required interfaces, when n tools are used in an MBSE tool-chain as shown in Fig. 1.1, can in the worst case reach 2^n. The interface development implies a considerable R&D cost during MBSE tool-chain development [52]. Furthermore, tool APIs are not properly formalized which makes it challenging to notify and activate services and tools in a unified way. Compared with end-to-end integration and standard-based integration, if data across domains is not described in a unified way, it makes data and knowledge sharing between different stakeholders challenging and leads to low efficiency of tool integration.

Currently, specialized modeling and simulation tools are used to model domain-specific subsystems and predict their performance. However, these tools often lead to a large number of heterogeneous models which have different data structures and syntax. During the integrated verification phases, system developers often face a large challenge in integrating such heterogeneous models. Therefore **co-simulation** between heterogeneous models is one challenge to support integrated verification for CPS. As with tool-integration, standards are also required to construct a seamless integrated simulation environment for different specific simulation tools [2].

During CPS development, simulations are widely used to select value candidates of design parameters. However, a large number of repeated simulations always lead to a large number of time costs. Thus **design automation** is always considered to promote efficiency of such processes for which current MBSE solutions formalize optimizations based on simulations in order to execute simulations automatically [53]. However, such automated operations are executed on a case-by-case basis, which is difficult to integrate with the development processes. Moreover, obtaining data from iterative simulations leads to a high R&D cost for deciding the required parameters because of the amount of manual operations. Furthermore, traceability between the development process and selected parameter values decided by early phases is important because system developers require such parameter values to test the configured models in later phases. Therefore, automated decision-making algorithms are difficult to define for different cases, particularly for parameter value selections during CPS development.

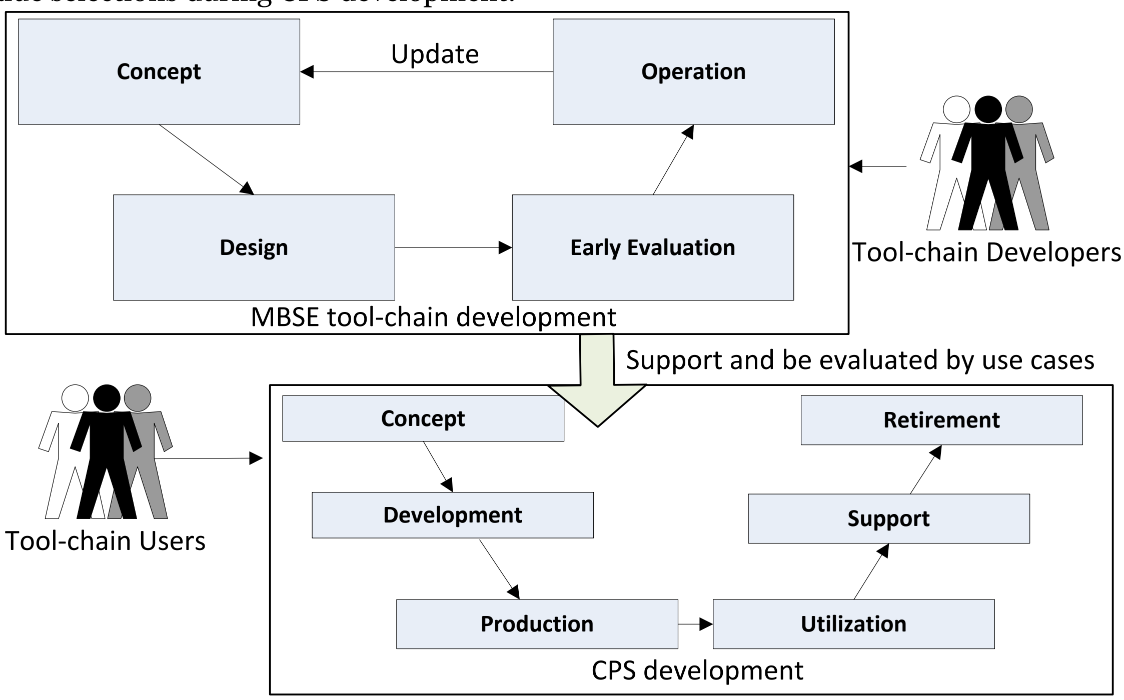

Fig. 1.2 MBSE tool-chain and CPS development

1.2.2 Stakeholders and Concerns

In order to analyze the stakeholders and their concerns, one MBSE tool-chain is considered as a "system". Based on the system architecture description [30], there are many important stakeholders of MBSE tool-chains, particularly, the tool-chain users (shortened to "users") and the tool-chain developers (shortened to "developers").

As shown in Fig. 1.2, **developers** refer to tool suppliers that develop tool-chains and IT managers who have a responsibility to develop tool-chains for users in their organization. Developers first define the design concepts of their MBSE tool-chains. After the tool-chains are designed using MBSE techniques, developers always assess MBSE tool-chains using their experiences for early evaluation [54]. Developers expect to develop the "correct concept" for the MBSE tool-chains and to evaluate their capabilities in the early phases in order to promote efficiency of CPS development and to reduce the R&D cost. Finally, when tool-chain prototypes are developed and implemented, the developers observe their effects to further refine their concepts.

Users are the CPS developers who use the tool-chains, for instance, systems engineers, project managers, domain engineers and testers [21]. They make use of MBSE tool-chains to develop CPS following the systems engineering lifecycle including concept, development, production, utilization, support and retirement phases. Through MBSE tool-chains implemented by users, tool-chains are evaluated by the effects on CPS development.

Users and developers expect their tool-chains to improve effectiveness and efficiency of CPS development [55]. Effectiveness refers to the capability of achieving the desired functions of the MBSE tool-chains [40]. The effectiveness is always considered as an important metric (success or not) when assessing the MBSE tool-chains. Efficiency here refers to the minimum consumption of time and technical resources during CPS development [11]. Some key performances related to efficiency are considered for MBSE tool-chains including: 1) data, model and tool interoperability [56]; 2) design automation, regarding the use of computers to automate the design process [10]; 3) human efforts in CPS development regarding how much manual work is required to implement MBSE tool-chains, such as mouse-clicks [57].

1.2.3 Challenges of MBSE Tool-chain Development

An MBSE tool-chain, aiming to promote efficiency of system development, is an IT solution to integrate related techniques to support CPS development. The tool-chain development faces both functional challenges (Challenge 1 – Challenge 4) and non-functional challenges (Challenge 5), which relate to the objectives enumerated in Section 1.1.2 and stakeholders' concerns of MBSE tool-chains:

1. ***Challenge 1: Formalisms of system artifacts and development processes supporting complexity management.***
   In general, the increasing complexity of CPS development stems from two sources [5]:
   1) Subjective sources referring to interpreting, understanding and anticipating design as a human activity. MBSE is proposed to promote understanding of the system characteristics and development for stakeholders [38]. However, it is difficult for stakeholders to make use of MBSE techniques to formalize the systems from system perspectives. This is due to: a) Users' lack of understanding of systems engineering theories and engineering knowledge; b) The lack of modeling techniques to formalize the system features and development in an integrated approach; c) The resistance of engineers to change their engineering cultures to a model-based systems engineering approach [47].
   2) Objective sources referring to work or information in an independent system of the people involved. Recently, MBSE tool-chains have been expected to integrate

and manage the development process and technical resources from different domains. However, it is difficult to describe the interrelationships between development processes and related technical resources which are used to support design automation in MSBE tool-chains [58].

2. ***Challenge 2: Data, model and tool interoperability.***
   The results from **Paper A** show that various system modeling and simulation approaches have been used by industry for different domains. Moreover, there are also different languages, simulation and modeling tools used for different stakeholders and target products. This is a challenge for tool-chain developers, as it requires careful application of methods to integrate data, models and tools based on the primary functionalities of tool-chains.
3. ***Challenge 3: Co-simulation across domains.***
   As simulation models are developed for different subsystems, it is challenging to integrate verification at the system level using integrated simulations. Co-simulation is proposed as one solution of integrated simulations to support multi-disciplinary collaborative and concurrent design [59]. Because these heterogeneous models have different semantics and data structures, specific simulation tools require unified interface specifications supporting data exchange when developing an integrated co-simulation tool-chain.
4. ***Challenge 4: Design automation.***
   Within the MBSE tool-chains for concurrent and collaborative designs [23], technical resources, including tools, data and models, are involved in the whole development process. When stakeholders from different domains implement their work tasks, considerable manual and iterative operations lead to a waste of R&D resources and the risk of erroneous decisions. Therefore, design automation, referring to creating and configuring CPS development efficiently using digital technologies [60], is challenging to the design of MBSE tool-chains, as a lot of case-by-case automation has to be considered.
5. ***Challenge 5: Tool-chain formalism and assessment.***
   Currently, industries have proposed different MBSE tool-chains to deal with the increasing complexity of CPS. However, when developing MBSE tool-chains, many factors need to be considered, such as tool capabilities and interoperability. In order to reduce the risks of product development due to erroneous decisions concerning MBSE tool-chains, tool chain assessment is important. An unsuitably designed tool-chain can lead to increasing R&D cost for both the tool-chain development itself and the CPS development using the tool-chain.

1.3 Thesis Objective

In this section the goals and research questions of this thesis are defined based on the concepts defined in Section 1.1.2-1.1.3 and further challenges analyzed in Section 1.2.1-1.2.3.

1.3.1 Goals

The main goal of this thesis is to propose a framework to construct MBSE tool-chains that overcome **Challenges 1-4** mentioned in Section 1.2. Moreover, based on this framework **Challenge 5** is addressed by the proposition of a model-based approach to formalize and assess tool-chains before their development.

The goals of the thesis are thus as follows:

1. ***Goal 1: A framework to support CPS tool-chain development by ensuring that the developed MBSE tool-chains:***
   Goal 1.1: Formalize system artifacts and development processes:

As described in **Challenge 1**, understanding of the system characteristics, system development, and the deployment of technical resources and their relationships are Key Performance Indicators (KPIs) for complexity management of CPS. The models employed by the MBSE tool-chains should formalize these aspects, their dependencies and the associated traceability of such information.

Goal 1.2: Support flexible deployment and unified representations of technical resources:

As described in **Challenge 2**, during CPS development, technical resources, such as models, data and tools, are required by different stakeholders. For example, simulations are required to connect with architecture design tasks and requirement models. The framework should provide a flexible solution for stakeholders and their tools to access and deploy technical resources for their work. Moreover, the framework needs to support unified representations of technical resources as middleware for information exchange in order to improve the interoperability of technical resources.

Goal 1.3: Support co-simulations across domains:

Challenge 3 describes the proliferation of heterogeneous models when implementing co-simulation, which are associated with different domains and stakeholders during CPS development. Therefore, the framework needs to integrate heterogeneous models using standards in order to make co-simulation more flexible.

Goal 1. 4: Support design automation:

Challenge 4 describes how design automation can impact the efficiency of CPS development, as it is challenging to implement considering all the case-by-case automation required. This suggests that the different scenarios in the specific domains should be formalized. Therefore, the framework needs to support design automation with such scenarios in the developed MBSE tool-chains. Three main features are then required for automation:

- ***Support of a model-driven process management of system developments:*** In the MBSE tool-chain, development processes of CPS are often formalized using models. However, such models are required to support real development process implementation in different process management systems, such as PLM [61], in order to eliminate the gaps between the developed models and real development processes.
- ***Support of flexible technical resource deployment:*** In order to deploy technical resources from the process management systems, operations of related tools for different scenarios need to be represented as services or service templates which are manipulated by different tasks in such systems. Moreover, models, data and development information are also required to be manipulated by other tools and stakeholders in a more flexible way.
- ***Support of decision-making in the development process for automated value selections of design parameters based on simulations:*** During CPS development, simulations are widely used for system design and V&V, for instance for functional behaviors of AI algorithms and system performance analysis of CPS. Repeated simulations are always adopted for parameter optimizations which take time and human efforts. Thus, automated decision-making for value selections of design parameters provides potential to promote the development efficiency for management and deployment of simulations.

2. ***Goal 2: MBSE tool-chain formalism and assessment***

   As described in **Challenge 5**, tool-chain developers expect to formalize the tool-chain concepts and assess tool-chain capabilities early before prototyping. This requires tool-chain concepts to be formalized and then assessed by qualitative and quantitative approaches. Such assessment results are able to provide clues as to the effects of tool-chain implementation for the tool-chain developers.

Goal 2.1: To support tool-chain formalisms

MBSE tool-chains can be considered as systems whose architecture descriptions are constructed by abstracted compositions and their interactions. Such elements should be considered as the basis for defining meta-models in order to formalize tool-chain architectures.

Goal 2.2: To support tool-chain assessment

A specification needs to be defined for assessing different tool-chains' capabilities. Both qualitative and quantitative approaches are useful to promote the confidence of tool-chain developers when assessing MBSE tool-chains. Therefore, MBSE tool-chain formalisms should be extended to support both qualitative and quantitative analysis.

1.3.2 Research Questions

A research question is, by nature, one for which no answers exist yet. Based on the problem formulation, research questions and associated hypotheses are proposed below. **RQ1** and **RQ2** are defined based on **Goal1**, while **RQ3** is defined based on **Goal2**.

1. ***RQ1:*** How should the concept of MBSE tool-chains be defined to allow the identified challenges to be considered in full when studying, developing and assessing MBSE tool-chains? This includes considering:
    a. The research scope and boundary of MBSE tool-chains.
    b. The research methodologies for describing MBSE tool-chain architectures.
    c. The general concept of MBSE tool-chains, such as how tool-chain developers construct and formalize their MBSE tool-chains.
    d. The KPIs for assessing MBSE tool-chains.

2. ***RQ2:*** How could a framework be constructed for developing MBSE tool-chains in order that they support formalisms of system artifacts and development, tool interoperability, co-simulation and design automation? This includes considering:
    a. The techniques required to construct an MBSE tool-chain, including how system developers make use of such techniques when designing MBSE tool-chains.
    b. The techniques required to support MBSE formalisms across the entire lifecycle, including how system developers make use of these techniques when using MBSE tool-chains.
    c. The techniques required to support tool-integration during CPS development, including how system developers make use of these techniques when using MBSE tool-chains.
    d. The techniques required to support co-simulation across domains in the MBSE tool-chains, including how system developers make use of techniques when using MBSE tool-chains.
    e. The techniques required to support design automation in MBSE tool-chains, including how system developers make use of these techniques when using MBSE tool-chains.

3. ***RQ3:*** How can MBSE tool-chains be assessed, particularly on MBSE capabilities [44] and Levels of Information Systems Interoperability (LISI) [15]? This includes considering:
    a. The compositions and capabilities in MBSE tool-chains that need to be formalized.

b. The measurements required to assess MBSE tool-chains
c. The techniques used to support MBSE tool-chain assessment, including how tool-chain developers make use of these techniques for early verification of MBSE tool-chain.

1.4 Delimitations and Alternatives

Based on historical perspectives and investigations into MBSE and MBSE tool-chains, there is indeed a clear need for research into MBSE tool-chain development for CPS development. Furthermore, specific techniques, such as domain-specific modeling, model-based process-driven approach, co-simulation, ontology design and tool-integration are currently isolated. They need to be integrated for MBSE tool-chain construction to improve both the efficiency of tool-chain development and the efficiency of CPS development. The work in this thesis focuses both on the development and effects of MBSE tool-chains. It's hoped that the proposed framework in this thesis would make it easier and more systematic for tool-chain developers to perform their work, so that they make correct decisions. However, a number of explicit delimitations are made for this work.

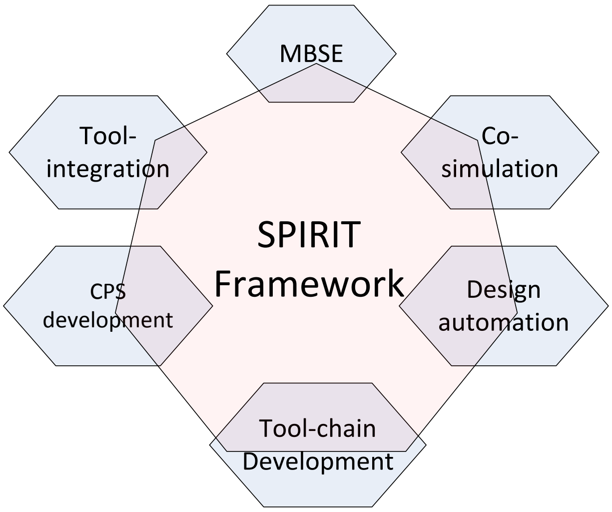

Fig. 1.3 Focuses in this thesis

1. The questionnaires of MBSE application are distributed to Chinese industry. Further investigations of other countries are part of future work.
2. From literature reviews and the questionnaire survey, this thesis focuses on the six aspects shown in Fig. 1.3.
3. The accuracy of co-simulation is not a primary concern. The tool-chains developed in this thesis have used co-simulations that have relied on commercial tools and open-source tools based on HLA and FMI.
4. MBSE in this thesis is focused on systems-level formalisms and verification; Specific domains, such as software and hardware of embedded systems are not considered.
5. Social factors of enterprises transitioning to using MBSE tool-chains are not considered, e.g. culture and organization.

1.5 Outline and Reading Guides

In this section, outline of the whole thesis is first introduced. Then a reading guide illustrates how appended papers support MBSE tool-chain development including concept, design, early evaluation and operation phases.

1.5.1 Outline

The thesis is divided into five chapters based on 8 published and under-review articles. A brief overview of each chapter is provided below. A more detailed map of the thesis contents – and where to find answers to the research questions – can be found in Section 3.6.1.

Chapter 1 presents an introduction to the thesis, discusses the problems faced in CPS development and identifies motivations and challenges of MBSE tool-chain development. Stakeholders are analyzed to support the problem formulation. The chapter ends by establishing research questions and limitations.

Chapter 2 explains the research design, which is largely based on a DRM Framework [62]. The research actions are totally implemented based on a soft system and systems thinking methodology ([63] and [64]). The chapter provides details about the research content and studies conducted. The methods for developing these studies and their detailed measurements are explained. The contents are highlighted in relation to the methods for answering and evaluating the research questions mentioned in Section 1.3.2.

Chapter 3 presents details about the main content of the papers on which the thesis is based. Focus is on a framework to develop CPS tool-chains using a service-oriented and MBSE approach. This chapter also summarizes the theoretical foundation of the included papers by discussing domain-specific modeling, co-simulation, a service-oriented approach for tool-integration, a model-driven process management approach, ontology design based on OWL and decision-making algorithms supporting automated co-simulations and a model-based approach for early evaluation of MBSE tool-chains.

Chapter 4 presents and discusses the research questions. A summary of three MBSE tool-chain prototypes is provided to illustrate the effects of MBSE tool-chains on CPS development. Finally, the threats to validity in each of the papers that the thesis is based on are explained.

Chapter 5 concludes the thesis by summarizing the appended papers and findings gained in the context of the proposed research and future work.

1.5.2 Reading Guide

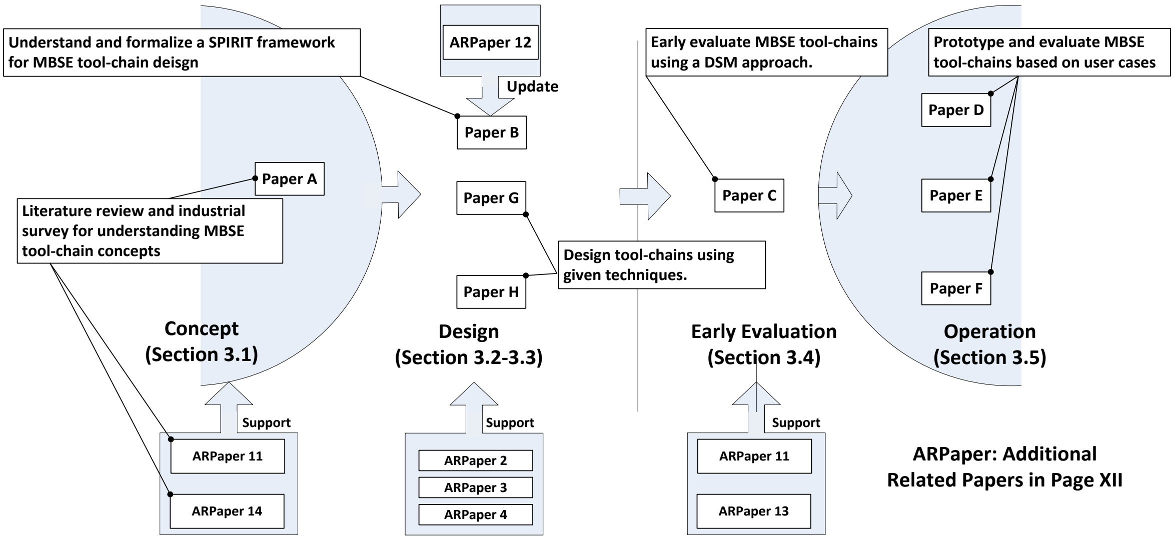

Fig. 1.4 Reading Guide

As shown in Fig. 1.4, the reading guide illustrates main papers as well as additional related papers (e.g. **ARPaper 11**). The appended publications are arranged for the concept, design, early evaluation and operation phases of MBSE tool-chain development. Though each paper is independent, appended papers and additional related papers are contributed to development phases of MBSE tool-chains. Moreover, they proposed and evaluated general techniques for MBSE tool-chain development. From the comprehensive perspectives of this thesis, the following is recommended:

Paper A helps readers to identify challenges and benefits of MBSE tool-chains for defining tool-chain concepts, where **ARPaper 11** and **ARPaper 14** also contribute to construct some concepts of MBSE tool-chains (Section 3.1).

Paper B demonstrates the SPIRIT framework (Section 3.2) which is an evolution from the SPIT framework (**ARPaper 12**) developed based on [43]. Compared with the SPIT framework, the SPIRIT framework supports tool-chain design based on a service-oriented approach. **Paper G** and **Paper H** clarify core techniques used in the SPIRIT framework: 1) Domain-specific modeling based on GOPPRR; 2) Service orchestration based on OSLC. Except for these two papers, **ARPaper 3** and **ARPaper 4** illustrate how the given GOPPRR approach supports architecture-driven, code-generation and ontology generation. In addition, **ARPaper 2** introduces an OSLC approach to support Internet of Things (IoT) system development and data integration. Such techniques are introduced in Section 3.3.

Paper C helps readers to make use of a DSM approach to support early evaluation of MBSE tool-chains (Section 3.4). Moreover, **ARPaper 11** and **ARPaper 13** are two additional related papers to support **Paper C.**

Paper D, **Paper E** and **Paper F** demonstrate three case studies of MBSE tool-chain prototypes which are developed based on the proposed SPIRIT framework (Section 3.5). In the operation phases, these prototypes are implemented and evaluated in the case studies, by comparing with manual operations without supports of tool-chains.

2 Research Design

In this chapter the research design is introduced. The research was conducted in four research projects according to the Design Research Methodology (DRM) Framework [62] and the Soft System and System thinking Method (SSM) [65].

An overview of the research and associated papers is first introduced based on the DRM Framework (Section 2.1). Then SSM is introduced to guide design actions (Section 2.2). The four research projects that form the basis for this thesis are described (Section 2.3). Finally, the methods used in the four projects are described in detail (Section 2.4).

2.1 Overview of Research and Appended Papers

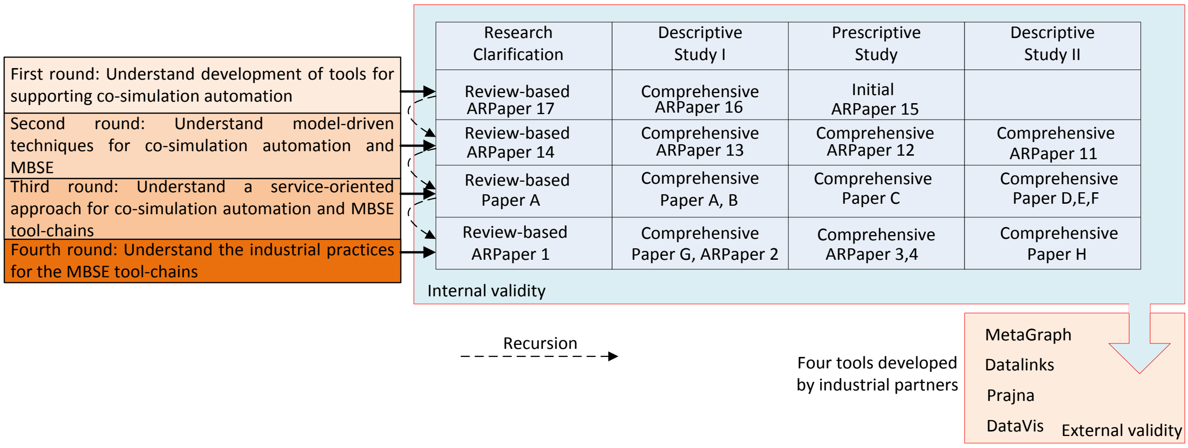

Fig. 2.1 The overview of the appended and additional related papers (ARPaper)

Fig. 2.1 describes the overview of the research and the associated papers appended to this thesis. Four research projects were used to iteratively refine the research, moving from vague concepts to the well-understood final targets. These research projects were structured according to the DRM Framework [62], which represents a main research method for this work. Each research project thus consists of three or four types of studies, each with a different motivation for being carried out [62].

- **Research Clarification**
  This type of study aims to identify clear challenges and goals, initial measurements and a realistic overall research plan using review-based methods. The deliverables are twofold: 1) an initial understanding and expectations of the research, and 2) an overall research plan.
- **Descriptive Study I**
  This type of study aims to increase understanding of the research and its key performance identifications (KPIs) by reviewing literature about empirical research or undertaking empirical research. The deliverables include: 1) a complete design for the remaining studies in the project, including measurement of KPIs, identified metrics and criteria, and 2) implications of findings achieved so far.
- **Prescriptive Study**
  This type of study aims to improve both research and design, even if it means extending the scope defined in the research clarification. The deliverables include: 1) improved KPIs, 2) measurement results for KPIs, and 3) further findings.
- **Descriptive Study II**

This type of study aims to evaluate the final KPIs. The deliverables include: 1) results of a success evaluation, which identifies whether the intended impact has been achieved, and 2) results of an application evaluation, referring to usability and applicability.

The studies can be either review-based, initial or comprehensive. A review-based study reviews relevant literature. An initial study indicates improvements using an empirical research. A comprehensive study includes both a literature review and empirical research.

The internal validity of the final results is supported through the use of the DRM Framework, as it encourages researchers to position, use and evaluate multiple methods [62]. Therefore, the implications for the internal validity of the final results due to the methodological choices of the four research projects are discussed in Section 2.4. As external validity is also influenced by the choice of method, this is also discussed in that section.

2.2 Soft System and Systems Thinking Method

The SSM approach is used to support the design actions in the third research project. In this way the author could use the DRM framework, while structuring the actual design of the MBSE tool-chain prototypes. SSM is proposed by Lancaster University for solving "management and business problems" using a systems engineering approach [65]. The core of SSM aims to identify "problems" using combinations of practical approaches. In this thesis, the SSM approach is integrated with a questionnaire survey, literature reviews, and case studies.

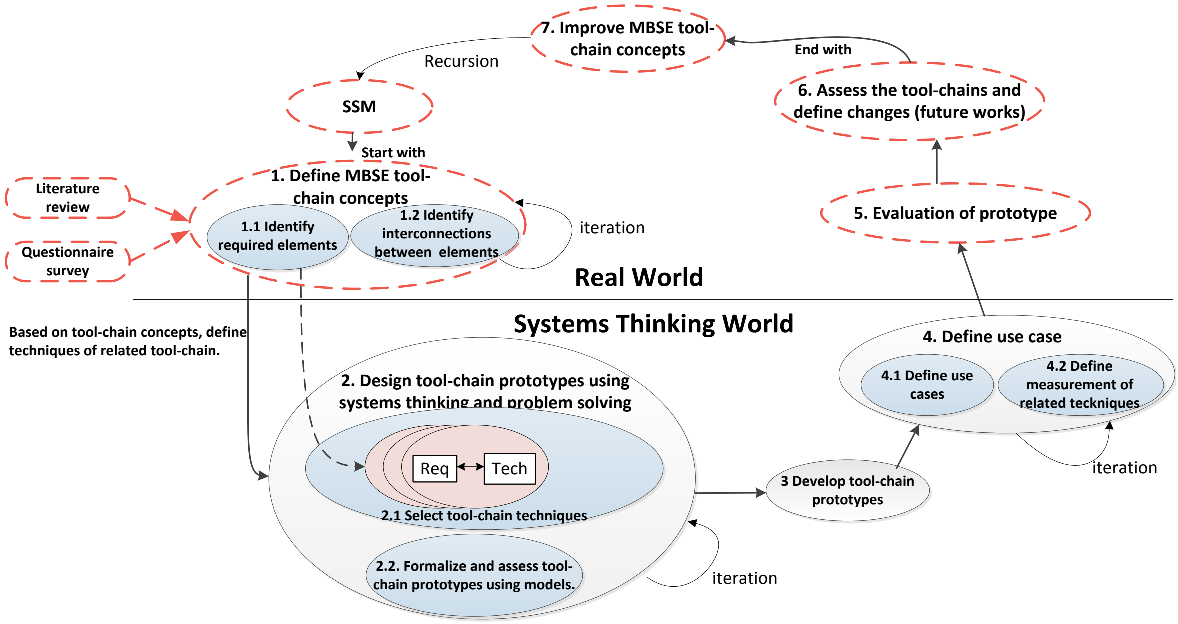

Fig. 2.2 The SSM approach for recursions

As shown in Fig. 2.2, the approach is designed to investigate and develop solutions for MBSE tool-chains in seven steps:

1. ***Define MBSE tool-chain concepts:*** In this step, concepts and their interconnections related to MBSE tool-chains are defined based on literature reviews and a questionnaire survey. The questionnaire survey is used to understand the industry's needs for MBSE tool-chains. The literature reviews are used to under-

stand how MBSE tool-chains are developed by academia. This process is iterative in that concepts are updated during the whole recursion process.

2. ***Design tool-chain prototypes based on the use of systems thinking and problem solving approaches:*** In this step, systems thinking and problem solving are used to design MBSE tool-chain prototypes. Based on concepts defined in ***Step 1***, a systems thinking approach is used to construct one MBSE tool-chain solution and to define its compositions and their interactions. Then using a problem solving approach, techniques are selected for prototyping MBSE tool-chains. Finally, DSM models are used to formalize and assess the tool-chain prototypes. KPIs of these tool-chains are defined and evaluated early using models before prototyping them.
3. ***Develop prototypes:*** Using selected techniques, prototypes are developed.
4. ***Define use case:*** Based on the prototypes, use cases and KPIs are defined for evaluating MBSE tool-chains.
5. ***Evaluation of prototype:*** After implementing the tool-chains they are evaluated in case studies.
6. ***Assess the tool-chains and define changes (future works):*** Through comparing the results with the real world, shortfalls of the current tool-chain concepts are captured and future work is defined.
7. ***Improve the MBSE tool-chain concepts:*** Based on the future work defined in the previous step, a new concept is developed. Finally, an additional SSM is launched in the next round study.

2.3 The Four Research Projects

This section describes the purpose of each research project and how it was addressed. Each research project refined the research goal. In this way the research iterated from vague concepts to the well-understood final targets.

2.3.1 First Round of Studies

In this round, the goal was to understand the development of tools for supporting co-simulation automation. **Additional Related Paper 17** (shortened to **ARPaper 17**) reports on the discourse on model-based frameworks for aero-engine control system design. Several modeling methods were surveyed and identified. Moreover, based on selected modeling frameworks, **ARPaper 16** demonstrates a systems engineering lifecycle and a model management system for model-based design of aero-engine control systems. It provides initial KPIs for co-simulation tools supporting aero-engine control system development. From the findings of **ARPaper 16**, **ARpaper 15** proposed and evaluated a tool for configuring and implementing co-simulations based on the Matlab GUI which indicates a co-simulation tool that can support automated configurations of co-simulation [66]. However, from this round of studies, results show that model-driven techniques are more flexible in supporting co-simulations, particularly to improve the connections between co-simulations and engineering processes.

2.3.2 Second Round of Studies

In the second round, the goal was to understand model-driven techniques for co-simulation automation and MBSE. **ARPaper 14** reports on a literature review of MBSE tool-chains in the aerospace domain. It investigated the concerns of stakeholders and the functionality required by their tool-chains. Moreover, it proposed an initial conceptual framework for formalizing architecture descriptions of tool-chains. **ARPaper 13** describes the formalization of co-simulation tool-chains using DSM models. This paper analyzed the

MBSE tool-chain's architectures using models and visualizations and obtained initial KPIs for co-simulation tool-chains. **ARPaper 12** describes the use of a tool-chain based on model-driven techniques to support the automated co-simulation for entire vehicles. The results show that tool interoperability and design automation are challenging when improving current practice. **ARPaper 11** reports on a comparison of tool-chains based on a service-oriented approach with tool-chains based on pure model-driven techniques to investigate the benefits of the service-oriented approach. Results indicate that the service-oriented approach should be included as a research goal for MBSE tool-chain development. In addition, tool interoperability and design automation were defined as the two main application KPIs of MBSE tool-chains.

2.3.3 Third Round of Studies

This thesis starts from the third round of studies, from which eight associated papers are appended. This round of studies aims to understand how to adopt a service-oriented approach for co-simulation automation and MBSE tool-chain development.

Paper A describes the results from two studies. The first is an in-depth literature review of MBSE and MBSE tool-chains. The second is an investigation into MBSE applicability in China using a questionnaire survey. The questionnaire was designed based on previous literature reviews and empirical research, and distributed to Chinese industry. In this paper, several challenges related to MBSE tool-chains were identified, such as tool-integration. Based on the survey results and previous rounds of studies **Paper B** proposes the SPIRIT framework, i.e. **S**ocial, **P**rocess, **I**nformation-and-se**R**vice-**I**nfrastructure and **T**echnical layers for MBSE tool-chain development. The paper considered the use of a service-oriented approach to support MBSE tool-chain development based on a systems thinking approach. KPIs were summarized in the SPIRIT framework, and served as the basic metrics for **Paper C**. **Paper C** proposes a DSM approach for MBSE tool-chain formalisms and assessment. Using the DSM approach, it compares two MBSE tool-chains for supporting auto-braking system development, which were developed based on a service-oriented approach and a model-driven technique, separately. DSM models were first used to formalize these two tool-chains. Then the models were transformed to visualizations and Bayesian network (BN) models to assess these MBSE tool-chains using qualitative and quantitative approaches.

Paper D, **Paper E** and **Paper F** evaluate the MBSE tool-chains developed based on the SPIRIT framework with three case studies: aero-engine performance analysis, automated co-simulation based on High-level Architecture (HLA) and Functional Mock-up Interface (FMI) and automated parameter value selections of auto-braking systems. **Paper D** proposes an MBSE tool-chain to support co-design of the aero-engine performance analysis. Stakeholders implement a co-design process through a web-based process management system (WPMS) where they manipulate technical resources across domains automatically without any manual operations. **Paper E** proposes an MBSE tool-chain supporting automated co-simulation using HLA and FMI. The MBSE tool-chain generates java codes to manipulate the co-simulations between FMUs referring to black boxes generated from other simulation models through HLA Run-Time Infrastructure (RTI). **Paper F** proposes a decision-making algorithm to support automated value selections of design parameter for AI algorithms of an auto-braking system using an MBSE tool-chain. The parameter value selection refers to an identification process to define design spaces of parameter values using simulation models. The identified parameters are used for configuring the co-simulation models in the later phases automatically.

2.3.4 Fourth Round of Studies

In order to validate the generalizability of the results from previous rounds, a fourth research project was implemented with industrial partners with access to real CPS development environments. This shifted the scope to understanding industrial practices for the MBSE tool-chains.

ARPaper 1 is a technical report describing the current state-of-the-art of MBSE tool-chains based on a literature review that included **Paper A**. This report identifies the challenges of MBSE tool-chain development and investigates techniques of MBSE tool-chains. **Paper G** illustrates a domain-specific modeling language (DSML), *Karma***,** for MBSE formalisms across the entire lifecycle. *Karma* is developed based on Graph-Object-Point-Property-Role-Relationship (GOPPRR) meta-meta models, i.e. it is a textual language for formalizing meta-meta models, meta-models and models. Based on the *Karma* language, a Commercial-Off-The-Shelf (COTS) tool, *MetaGraph*, was developed in which several industrial practices are implemented. This is introduced together with an Open Services for Lifecycle Collaboration (OSLC) approach to support tool-integration in **ARPaper 2**. Using this approach, another COTS tool-*Datalinks* was developed for development, configuration and management of OSLC adapters. From the development of COTS tools and their industrial practices, some KPIs are identified, primarily to define measurements during the third round of studies.

In **ARPaper 3** the research scope is extended. **ARPaper 3** introduces new functionalities of the *Karma* language to support architecture-driven and code-generation implementations. **ARPaper 4** focuses on a unified ontology for describing MBSE formalism based on GOPPRR meta-meta models. These techniques were implemented in *MetaGraph*. Moreover, two other COTS tools were developed: 1) *Prajna* was developed to support co-simulation based on FMI, and 2) *dataVis* was a data visualization tool which supports OSLC specifications. Case studies were used to separately evaluate these techniques.

The papers mentioned in the previous paragraphs identified service orchestration as a large challenge when using our MBSE tool-chains. Thus, the research project focused on a service orchestration design approach using OWL in order to support automated co-simulations. In **Paper H**, a scenario-based ontology design approach is illustrated, aiming to develop the ontology representing the interrelationships between WPMS, OSLC entities representing DSM models and technical resources, and service orchestration templates. The designed ontology is used to support co-simulation automation in the MBSE tool-chains.

2.4 Methods

This section describes the different research methods used within the separate studies in the appended and additional related papers, with particular attention paid to internal and external validity where appropriate.

2.4.1 Questionnaire Survey

The questionnaire survey is a well-known research technique for gathering and analyzing data from a set of defined questions [67]. In this thesis, a questionnaire survey is adopted to understand the MBSE tool-chains from industry. As MBSE is a new concept in China, the internal validity of this aim relies on achieving a broad enough coverage to reach the few engineers who understand this concept. Therefore, the internal validity does not rely on achieving a representative sample.

Table 2.1 Scopes of survey

Interest group	Domain	Potential response for survey	Social network
CCOSE(http://www.ccose.org/), a systems engineering interest group	systems engineering	422	Weichat
Four forums about model-based design	CAD, CAE and Mechanical engineering	Open	Internet
Modelica interest group	Multi-domain modeling	116	Weichat
PDM/PLM interest group	PLM/PDM	558	QQ
Multi-domain modeling group and co-simulation interest group	CAE, Modelica and co-simulation	998	QQ
Interest groups of AMESim, Matlab/Simulink and Flowmaster	CAE and modeling	330	QQ
Forums of systems engineering, systems engineering methodology,	Systems engineering	1060	Weichat
Forums of Forward design	Design method	86	Weichat
Forums of Suzhou Tongyuan	Modelica	109	Weichat

The survey was designed with a set of questions administrated by a questionnaire survey website (https://www.wjx.cn/), including multiple-choice, single-choice and matrix-single-choice questions. The respondent scopes of this survey were the CCOSE group (one systems engineering group in China, 500 people), the ModelicaChina group (110 people) and a co-simulation group (1000 people). The details are shown in Table 2.1. QQ and Weichat are two platforms referring to the most popular live-chat tools in China [68]. The potential number of responses is estimated according to the size of the interest groups listed in these two platforms. The questionnaire includes 50 questions about four aspects:

– Basic information of the respondents
– The extent of MBSE usage
– Related techniques about MBSE tool-chains
– MBSE transitioning

The internal validity also relies on how well the respondents can understand the survey. Therefore, the survey was tested on two people, one expert in the industrial application of MBSE and one expert on MBSE tool development.

The result of the survey is shown in **Paper A** where concepts and challenges for developing MBSE tool-chains are first identified.

2.4.2 Literature Review

A literature review gathers, critiques and synthesizes representative literature on a topic in an integrated framework [69]. In order to understand the state-of-the-art of MBSE tool-chains, literature reviews are used to analyze the techniques used in MBSE tool-chains and to identify KPIs of MBSE tool-chain in detail. The scope of the systematic literature reviews carried out divide them into two types:

- Domain-specific literature reviews: Aerospace and aircraft industries are considered as two important domains for MBSE application. Therefore, they were selected for literature reviews. Papers from 28 journals and 26 conferences of AIAA were surveyed by searching for the keywords “tool-chain” and “MBSE” in *Google scholar*. The results in **ARPaper 14** provide an initial definition of MBSE tool-chains.
- Exploratory literature review: “MBSE”, “Model-based Systems Engineering” and “tool-chain” were searched for in *Google scholar*. About 200 papers were collected, with several topics analyzed as shown in **ARPaper 1**.

The internal validity of a literature review depends on the systematic process applied when conducting it. Kitchenham’s procedure for conducting literature reviews is implemented using seven steps: 1) research questions; 2) search processes; 3) inclusion criteria; 4) exclusion criteria; 5) quality assessment; 6) data collection; and 7) data analysis [70]. Details are shown in **ARPaper 1**.

2.4.3 Problem Solving

Problem solving is the act of defining a problem, understanding the cause of the problem, identifying, prioritizing and selecting alternatives for a solution, and implementing the solution [71]. In order to design tool-chain prototypes, tool-chain compositions are analyzed using a systems thinking approach for examining the interactions between them in a defined system boundary of MBSE tool-chains. A workflow is illustrated for design MBSE tool-chain concepts:

Table 2.2 Technical selection

Requirement	Technical solution	Key technique selection
Techniques for Constructing Tool-chains		
Capability to formalize system artifacts and development	DSM for formalizing system artifacts and development.	1) DSM supported by a DSM tool, *MetaEdit+* [13].
Capability to support flexible deployments and unified representations of technical resources	DSM and a service-oriented approach for linking technical resources and design information with development processes.	1) DSM supported by Meta Edit+. 2) OSLC [72]. 3) FMI [73].
Capability to support integrated verification based on simulations across domains	Co-simulation for integrating multi-domain models.	1) HLA [74] and FMI. 2) Simulink and FMI.
Capability to support a model-driven process management approach	Model-based process management and IT platforms for process control and management.	1) *BPMN Camunda* and its process engine [75] .
Capability to support decision-makings in the development processes for auto-mated parameter value selections	Automated decision-makings for parameter value selections based on simulation optimizations	1) One developed algorithm to support service orchestration between WPMS and service-oriented resources.
Techniques for Tool-chain assessment		
Capability to formalize tool-chains	Systems thinking and tool-chain formalisms	1) DSM approach supported by *MetaEdit+* [13].
Capability to assess tool-chains in a qualitative approach	Visualization	1) Data visualization supported by *Baidu echart* [76].
Capability to assess tool-chains in a quantitative approach	Capability assessment	1) BN models.

- ***Define system boundary:*** Based on the literature reviews and questionnaire surveys, the system boundary of the MBSE tool-chain is defined.
- ***Define the requirements of MBSE tool-chains:*** Within the system boundary, requirements related to MBSE tool-chains are captured from industrial needs and trends, and academic focuses (Section 1.3.1), as shown in Table 2.2.
- ***Define tool-chains' technical solutions:*** Based on the requirements, technical solutions of the prototypes are defined to construct MBSE tool-chains.
- ***Key technique selection:*** In order to realize the technical solutions, key techniques are selected to develop the prototypes.

Moreover, in order to assess the tool-chain prototypes before development, several techniques are selected, including DSM models and visualization as shown in Table 2.2.

2.4.4 Case studies

Case study refers to a specific research design for examining a problem and acquiring knowledge of the phenomenon from intensive exploration of a single case [77]. Internal validity and external validity are two important aspects of case studies.

The internal validity means ensuring that the claims made by a study are correct, e.g. how sure a researcher is that there are no plausible alternative explanations to the results of an experiment [78]. In this thesis, it refers to a process to identify whether industrial problems will be solved by the solutions put forward by this thesis.

Table 2.3 Evaluation design to support Prescriptive Study

Evaluation Types	Evaluation method	Metrics		Criteria
Qualitative approach	Visualization	Tool-chain complexities: interactions among views of development processes, systems, models, tools, stakeholders and their relationships.		**Paper C** and **ARpaper 11**
		Design automation: the degree DSM models automate technical resources.		
		Interactions between model flows: the way technical resources (models and data) link to each other.		
		Interactions between tools: the way tools link to each other.		
Quantitative analysis	BN models	TMCL	Interoperability of technical resources.	**Paper C**
			Social Procedures refers to standards and management policies supporting MBSE.	
			MBSE application refers to MBSE capability of each tool.	
			Infrastructure refers to capabilities to integrate with existing infrastructure systems, such as IT systems.	

Three case studies are used to support the claims in the **Prescriptive Study** and **Descriptive Study II** phases of the third round study: Case 1: co-design of aero-engine performance analysis; Case 2: automated co-simulation based on HLA and FMI; and Case 3: automated parameter value selections for AI algorithms of auto-braking systems. These cases are used to illustrate the effects of MBSE tool-chains developed by the proposed SPIRIT framework. Several steps are implemented:

- Prototyping and implementing the MBSE tool-chains in real use cases.
- Obtaining feedback from industry and academia.
- Eliciting the important experiences of the implemented MBSE tool-chains.

In addition, based on concerns of the tool-chains' users and developers introduced in Section 1.2.2, the KPIs of MBSE tool-chains are designed and measured, including: 1) data, model and tool interoperability; 2) design automation; 3) human efforts on implementing their works. The KPIs are evaluated in two phases.

In the Prescriptive Study, a DSM approach is used to assess KPIs of two different MBSE tool-chains (**Paper C**). The measurements of KPIs are designed to support qualitative and quantitative analysis of MBSE tool-chains. The qualitative analysis is implemented based on visualizing tool-chain architectures. The quantitative analysis is implemented based on BN models. Among these measurements, metrics are defined as shown in Table 2.3, in particular, a metric called tool maturity capability level (TMCL) is defined to support quantitative analysis of the MBSE tool-chains (see details in **Paper C**).

In the Descriptive Study II, these cases are prototyped and evaluated based on the defined measurements separately. As shown in Table 2.4, two main types of KPI evaluation are considered: 1) success evaluation, referring to identification of whether the MBSE tool-chains contribute to the success of CPS development; 2) application evaluation, referring to identification whether the support can be used in the situation for which it is intended and that it addresses the factors it is supposed to address.

In Table 2.4, the success evaluations of case studies are implemented by prototyping to verify if the MBSE tool-chains satisfy the key demands of case studies. The application evaluations are implemented by comparing the effects of CPS development with and without using tool-chains.

The claim that the identified industrial problems are handled by the solutions put forward by this thesis seems well founded.

Table 2.4 Evaluation design to support Descriptive Study II

Case Study	Success evaluation	Application evaluation	Metrics
Case 1	Prototyping	Comparison with manual operations of co-simulation without MBSE tool-chains.	**Paper D**
Case 2	Prototyping	Comparison with manual code programing for co-simulation based on HLA and FMI.	**Paper E**
Case 3	Prototyping	Comparison with tool-chain operations without automated parameter value selections.	**Paper F**

The external validity of a study refers to the extent to which its results can be replicated in other groups, environments, and contexts [78]. In this thesis, it refers to the case studies contributing towards understanding the use of MBSE tool-chains in other domains and countries. In this thesis, a team including software developers and systems engineers from industry is led by me to develop four COTS tools which I contribute all the scientific ideas and lead software development:

- *MetaGraph*, a DSM tool supporting CPS development (http://www.zkhoneycomb.com/).
- *Prajna*, an integrated verification tool for co-simulation based on FMI, distributed simulation based on HLA and BN modeling (http://www.zkhoneycomb.com/).
- *Datalinks*, an OSLC tool supporting OSLC adapter development (http://www.zkhoneycomb.com/).
- *DataVis*, a data visualization tool based on OSLC (http://www.zkhoneycomb.com/).

The appended papers only detail their use in the aerospace domain, but they have also been used in the automotive and IoT domains. Although they have not been rigorously evaluated in that domain, the results seem promising and the involved experts have provided positive feedback. However, the appended papers only detail the use of these tools in a context where MBSE is novel and recently adopted, i.e. in Internet of Things (IoT) companies in China. In contexts where MBSE is more mature, the claims made in Section 4 need further study. More challenges might appear or existing challenges might evolve in appearance as MBSE is more extensively used.

3 The Proposed Framework and Research Contributions

In this chapter, the Social, Process, Information-and-seRvice-Infrastructure, Technique (SPIRIT) Framework is introduced to support MBSE tool-chain development including concept, design, early evaluation and operation phases. The chapter starts by providing tool-chain definitions, aiming to formalize the tool-chain architectures during the concept phases (Section 3.1). During the design phase, the framework and its workflow are described (Section 3.2). The techniques used in the framework include a domain-specific modeling approach based on GOPPRR meta-meta models, a service-oriented approach based on OSLC, co-simulation and automated decision-making algorithms to support co-simulation and design automation in the MBSE tool-chains (Section 3.3). During the early evaluation phase, a DSM approach is used to assess MBSE tool-chains using qualitative and quantitative analysis (Section 3.4). In addition, three MBSE tool-chain prototypes are developed in different case studies during the operation phase (Section 3.5). Finally, contributions are summarized (Section 3.6).

3.1 MBSE Tool-chain Concepts

3.1.1 Model Definitions

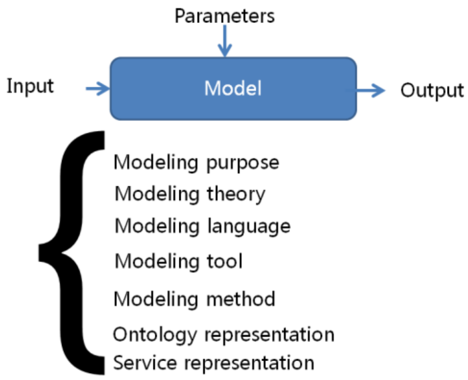

Fig. 3.1 A model in MBSE tool-chains

In order to develop basic concepts, models constructing the MBSE tool-chains are firstly defined in Fig. 3.1. Parameters are used to configure the models which provide outputs based on inputs for realizing modeling purposes. The model is defined associated with several views:

- Model purpose: the purposes of modeling, “why is the model needed?”
- Modeling theory: the mathematical foundation of modeling, e.g. differential-algebraic system of equations.
- Modeling language: any language expressing information or knowledge or systems in a structure that is defined by a consistent set of rules.
- Modeling tool: tools implementing models.
- Modeling method: a set of concepts to explain “how to develop models using a given language in one modeling tool to represent the formalisms?”, e.g. finite element modeling and structural equation modeling.
- Ontology representation: unified formalisms representing models of different languages in the CPS development.

- Service representation: service descriptions of models. The service refers to technical resources and activities that cause a state transformation of a technical resource for models, such as changing a parameter in a Simulink model.

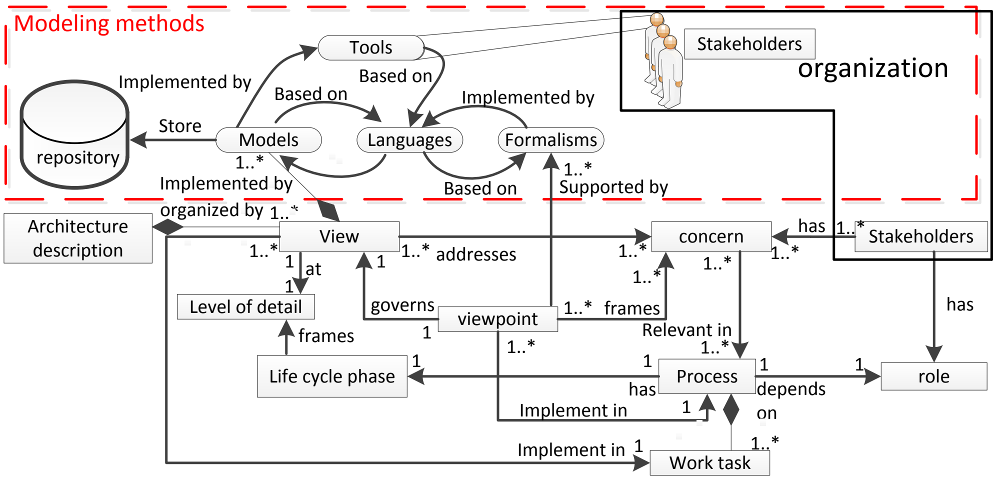

Fig. 3.2 Entity relationships between systems engineering and modeling methods

3.1.2 Modeling Methods and Systems Engineering

As illustrated in Fig. 3.2, we integrate modeling methods [79] with systems engineering including the development process and system architecture [80], aiming to illustrate the relationships between systems engineering and modeling methods. The core is to clarify several interrelationships between work tasks in the process, viewpoints related to the system architecture, models and their connections as shown in **ARPaper 11**.

Fig. 3.2 provides clues to integrate modeling methods, organizations and the development processes, and system architecture of CPS for MBSE tool-chain development. The modeling methods include models, language, formalisms, tools and repository. The models are developed based on languages representing formalisms of system artifacts and development processes. Moreover, stakeholders make use of tools to build models which are stored in a repository.

A system has an architecture description which is organized by views to address stakeholders' concerns. Each stakeholder with one role requires views governed by viewpoints to frame their concerns. No matter which roles they have, each stakeholder has their own development process. The development process, in which viewpoints are implemented, includes work tasks at different lifecycle phases for framing levels of detail which refer to product hierarchies (such as system of systems (SoS), system, subsystem).

A view includes one or several models stored in a repository and is implemented by tools based on languages. They are used to represent system artifacts and development processes. The languages are developed based on formalisms determined by viewpoints. Stakeholders make use of tools to build models based on appropriate languages for formalizing the viewpoints in the development process.

3.1.3 MBSE Tool-chain Concept Development

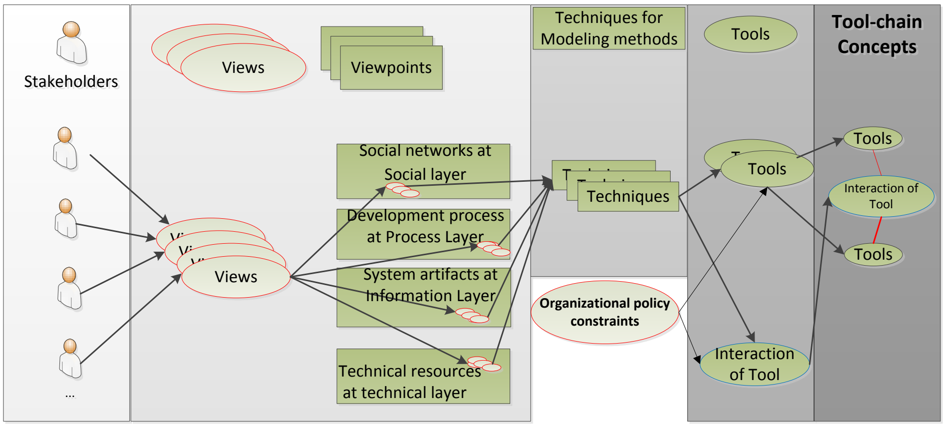

Fig. 3.3 A workflow for developing tool-chain concepts

Based on the interrelationships between the development processes, system architectures and modeling methods as shown in Fig. 3.3, a workflow is defined to develop a tool-chain concept for stakeholders' requirements. The tool-chain concept refers to a tool-chain reference architecture description.

First, the tool-chain stakeholders are selected (developers, users, maintainers and acquirers) to provide development views. Then, such views are classified into the four layers (viewpoints): 1) Social layer; 2) Process layer; 3) Information layer; 4) Technical layer (details in [30]). These views are captured as requirements for evaluating techniques for modeling methods including modeling and integration techniques during MBSE tool-chain development. Based on organizational policy constraints and selected techniques, tools and tool interactions are defined. For example, if there is a policy that MySQL is allowed as the only data base, software developed based on another data base cannot be selected. Finally, a tool-chain concept is defined based on the candidate tools and their planned interactions.

3.1.4 MBSE Tool-chain Concepts

When tool chains are developed, a three-dimensional concept model is proposed to describe the MBSE tool-chains based on integration of Hall's model [81] and DIKW model [82], as shown in Fig. 3.4 (see details in **Paper C**, **ARPaper14** [40] and **ARPaper11** [42]). Using the model, the SPIRIT framework is developed by considering different viewpoints related to MBSE tool-chain development. Moreover, a domain-specific modeling language is developed to formalize the MBSE tool-chains based on the elements in the conceptual model (introduced in Section 3.4).

In the conceptual model, the DIKW model is used, with roots in knowledge management, to explain the ways of moving from data (D) to information (I), knowledge (K) and wisdom (W) which is represented in the *Dimension of data levels*. The DIKW model is used to represent collections of technical resources at different data levels and their interrelationships. Therefore, development processes, system artifacts (collections of views in different viewpoints, such as requirement, architecture) and data levels of technical resources are used to define tool-chain concepts. Details are introduced as follows:

- *Dimension of the product lifecycle processes* refers to the development process including various stages, phases and work tasks.
- *Dimension of system artifacts* refers to viewpoints (and views) of system developers, e.g. requirement, function and architecture.

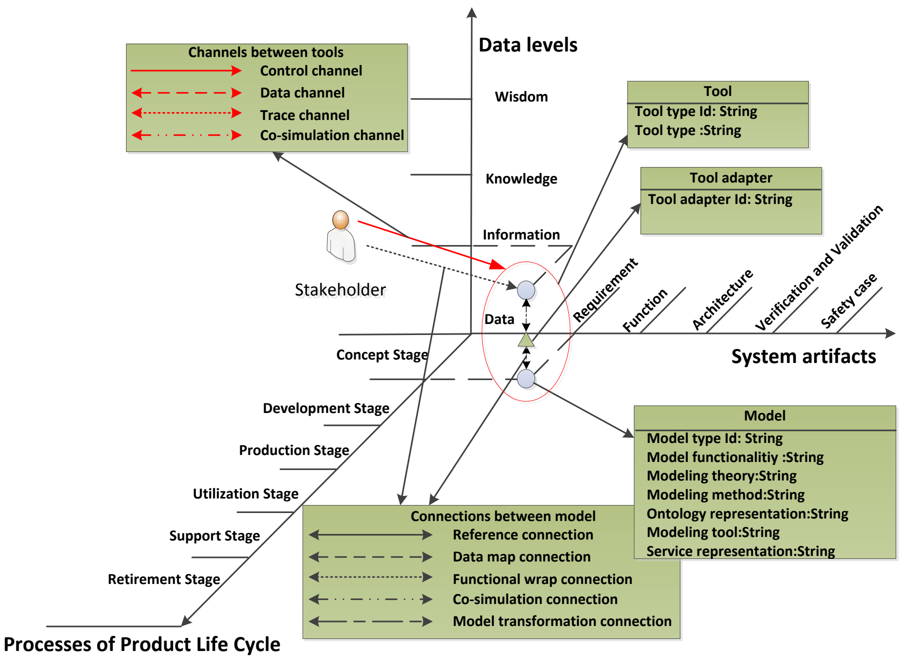

Fig. 3.4 A three-dimensional conceptual model to formalize MBSE tool-chains

– *Dimension of data levels* refers to characteristics and purposes of technical resources including data, information, knowledge and wisdom.
  - Elements in Data layer refer to technical resources implementing simulations, design, analysis, verification & validation and optimization in the entire lifecycle.
  - Elements in Information layer refer to technical resources describing 'information' which formalize system artifacts and development processes, such as SysML models for describing system requirements of CPS development.
  - Elements in Knowledge layer refer to technical resources that have been represented as services which are verified and able to share with others, such as OSLC services of data which are accessible to other stakeholders.
  - Elements in Wisdom layer refer to technical resources supporting decision-making during development, such as machine learning models for decision-making of one decision gate.

– In addition to these dimensions, stakeholders, models, tools, tool adapters and their interactions are defined as follow:
  - Stakeholder elements refer to the users implementing the MBSE tool-chains.
  - Model elements refer to any forms of data, models and documents used in the MBSE tool-chains.
  - Tool adapter elements refer to a tool set configuring interfaces for tool integration.
  - Connection elements between models include:1) Reference connection; 2) Data map connection; 3) Functional wrap connection; 4) co-simulation connection; 5) model transformation connection (details in [42]);

The above mentioned elements are used to define model flows for the MBSE tool-chains. Moreover, based on the model flows, tool-chain concepts are constructed using the following elements:

  - Tool elements implementing models and tool adapters refer to tools used in the MBSE tool-chains.

- Channel elements between tools include: 1) control channel; 2) data channel; 3) Trace channel; 4) co-simulation channel (details in [42] and [83]).

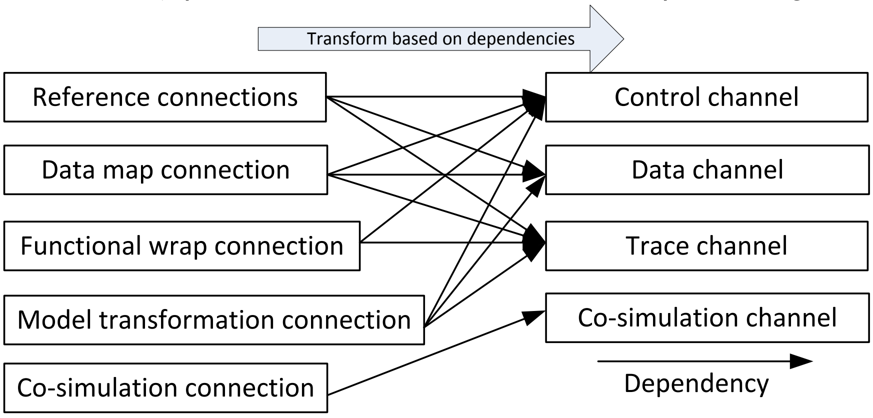

Fig. 3.5 Dependencies between model connections and tool-chains

During tool-chain development, model flows are firstly defined based on modeling techniques for stakeholders' views as shown in Fig. 3.3. Then tool-chain concepts are developed based on model flows. There are some dependencies between model connections and tool channels as shown in Fig. 3.5. Based on such dependencies, tool-chain channels are automatically created based on model flows.

3.2 A Framework for CPS Tool-chain Design

3.2.1 The SPIRIT Framework

In order to support CPS tool-chain design and implementation, the SPIRIT framework is developed based on the three-dimensional conceptual model as shown in Fig. 3.4. The SPIRIT framework is extended from SPIT (Social, Process, Information and Technical layers) (introduced in Section 1.1.3) and makes use of an IRI (Information-seRvice-Infrastructure) layer to replace the Information layer, shown as Fig. 3.6. The updated service-oriented framework aims to improve tool interoperability and design automation using services. We summarize the concepts in each layer as follows:

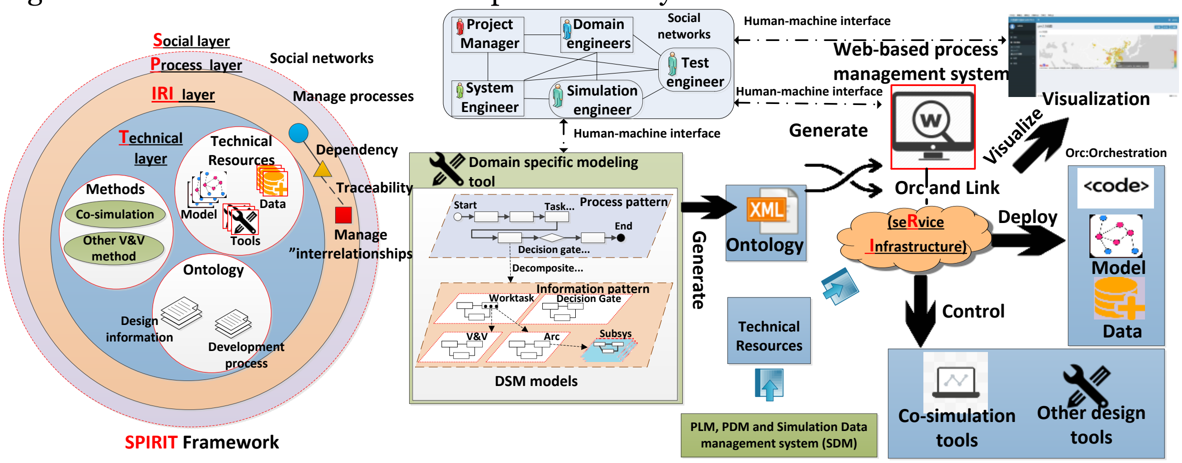

Fig. 3.6 The SPIRIT Framework

- The **Social layer** involves an explicit network of the stakeholders related to CPS development (and the interrelationships between them) and implicit organizational constraints or policies for MBSE tool-chains. The stakeholders implement development processes using MBSE tool-chains and address concerns in architecture descriptions of their target products. They also implement their own tasks and manipulate the corresponding technical resources (models, data and tools) to support the CPS development. The implicit factors refer to the rules and policies of such social networks, concerns of tool-chain developers and organizational constraints of the related tool-chains.

- In the **Process layer**, development processes related to the target products are considered. Each development process consists of work tasks (including *Human Work Task* and *Automated Work Task*) and decision gates representing decision-making processes. In the SPIRIT framework, the development process is formalized by process patterns in DSM models as indicated in Fig. 3.6.
- In the **Information and seRvice Infrastructure (IRI) layer**, viewpoints related to system architectures are represented as information, such as requirements, model structures and functions which are defined as system artifacts (e.g. what's the requirement?) and deployment of technical resources in each work task. These system artifacts and technical resources are wrapped as services in a service infrastructure which can be accessed through URLs. System artifacts are formalized by information patterns in a domain-specific modeling tool (shown in Fig. 3.6) which are transformed to an ontology for generating services. These services are used to access design information and to manipulate technical resources for CPS developers (introduced in Section 3.3.3).
- The **Technical layer** includes ontology, modeling methods and technical resources. Ontology refers to unified and standardized representations of development processes in the Process layer and information of system artifacts and technical resource deployment in the IRI layer. Modeling methods are used for formalizing the development process and related system views, e.g. simulation in Matlab/Simulink. The technical resources include data, model, codes and tool APIs supporting the development process and modeling methods.

3.2.2 The Workflow for MBSE tool-chain Implementation

Based on the SPIRIT framework, a workflow is used to support MBSE tool-chain implementations to integrate domain-specific views, tool-integration and process management based on a service infrastructure. The details are illustrated in Fig. 3.7 (numbered to represent the main part of the workflow):

In the social networks (Part 1), stakeholders build DSM models and implement the development process in the WPMS. Therefore, a social network (red box) includes: 1) Stakeholders participating in the product development (Part 1.1); 2) Stakeholders building models (Part 2) to formalize the development processes (Part 1.4) and system artifacts (Part 1.5) in DSM tools (Part 1.2); 3) Stakeholders using the WPMS embedded with OSLC services of design information and technical resources to implement work tasks (Part 1.3).

Development processes (Part 1.4) are formalized by the process patterns (Part 2.1) of DSM models including human work task (HWT), automated work task (AWT) and decision-gates. System artifacts (Part 1.5) in each work task of development processes are formalized by the information patterns (Part 2.2) in DSM models, such as requirement, model structures and V&V. Through a developed code generator (Part 2.3), DSM models are transformed to an ontology in XML (Part 5) including process and information patterns (Part 5.1 and Part 5.2).

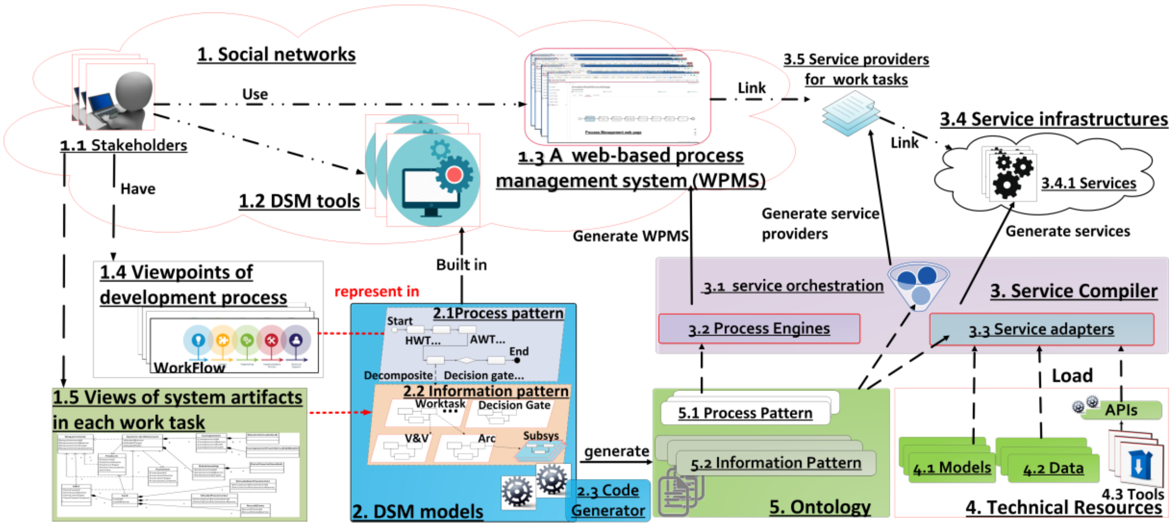

Fig. 3.7 Workflow to implement MBSE tool-chains

Technical resources (Part 4) include: 1) Models (Part 4.1), e.g. Simulink models; 2) Data (Part 4.2); 3) Tools (Part 4.3) used for work tasks, in particular, providing APIs. A service compiler (Part 3) is used to generate WPMS embedded with services of technical resources and information representing system artifacts whereby: 1) A process engine (Part 3.2) loads process pattern of ontology to generate WPMS; 2) Service adapters (Part 3.3) are used to generate the services of ontology and deployment of the related technical resources; 3) A service orchestration (Part 3.1) loads ontology to manage, configure and orchestrate services to the work tasks in the WPMS (introduced in Section 3.3.4) using service orchestration templates. Thus such work tasks link the required services of information and technical resources from a service infrastructure (Part 3.4) which refers to service collections for accessing design information and deploying technical resources. Finally, stakeholders implement their development processes through WPMS.

3.3 Techniques for Tool-chain Design

Several techniques used in the SPIRIT framework are introduced in the following sections.

3.3.1 Domain-specific Modeling based on GOPPRR Approach

As illustrated in Fig. 3.7, DSM is used to formalize the development process and system artifacts of CPS development. From the literature review, the GOPPRR approach is considered as one of the most powerful meta-meta models [84]. Thus, a GOPPRR approach based on a M0-M3 modeling framework is used to support MBSE formalisms shown in Fig. 3.8 (details in **Paper G**).

– M3 (Meta-meta models): In this paper, meta-meta models are defined based on the GOPPRR meta-meta modeling language [7], which include:
1) The key meta-meta models including **G**raph, **O**bject, **P**roperty, **P**oint, **R**elationship and **R**ole;
  - **Graph** refers to a collection of **Object** and **Relationship**, represented as one window. The **Graph** is a visual diagram on the top level or lower levels decomposed or explored by one **Object**.
  - **Object** refers to one entity in **Graphs** (for example, one class concept in UML).

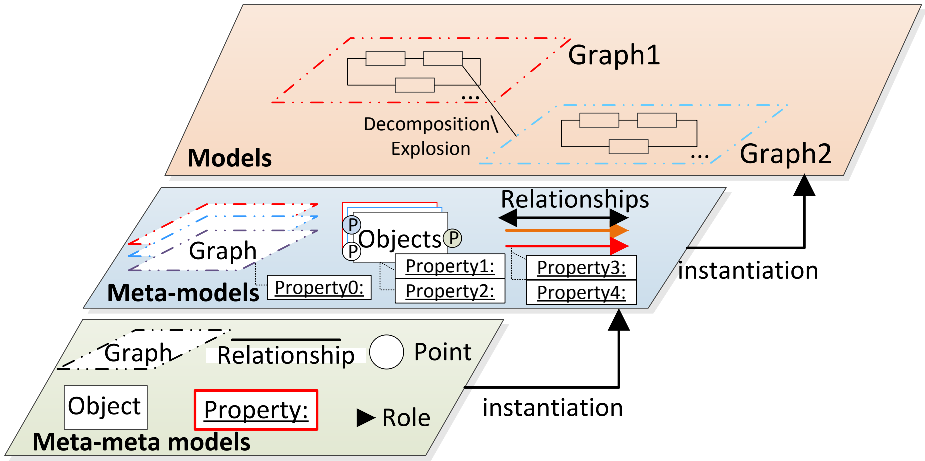

Fig. 3.8 M0-M3 Domain-specific modeling based on a GOPPRR approach

- **Property** refers to one attribute of the meta-models based on the other five meta-meta models (**Graph**, **Object**, **Point**, **Relationship** and **Role**).
- **Point** refers to one port in **Objects**.
- **Relationship** refers to one connection between the different **Points** of **Objects** or **Objects**.
- **Role** is used to define a connection mirrored to the relevant **Relationship**. For example, one **Relationship** has two **Roles**. Each **Role** is defined to connect with one **Point** in Objects. Based on these connections, **Relationships** is allowed for connecting with these **Points** in the **Objects**.

2) The complementary meta-meta models.

- **Binding** is defined in the **Graph** referring to a connection rule between **Object**, **Point**, **Role** and **Relationship**. A **Relationship** has one or more **Bindings**. In each **Binding**, one **Relationship** is connected with one or more **Roles**. Each **Role** is defined to connect with one or more **Points** in related **Objects**.
- **PropertyLinks** refers to a **Property** subordinate of the other five types of meta-meta models.
- **Property Link Type** refers to each **Property** having only one data type.
- **Explosion** represents one or more mappings from one **Object**, **Role**, and **Relationship** to one or more **Graphs**.
- **Decomposition** refers to the fact that an **Object** can be decomposed into a new **Graph**.

– M2 (Meta-models): Meta-models are developed in M2, referring to instantiations of meta-meta-models which are used to build DSM models for formalizing development process and system artifacts of CPS.

– M1 (DSM models): DSM models refer to models describing development process and system artifacts.

– M0 (System views): System views refer to systems engineering views which DSM models describe in the real world.

Using the GOPPRR approach, developers build DSM models using meta-models to describe the development process and system artifacts. Moreover, DSM models are transformed to the ontology[1] using a code generator in **ARPaper 4**.

[1] I am the leader to design a new ontology based on OWL with Chinese partners as shown in [103] .

3.3.2 Co-simulation Supporting Integrated Verification

In this thesis, co-simulation is adopted as an approach to support integrated V&V. Two types of co-simulations are introduced as follows:

3.3.2.1 Co-simulation Environment in Simulink

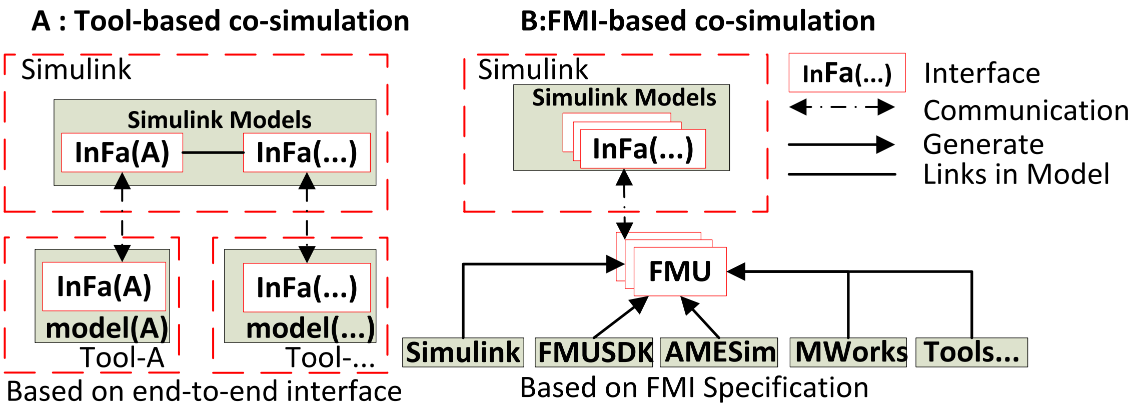

Fig. 3.9 A: a co-simulation environment for tool-based co-simulation in Simulink; B: a co-simulation environment based on FMI in Simulink.

As shown in Fig. 3.9, a co-simulation environment is developed to support tool-based and FMI-based co-simulations in Simulink (details are shown in **Paper D**). Tool-based co-simulation refers to one co-simulation integrated using end-to-end interfaces provided by other simulation tools as shown in Fig. 3.9-A. Based on the interfaces, other simulation tools transform their models to specific formats connected with Simulink to execute co-simulations. All simulation tools execute their own models and communicate with each other during co-simulation. FMI-based co-simulations are executed by Matlab/Simulink based on FMI specifications as shown in Fig. 3.9-B. FMI refers to a tool independent standard to support model exchange and co-simulation between dynamic models in different simulation tools using a combination of XML and executable files (either compiled in DLL/shared libraries or in source code), including: model-exchange and co-simulation [12]. Using FMI, simulation models are transformed into black boxes for co-simulations between tools, called FMU, including an XML file and a dll file:

- The XML file represents model information (e.g. information about simulation platforms).
- The dll file formalizes the original models as a black box used by other tools.

FMI provides a set of methods for other tools to manipulate FMUs. During co-simulation, a master tool sets up, coordinates, communicates and handles the interactions among FMUs.

In FMI-based co-simulation, simulation tools first transform their own models to FMUs. Then, they are integrated in Simulink using a Matlab FMI toolbox which executes all the FMUs for co-simulation [85].

In the co-simulation environment, several tools are adopted to simulate their own domain models for verifying CPS. FMUSDK[2] provides an open source environment to transform C programs to FMUs. AMESim[3] is a modeling environment, particularly for fuel systems supporting FMU generation. Matlab/Simulink is adopted for control systems design

[2] FMUSDK is a free software development kit developed by QTronic, https://resources.qtronic.de/

[3] LMS Imagine.Lab Amesim is a COTS simulation software for the modeling and analysis of multi-domain systems, https://www.plm.automation.siemens.com/en/products/lms/imagine-lab/amesim/.

and to generate FMUs through the Matlab FMI toolbox [85]. MWorks is a Modelica tool to support multi-domain modeling [86].

3.3.2.2 Co-simulation Environment based on HLA and FMI

Except for Simulink, a master engine is developed based on jCerti[23], Certi RTI [87] and javaFMI [88] to execute co-simulations based on High level architecture (HLA) and FMI.

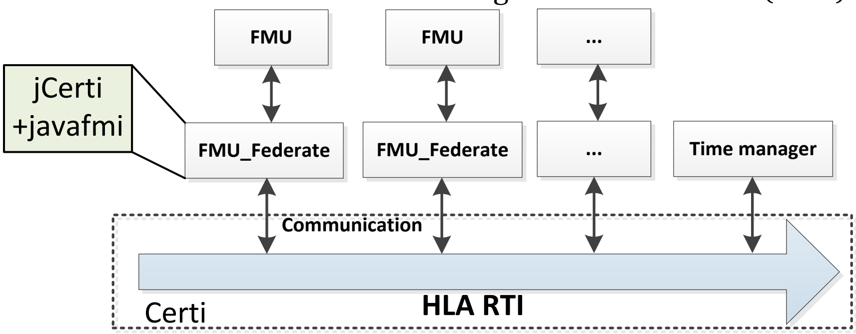

Fig. 3.10 Co-simulation based on HLA and FMI

HLA is an IEEE standard to provide an architecture to facilitate distributed simulation [89]. Based on HLA, each instance of distributed simulation is called a *Federation* constituting several HLA entities, called *Federate*. The HLA entities communicate with each other by a Run-Time Infrastructure (RTI) which refers to a middleware for manipulating communication and execution to manage *Federation* and *Federate*.

As shown in Fig. 3.10, Certi/RTI is used as a master to handle different FMUs. The jCerti (a Java library for developing *Federates*), and javafmi (a Java library for developing FMU interfaces) are used to develop *FMU_ Federate* in order to support co-simulations between different FMUs. *FMU_ Federate* refers to one interface for communicating between FMUs and RTI (details are shown in **Paper E**).

3.3.3 A Service-oriented Approach Supporting Tool-integration

Open Services for Lifecycle Collaboration (OSLC) is an open community to develop practical specifications for data integration building upon linked data [72]. It exposes data, models and tool APIs as web-based services through which other tools and stakeholders can access such data and tool APIs.

As shown in Fig. 3.11, an OSLC service management tool, *Datalinks*[4], is developed to configure and develop OSLC adapters for ontology, FMUs, data, models and tool APIs in the MBSE tool-chain (details are shown in **ARPaper 2**). The *Datalinks* manages the developed OSLC adapters to generate OSLC catalogs, service providers and services representing technical resources based on OSLC core model specifications for tool-integration as shown in Fig. 3.11. The OSLC catalogs, service providers and services refer to web-based services expressing data, models and related APIs in unified formats accessed by defined URLs. The details are shown as follows (see Fig. 3.11):

[4] A OSLC-based tool integration management tool, http://www.zkhoneycomb.com/

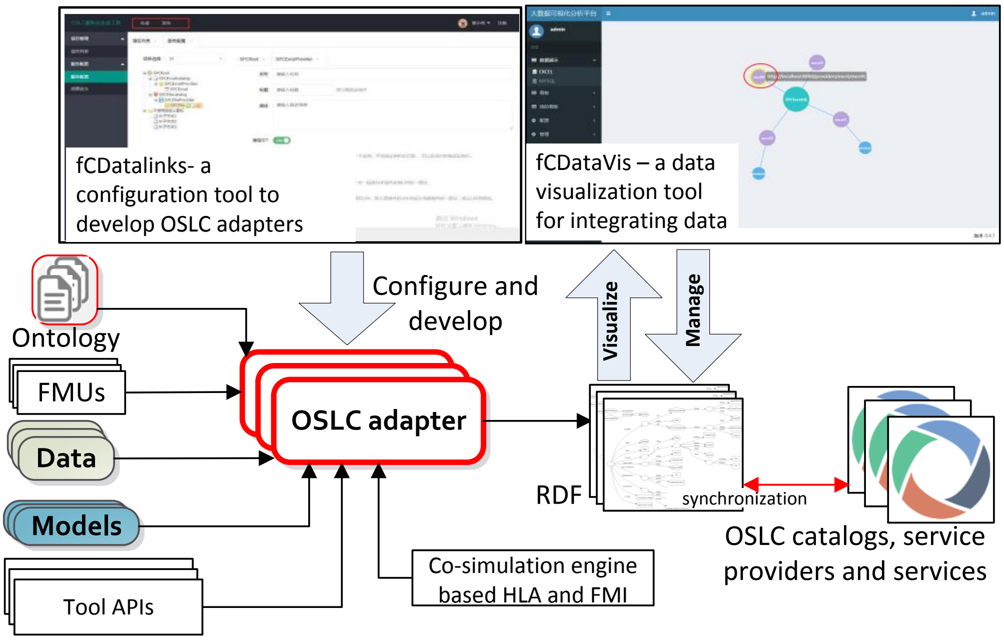

Fig. 3.11 OSLC adapters to generate OSLC service providers and services

- APIs of tools, e.g. Matlab/Simulink, are wrapped into OSLC entities to implement tool operations through their URLs.
- Models, data and FMUs are transformed to OSLC services for being accessed.
- Ontologies in XML files are transformed to related OSLC service providers and services describing development processes and information mirrored with DSM model concepts.

These OSLC adapters also represent such technical resources as unified RDF based on OSLC specifications. When the OSLC adapters transform technical resources to RDF, *DataVis*[5], referring to a data visualization and management tool, is used to support and to manage data visualizations where graphs under different visualization approaches are used to visualize the data.

3.3.4 Design Automation in the MBSE Tool-chains

3.3.4.1 The Model-driven Process Management Approach

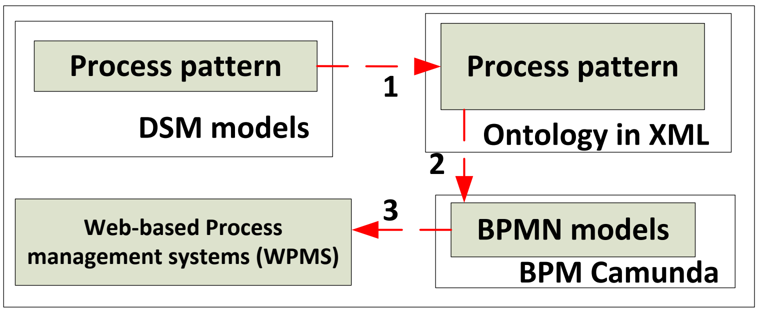

Fig. 3.12 A model-driven process management approach to generate a WPMS

[5] A data management and visualization tool, http://www.zkhoneycomb.com/

In order to generate a web-based process management system (WPMS), process patterns in the DSM models are first transformed to process pattern ontologies in XML. Then the ontology is transformed to WPMS for personal task management, process control and monitoring. After building DSM models, the workflow for transforming the process pattern of DSM models to WPMS is shown in Fig. 3.12:

– The process pattern of DSM models is transformed to process pattern ontologies by a developed code generator in DSM modeling tools.
– Then the process pattern ontologies are transformed to BPMN models by a model transformer developed based on activiti [90] (an Eclipse plugin open source engine transforming ontology described in XML files to BPMN models describing business process diagrams). Model transformers are developed to transform ontology generated from DSM models to a WPMS based on activiti and BPM Camunda [75].
– Finally, WPMS is generated from the BPMN models by a model transformer based on BPM Camunda (an open source process engine to transform BPMN models to WPMS).

3.3.4.2 Design Ontology to Support Service Orchestration in MBSE tool-chains

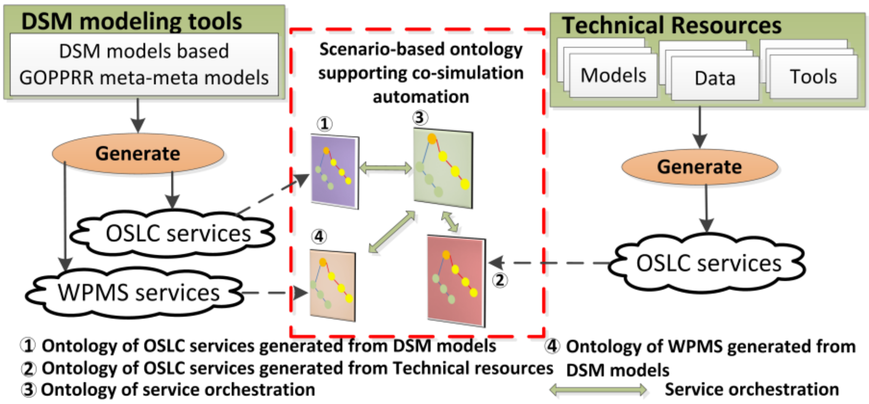

Fig. 3.13 A scenario-based ontology representing service orchestrations

In the MBSE tool-chains developed based on the SPIRIT framework, WPMS needs to link to the required OSLC entities - including OSLC catalogs, service providers and services, generated from ontology and technical resources - using a service orchestration. In order to support service orchestration between WPMS and OSLC entities, a scenario-based ontology is defined to formalize their topologies and logics for implementing CPS development and deploying technical resources.

Fig. 3.13 depicts a scenario-based ontology for integrated representations and logic for implementing CPS development and deploying technical resources in the MBSE tool-chains, in four parts: 1) Ontology representing the OSLC entities generated from the DSM models through their ontologies, to formalize the development process, system artifacts, and deployment of technical resources; 2) Ontology representing the OSLC entities for accessing and manipulating technical resources. 3) Ontology representing service orchestration templates to link the WPMS to OSLC entities representing technical resources and ontology generated from DSM models. 4) Ontology representing the WPMS generated by DSM models for implementing the development processes. The details are described in **Paper H**.

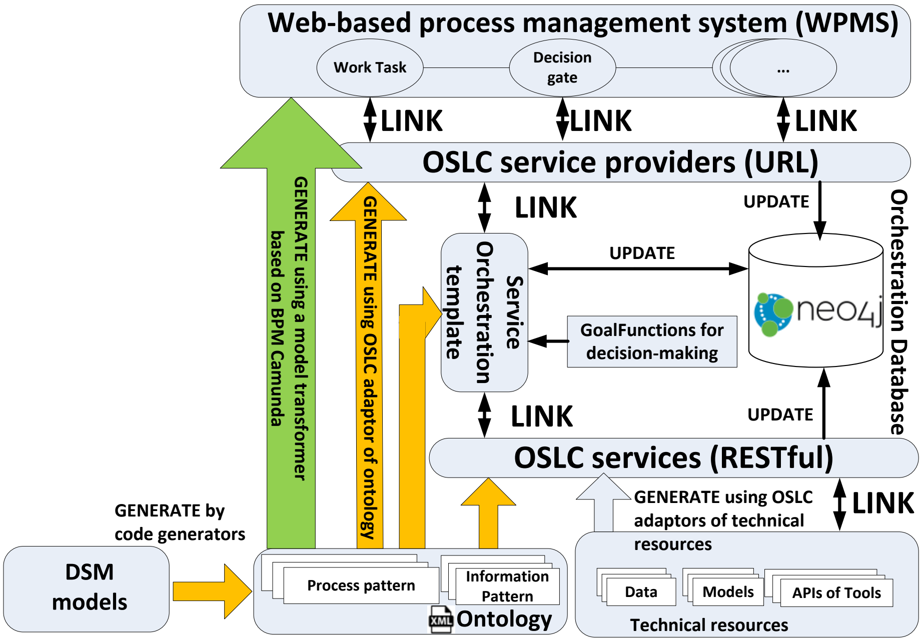

Fig. 3.14 Service orchestration templates

The ontology is defined based on different scenarios for CPS development. Based on the ontology, a template-based service orchestration approach is proposed for the WPMS to implement the development process. In MBSE tool-chains, DSM models are transformed to WPMS embedded with OSLC entities through ontologies as shown in Fig. 3.14. First, the DSM models are transformed to ontology described in XML files. Then a service compiler is developed to support the model transformation as follows:

- OSLC entities, including OSLC catalogs, service providers, services representing technical resources (e.g. models, data and APIs of tools), are generated through OSLC adapters developed using *Datalinks*. Such OSLC entities referring to RESTful services support access and manipulation of technical resources by stakeholders and tools via URLs.
- The process patterns of ontologies in XML files are transformed to WPMS by a compiler based on BPM Camunda, where each work task is linked to the related OSLC service provider through its URL (details introduced in Fig. 3.15).
- The process patterns of ontologies are generated to OSLC service providers to link to the related work tasks in WPMS. The information pattern in each work task of the process pattern is transformed to OSLC services in the related service provider. Such transformations are implemented based on a developed ontology OSLC adapter.
- Based on ontologies, service orchestration templates are implemented to deliver end-to-end OSLC entities from WPMS to technical resources. The OSLC entities generated from the ontology are updated to *Neo4j*, a graph database [91]. Through such service orchestration templates, each OSLC service provider is linked to the OSLC services of required information and technical resources. Thus developers implement required technical resources from WPMS without any manual operations.
- Moreover, a decision-making algorithm (introduced in Section 3.3.4.3) is implemented supported by the service orchestration templates in order to make decisions automatically in the WPMS.

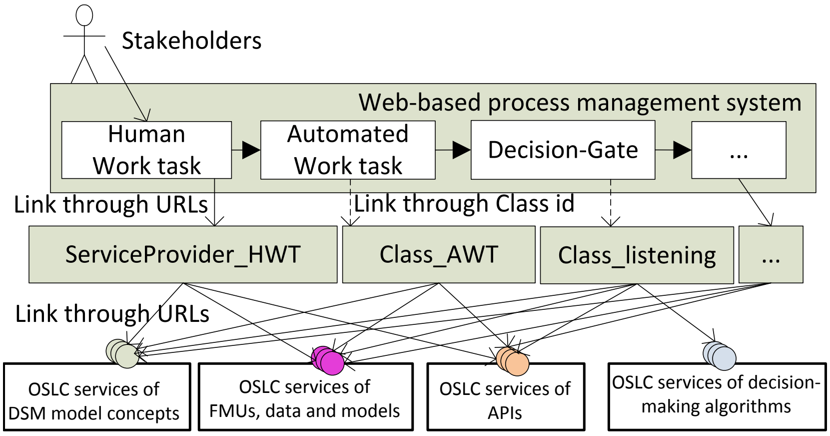

Fig. 3.15 Service orchestrations through OSLC entities

Fig. 3.15 illustrates how service orchestrations are implemented through OSLC entities. In *WPMS*, stakeholders log in their *Human Work tasks* to access the corresponding information through the linked service provider (*ServiceProvider_HWT*) and to manipulate the required technical resources through their OSLC entities. In the *Automated Work task*, WPMS implement automated tool operations using its java class (*Class_AWT*) through the Class id. In *Decision-Gates*, decision-making algorithms are implemented in its java class (*Class_listening*) through their OSLC services. Details are introduced as follows:

In the WPMS, each *Human Work task* links with their service providers, *ServiceProvider_HWT* using URLs. The *ServiceProvider_HWT* links with OSLC entities of required information patterns which represent design information and configurations of technical resources to support required deployment of technical resources.

The *Automated Work task* links with *Class_AWT* referring to a java class to implement OSLC entities using *class id*. *Class_ AWT* implements service orchestration templates constructed by OSLC entities to manipulate technical resources automatically.

Decision-gate links to its related *Class_listening* class to implement decision-making algorithms using OSLC services.

3.3.4.3 Automated Decision-making based on Service-oriented Technical Resources

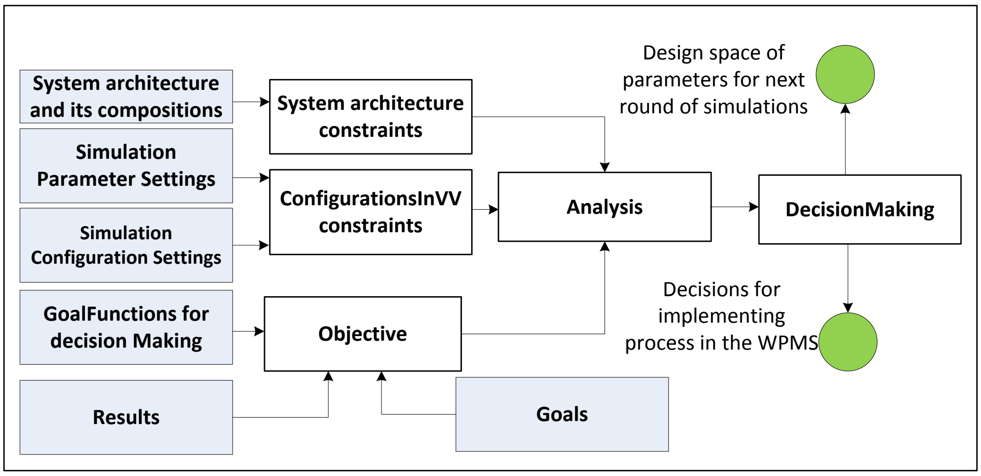

Fig. 3.16 An analysis process to support decision-makings for parameter value selections

In this thesis, a decision-making algorithm is developed to support parameter value selections (details are introduced in **Paper F**). Fig. 3.16 illustrates an analysis process based on the algorithm to support decision-making with three inputs:

– *System architecture constraints* represent the constraints related to a system architecture describing the model structures and its compositions for the analysis process.
– *ConfigurationsInVV constraints* refer to configurations in the V&V for the analysis process for simulations and parameters.
– *Objective* refers to a decision-making objective. The *Objective* defines *GoalFunctions,* referring to selected algorithms for decision-making, *goals* and simulation results in the V&V. In order to realize the purpose, *GoalFunctions* are implemented to check if the *goal* is satisfied by results of V&V.

Based on these inputs, the analysis process creates two outputs: 1) decisions for implementing process in the WPMS; 2) design space of parameter for the next round of V&V. The design spaces of parameters refer to collections of value candidates of the given parameters. The *GoalFunction* here follows the rules in order to realize the purpose in **Paper F**: if all the simulation results are not satisfied by the Objective, the process terminates and returns to the previous task connected with decision gate. If some of the simulation results satisfy the Objective, the next tasks connected with decision gate are implemented until the next decision gate. *System architecture* and *ConfigurationsInVV* constraints which satisfy the *Objective* are recorded in the design space. During the next round actions, *V&V* is implemented under the configurations in the design space.

3.4 Early Evaluation of Tool-chains Using a DSM Approach

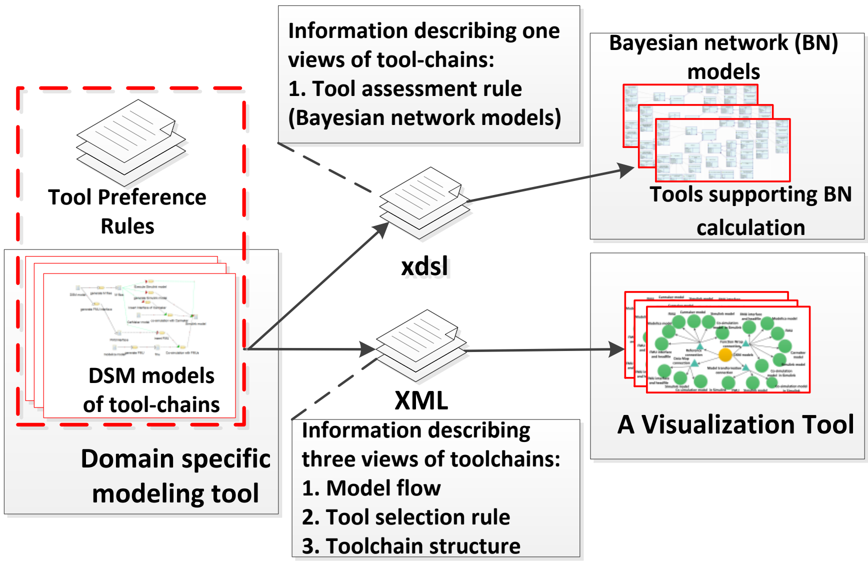

Fig. 3.17 A DSM approach for early evaluation

In order to assess MBSE tool-chains, a DSM approach, whose workflow is shown in Fig. 3.17, is used to support their qualitative and quantitative analysis. *MetaEdit+* is adopted to develop meta-models for tool-chain formalisms including the compositions, model flows and workflows based on Software Process Engineering Meta-model (SPEM) (details in **Paper C**) [92]. Moreover, additional meta-models are designed to formalize tool capabilities and metrics based on Level of Information Systems Interoperability (LISI) [15] and MBSE capabilities [44].

When developers build the DSM models for comparing tool-chains, tool preference rules are defined to formalize dependency relationships between metrics for transforming DSM models into BN models. For example, Table 3 (**Paper C**) represents relationships between criteria in metrics and other metrics for making decisions of such criteria. Using a developed code generator in *MetaEdit+*, XML files are generated to represent tool-chain information for visualizations as well as XDSL files referring to BN models. Finally, BN models are calculated to obtain assessment results of tool capabilities for a quantitative analysis. In addition, a visualization tool loads the XML files to visualize the related views of tool-chains for a qualitative analysis.

3.5 Evaluating Tool-chains in Operation Phases

After early evaluation of MBSE tool-chains, three cases are used to develop tool-chain prototypes based on the SPIRIT framework for evaluating their success KPIs and application KPIs. In
Table 3.1, success KPIs and application KPIs are introduced for each case study. The success KPIs refer to the main purpose of the prototypes in each case study. It refers to whether the developed tool-chain can provide the following capabilities or not: 1) co-design of aero-engine performance analysis; 2) automated co-simulation based on HLA and FMI; 3) automated value selections for parameters of auto-braking systems.

Moreover, several metrics are selected as application KPIs which are evaluated through implementing the prototypes in each case study. Application KPIs are metrics that are compared with the corresponding manual operations in each case. For example, Levels of Communication between Stakeholders refers to the degree that an MBSE tool-chain supports communication among stakeholders when comparing the effects of an MBSE tool-chain with manual operations of doing the same work tasks.

Table 3.1 Success KPIs and application KPIs for case studies

<table>
<tr><th>Case study</th><th>Success KPIs</th><th>Application KPIs</th><th>Paper</th></tr>
<tr><td>Case1</td><td>Co-design of aero-engine performance analysis</td><td rowspan="3">1. Levels of Communication between Stakeholders.
2. Capability of Process Management.
3. Interoperability of Integrated System Simulation.
4. Traceability among development Processes, related Information and Technical resources (models, data and tools).
5. Tool Interoperability.
6. Efficiency of CPS development.</td><td>Paper D</td></tr>
<tr><td>Case2</td><td>Automated co-simulation based HLA and FMI</td><td>Paper E</td></tr>
<tr><td>Case3</td><td>Automated parameter value selections for auto-braking systems</td><td>Paper F</td></tr>
</table>

In Table 3.2, several technical solutions used in the SPIRIT framework are shown first, including DSM approaches, co-simulation, service-oriented approaches, model-based process management approaches and automated parameter value selections. Moreover, the selected techniques and their purposes in each case study are described. For each case study in Table 3.2, different techniques for each solution are used to support their purposes as shown in **Paper D**, **Paper E** and **Paper F**. The evaluation results of these papers are shown in Table 4.1 in detail.

Table 3.2 Techniques of case studies

Technical solutions		Case1	Case2	Case3
DSM approach	Techniques	*MetaEdit+*	*MetaEdit+*	*MetaEdit+*
	Purpose	1. Formalize co-design processes of aero-engine performance analysis 2. Formalize system artifacts of aero-engine performance analysis 3. Formalize deployments of co-simulation between FMUs and Simulink	1. Formalize development processes of co-simulations based on HLA and FMI 2. Formalize system artifacts of the complex systems for co-simulations 3. Formalize deployments of co-simulation between FMUs and Certi RTI	1. Formalize development processes of auto-braking system development 2. Formalize system artifacts of the auto-braking systems for co-simulations 3. Formalize automated decision-making algorithms for co-simulations
Co-simulation	Techniques	Simulink and FMUs	HLA and FMUs	Simulink, Carmaker and FMUs
	Purpose	Support co-simulations of aero-engine components for performance analysis	Support automated co-simulations based HLA andFMI	Support co-simulations of auto-braking system
A service-oriented approach	Techniques	*OSLC*	*OSLC*	*OSLC*
	Purpose	1. Support tool-integration 2. Support service orchestration	1. Support tool-integration 2. Support service orchestration	1. Support tool-integration 2. Support service orchestration
A model-based process management approach	Techniques	*BPMN Camunda*	*BPMN Camunda*	*BPMN Camunda*
	Purpose	1. Generate WPMS 2. Support process control and monitoring 3. Support co-design	1. Generate WPMS 2. Support process control and monitoring	1. Generate WPMS 2. Support process control and monitoring 3. Support automated executions of work task
Automated parameter value selections	Techniques	Manual	Manual	Automated decision-making algorithms
	Purpose	-	-	Automated parameter value selections for co-simulations of auto-braking systems

3.6 Contribution Summary

3.6.1 Overview

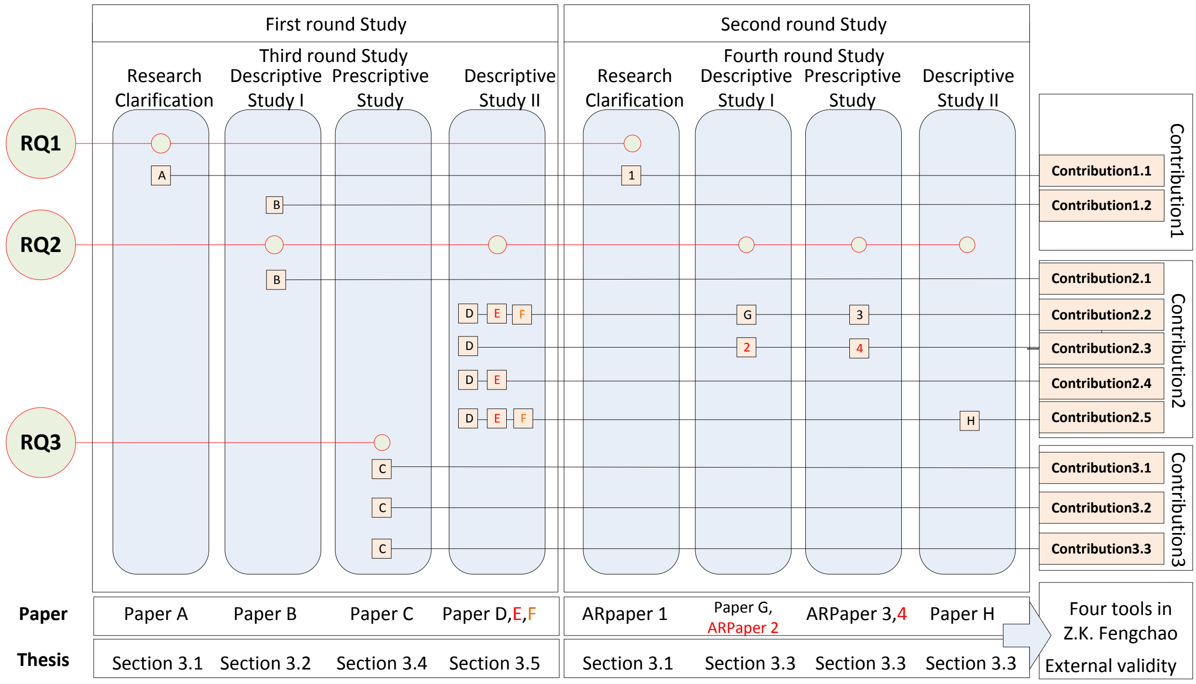

Fig. 3.18 Overview of research questions and contributions

Fig. 3.18 illustrates the relationships between the research questions, appended papers, additional related papers, thesis sections, contributions and four-round study. Fig. 3.18 will be interpreted from reviews of papers and additional related papers as follows:

- **Paper A** considers a questionnaire survey about MBSE and MBSE tool-chains in Chinese industry. The questionnaire includes 4 parts: 1) Basic information of the respondents; 2) The extent of MBSE usage; 3) Related techniques about MBSE tool-chains; and 4) MBSE transitioning. From the results, challenges and benefits of MBSE tool-chains are identified. Supplementary to **Paper A**, **ARPaper 1** refers to one technical report including the questionnaire results in **Paper A** and literature reviews of 200 papers searched through *Google scholar* using the key words "MBSE" and "tool-chain". The comparisons between results of the questionnaire and literature reviews are shown to demonstrate gaps between industry and academia in **ARPaper 1**.
- **Paper B** proposes the SPIRIT framework including **S**ocial, **P**rocess, **I**nformation-se**R**vice-**I**nfrastructure and **T**echnical layers to support MBSE tool-chain development. Using this framework, stakeholders formalize their MBSE tool-chain concepts and design prototypes using systems thinking, particularly from social, process, information and technical views. In this framework, a service-oriented approach is used to support tool integration for the entire tool-chains.
- **Paper C** proposes a DSM approach with BN models and visualizations to formalize and assess MBSE tool-chains before prototyping. Using defined metrics based on MBSE capabilities and LISI, BN models are used to assess MBSE tool-chains for a quantitative analysis. Visualizations are used to analyze the effects of MBSE tool-chains using a qual-

itative approach. Using these two approaches, stakeholders make an early evaluation of the tool-chain concepts before prototyping.

- **Paper D** proposes an MBSE tool-chain prototype based on the SPIRIT framework, aiming to support aero-engine performance analysis. It focuses on the GOPPRR approach for formalizing co-design processes and system artifacts of aero-engine. Moreover, co-simulations based on FMI and Simulink are adopted to support aero-engine performance analysis. Based on OSLC, technical resources are represented as OSLC entities (catalog, service provider and service) which are accessed by other tools through URLs. Then one WPMS is generated based on BPM Camunda， aiming to monitor and control co-design processes. Stakeholders implement such WPMS to manipulate the technical resources for their own jobs through OSLC services automatically.
- **Paper E** proposes an MBSE tool-chain prototype for automated co-simulations based on HLA and FMI. Compared with **Paper D**, it focuses on a DSM approach formalizing development processes, configuration and execution of co-simulation based on HLA and FMI. Certi RTI, jCerti and jFMI are used to support co-simulation executions between FMUs and Run-time Infrastructure (RTI). Using the MBSE tool-chain, java codes based on jCerti RTI and jFMI are generated and implemented through OSLC services. Finally, stakeholders implement their co-simulations through the WPMS automatically.
- **Paper F** proposes an MBSE tool-chain prototype for automated parameter value selections of auto-braking systems. First, a DSM approach is used to formalize not only development processes and system artifacts of auto-braking systems, but also a decision-making algorithm for automated parameter value selections. Through the MBSE tool-chains, decision-making algorithms are implemented to support parameter value selections for configuring co-simulations in the later work tasks by analyzing the simulation results obtained from Simulink models used in the previous work tasks.
- **Paper G** presents the *Karma* language designed based on a GOPPRR approach, aiming to support MBSE formalisms. Moreover, a COTS tool *MetaGraph* is introduced to explain how the *Karma* language formalizes meta-models and models for different general modeling languages and domain-specific modeling languages.
- **Paper H** presents an ontology design approach for formalizing service orchestration in the MBSE tool-chains which is implemented using OWL in protégé [93]. It aims to support deployment of technical resource for implementing co-simulations automatically in an MSBE tool-chain.

3.6.2 Contributions

As shown in Fig. 3.18, the main contributions of this thesis are shown to address the challenges laid out in Section 1.2.3, and to realize the thesis objectives (see details in Section 1.3):

1. ***Contribution 1: Identify the challenges and benefits of MBSE tool-chains***
   - ***Contribution 1.1: Identify backgrounds and definitions of MBSE tool-chains***

   The challenges and benefits of MBSE tool-chains are identified from the questionnaire survey and literature reviews. Questionnaire survey (**Paper A**) focuses on Chinese industry and is compared with a literature review (**ARPaper 1**) based on 200 articles obtained from *Google Scholar* using key words "MBSE" and "tool-chain". Among the identified challenges, there are five considered as background in this paper: 1) formalisms of system artifacts and development processes supporting complexity management; 2) Data, model and tool interoperability; 3) Co-simulation across domains; 4) Design automation; 5) Assess MBSE tool-chains before prototyping. Moreover, from the results, an initial definition of MBSE tool-chains is proposed (Section 1.1.3) where the expected benefits are summarized.

- ***Contribution 1.2: Define a conceptual model of MBSE tool-chains***

One concept model is proposed based on the literature reviews, questionnaire survey and lessons learned from the whole study. In Section 3.1, several definitions are proposed to support formalisms of MBSE tool-chains. Particularly, in Section 3.1.4, a conceptual model is introduced to define MBSE tool-chain concepts from the perspectives of development process, system artifacts and data levels of technical resources (**Paper B**).

2. ***Contribution 2: Develop a framework to support CPS tool-chain development***

- ***Contribution 2.1: A framework supporting CPS tool-chain development using a service-oriented and MBSE approach***

In Section 3.2, a framework is proposed for CPS tool-chain development using a service-oriented and MBSE approach (details are introduced in **Paper B**). It includes **S**ocial, **P**rocess, **I**nformation-se**R**vice-**I**nfrastructure and **T**echnical layers to capture views for MBSE tool-chain development using systems thinking. In the framework, a service-oriented approach is adopted in order to support tool integration and design automation. Moreover, based on several selected techniques, a workflow is developed to implement MSBE tool-chains for system developers.

- ***Contribution2.2: Domain-specific modeling supporting system and development formalisms for complexity management:***

The development processes and system artifacts are formalized as DSM models based on GOPPRR meta-meta models where dependencies and traceabilities of the information are described. The proposed approach is shown to formalize the system characteristics and development of auto-braking systems (**Paper G** and **ARPaper9**). Moreover, in **Paper D-F**, three other case studies are used to illustrate that the GOPPRR approach formalizes not only the system artifacts and development, but also deployments of technical resources.

- ***Contribution 2.3: Standard-based data, model and tool integration to promote data, model and tool interoperability:***

The use of Open Services for Lifecycle Collaboration (OSLC) specifications [94] facilitates development information and technical resources (models, data and tools) being expressed with unified formats which are accessed through URLs. The use of Functional Mock-up Interface (FMI) [12] enables multi-domain models of CPS to be integrated into one co-simulation environment. In the internal validity, **ARPaper2** demonstrates an OSLC approach that supports data integration for IoT development. In **Paper D-F**, FMI is used to support co-simulations across domains in order to support integrated verifications of the target products. Moreover, OSLC is adopted for representing technical resources in order to promote their interoperability in the MBSE tool-chains.

- ***Contribution 2.4: A service-oriented approach for deploying technical resources in order to support co-simulation across domains:***

A service-oriented approach based on OSLC is proposed to support tool integration when implementing co-simulations. Technical resources (e.g. models, data and tools) are accessed and deployed through service-oriented webs [95]. Compared to model-driven approaches, the service-oriented (OSLC) approach enables assessment and manipulation of technical resources by other more flexible technical resources (see details in **Paper C**). Moreover, different types of co-simulations are implemented through the developed OSLC services. **Papers C-E** demonstrate how

OSLC services enable manipulation of different co-simulations (based on HLA/FMI and FMI/Simulink) automatically.

- ***Contribution 2.5: Service orchestration and model-driven process management supporting design automation in the MBSE tool-chains***

Model-based process management supporting system development: The development processes described by DSM models are transformed to a web-based process management system (WPMS). **Papers D-F** demonstrate how DSM models formalizing CPS development are transformed to WPMS which support process control and monitoring of the co-design processes. Using the WPMS, stakeholders deploy technical resources and implement their work efficiently without any manual operations.

Design ontology for service-orchestration using system-thinking: A scenario-based ontology is designed to support service orchestration in the MBSE tool-chains. The ontology is proposed to represent OSLC entities generated from technical resources, formalisms of system development and system artifacts, service orchestration templates and web-based process management systems. It is the basis to realize co-simulation automation through service orchestrations during system development. **Paper H** demonstrates a scenario-based ontology design approach to support service orchestrations for facilitating automated co-simulations in one MBSE tool-chain.

Algorithms support automated decision-makings in the WPMS: Simulations are used to support design and verification of system functional behaviors in case studies. In order to promote efficiency caused by repeated simulations, a decision-making algorithm is used to manage and to deploy simulations for automated parameter value selections during the system development. **Paper F** demonstrates an algorithm for automated design-makings in the WPMS which support parameter value selections during auto-braking system development. The algorithm selects parameters for next round co-simulation models based on the results of Simulink models in the previous phases. Then the selected parameters are used to configure co-simulation models in the later work tasks of the WPMS to promote efficiency of the development process.

3. ***Contribution 3: Provide a model-based approach to formalize MBSE tool-chains and assess capabilities of MBSE tool-chains***

- ***Contribution 3.1: Propose a DSM approach for formalizing MBSE tool-chains using systems thinking***

A conceptual model is defined (**ARPaper 11**, **ARPaper 14** and **Paper B**) for MBSE tool-chains as introduced in Contribution 1.2. Using this concept model, tool-chain developers identify their concerns for MBSE tool-chains using systems thinking. Several views are considered as a reference to design tool-chain prototypes based on the conceptual model.

A DSM approach is developed to formalize MBSE tool-chains based on the conceptual model mentioned in Section 3.4. In the DSM approach, meta-models are developed to represent compositions, interrelationships and capabilities of MBSE tool-chains. Then such meta-models are used to construct DSM models for formalizing different tool-chain prototypes. In **Paper C**, the DSM approach represents two types of MBSE tool-chains using the developed meta-models. Using such meta-models, the tool-chains' architectures are represented in a formal way which supports communication between tool-chain developers and users and enhances their understanding of MBSE tool-chains using a visualized way.

- ***Contribution 3.2: Define measurements for assessing MBSE tool-chains***

Metrics are defined to assess MBSE tool-chains in order to measure their MBSE capabilities and Levels of Information Systems Interoperability. These metrics are defined to implement quantitative analysis of MBSE tool-chains. Moreover, several metrics are defined to analyze the complexity of MBSE tool-chains from a qualitative approach: 1) tool-chain complexity; 2) Design automation; 3) Interactions between model flows; 4) Interactions between tools. Finally, in each case study (**Paper D-F**), specific metrics are defined in order to assess their own MBSE tool-chain prototypes.

- ***Contribution 3.3: A DSM approach supporting early assessment of MBSE tool-chains***

After the DSM models formalize MBSE tool-chains, they are transformed to BN models and visualizations for analyzing the tool-chains' capabilities using a qualitative and quantitative approach, separately. In **Paper C**, visualizations promote developers' understandings of tool-chains and provide clues for quantitative assessment of MBSE tool-chains. Moreover, results of BN models provide quantitative clues for selecting tool-chain prototypes. Using these two approaches, tool-chain developers obtain more confidence about their work before prototyping the tool-chains. This decreases the risks that the developed tool-chains cannot satisfy the demands of their users.

4 Discussion

In this chapter, we discuss our research from three alternative perspectives. We first discuss the responses to the research questions in (Section 1.3.2). Then MBSE tool-chain prototypes developed based on the SPIRIT framework are presented in (Section 4.2). The threats to validity of each paper are explained (Section 4.3.1-4.3.9). Finally, external validity is discussed (Section 4.3.10).

4.1 Elaborating on the Research Questions

RQ1: How should the concept of MBSE tool-chains be defined to allow the identified challenges to be considered in full when studying, developing and assessing MBSE tool-chains?

In this thesis, two types of survey have been conducted: 1) questionnaire survey (**Paper A**); and 2) literature review from scientific articles (**ARPaper 1**). The first survey is used to understand MBSE tool-chains from the needs of Chinese industry. 50 questions were designed related to MBSE and MBSE tool-chains, including: 1) respondents' information; 2) the extent of MBSE usage; 3) techniques used for MBSE tool-chains; and 4) MBSE transitioning. The survey results identify challenges and benefits of MBSE from stakeholders of Chinese industry. The second survey is used to analyze the state-of-the-art of MBSE tool-chains from academia (**ARPaper 1**). Based on the results, several topics are investigated to understand the trends and technical solutions of MBSE tool-chains. The results of these two surveys are the basis to define the concepts related to MBSE tool-chains and to provide clues for the MBSE tool-chain solutions.

- ***What are the research scope and boundary of MBSE tool-chains?***
  A conceptual model of MBSE tool-chains is defined in Section 3.1. Details are seen in **Paper C**, **ARPaper14** [40] and **ARPaper11** [42]. Through the conceptual model (Section 3.1.4), scope and boundary of MBSE tool-chains are identified using systems thinking in **Paper C**.
- ***What are the research methodologies for describing MBSE tool-chain architectures?***
  In order to describe the architectures of MBSE tool-chains, a research methodology is proposed based on a systems thinking approach (Section 3.1.2-3.1.3) in which views are considered in four layers: social, process, information and technical layers (**ARPaper14** [40]). Moreover, based on this methodology, the SPIRIT framework is proposed inspired by the SPIT framework developed by H. Sillitto [43]. This framework enhances the **I**nformation-and-se**R**vice-**I**nfrastructure layer to replace the information layer. The new framework supports tool-integration in order that its four layers are considered together with a service-oriented approach (**Paper B** [26]).
- ***What is a general concept of MBSE tool-chains? How do tool-chain developers construct and formalize their MBSE tool-chains?***
  Concepts and definitions related to MBSE tool-chains are investigated through a literature review (**ARPaper 1**) and questionnaires (**Paper A**). Based on the results, Section 3.1.4 contributes to provide one conceptual model of MBSE tool-chains (see details in **Paper B**, **ARPaper 11** and **ARPaper 14**). Moreover, Section 3.1.4 also provided several concepts for developing meta-models of compositions and interrelationships in MBSE tool-chains (see details in **ARPaper 13** [83] and **Paper C**). Stakeholders make use of these meta-models to build DSM models to describe tool-chains architecture.
- ***What are the metrics for assessing MBSE tool-chains?***

The metrics are summarized in Section 2.4.4 (see details in **Paper C**). In particular, metrics of tool maturity capability level (TMCL) are defined and summarized for quantitative analysis of tool-chains in the Prescriptive Study in Section 2.4.4 (see details in **Paper C**). A DSM approach is developed based on the TMCL metrics for assessing tool-chains using BN models. Moreover, several measurements are proposed for qualitative analysis of MBSE tool-chains using visualizations (Section 2.4.4). In addition, metrics are defined as success and application KPIs in each case study, aiming to assess the MBSE tool-chains using the quantitative and qualitative approaches which are introduced in Section 3.5 (see details in **Paper D, E** and **F**).

RQ2: How could a framework be constructed for developing MBSE tool-chains in order that they support formalisms of system artifacts and development, tool interoperability, co-simulation and design automation?

The SPIRIT framework is proposed to support MBSE tool-chain development in Section 3.2 (see details in **Paper B**). Using this framework, tool-chain developers make use of systems thinking to capture social, process, information and technical views together with tool-integration. Using such views, MBSE tool-chain prototypes are constructed by use of several proposed techniques which are introduced in Section 3.3.

- ***What techniques are required to construct an MBSE tool-chain? How do system developers make use of such techniques when designing MBSE tool-chains?***

  In Section 3.2, the SPIRIT framework is proposed to support MBSE tool-chain development. In this framework, several key techniques are selected using systems thinking and problem solving as shown in Section 2.4.3. The detailed introductions are described in Section 3.3: 1) A domain-specific modeling approach based on GOPPRR meta-meta models; 2) Co-simulation; 3) A service-oriented approach based on OSLC; 4) A model-driven process management approach; 4) Ontology design for service orchestration; and 5) Automated decision-making based on service-oriented technical resources. In Section 4.2, three case studies are defined to develop, implement and evaluate MBSE tool-chain prototypes. These case studies and corresponding prototypes are adopted to support aero-engine co-design, automated co-simulation based on HLA and FMI, and automated parameter value selections for AI algorithms in an auto-braking case. From these case studies, we summarize one workflow for MBSE tool-chain implementation as shown in Section 3.2.2.

- **What techniques are required to support MBSE formalisms? How do system developers make use of these techniques when using the MBSE tool-chains?**

  In this thesis, a GOPPRR approach is proposed to support MBSE formalisms which refer to an integrated description of system artifacts and development. The GOPPRR approach includes meta-meta models, meta-models and models which are introduced in Section 3.3.1 (**Paper G**). Using the GOPPRR approach, meta-models are developed to formalize the development process based on BPMN as shown in **Paper D-F**. Using such meta-models, process patterns are built to describe sequential tasks that humans participate in and computers do automatically. Each work task in the process pattern is broken down into one work task or decision-gate **Graph** formalizing the information patterns referring to system artifacts related to CPS and decision-making algorithms based on developed meta-models. These meta-models are extended from SysML and other languages in order to describe complete information of CPS and CPS development.

- ***What techniques are required to support tool-integration? How do system developers make use of these techniques when using the MBSE tool-chains?***

  During CPS development, different technical resources (models, data and tools) are needed to support CPS development. However, such resources have different syntax, data structures and APIs making it difficult to access them from other tools and stakeholders. As shown in Section 3.3.3 (**ARPaper 2**), a service-oriented approach is proposed to represent data, models and tool APIs as unified OSLC entities with the corresponding URLs. Other stakeholders and tools access the related technical resources and implement related tool APIs through their URLs.

- ***What techniques are required to support co-simulation across domains in the MBSE tool-chains? How do system developers make use of these techniques when using the MBSE tool-chains?***

  In this thesis, FMI is selected as middleware to support model exchange in co-simulation for system level verification as shown in Section 3.3.2. Several specific domain simulation tools are integrated in one co-simulation environment for system verification where the models have different data structures and syntax. Moreover, system of system level verification is also required for CPS development. Therefore, High-level architecture (HLA) is another technique considered in this thesis. As shown in **Paper D-F**, Certi RTI (HLA RTI) and Simulink are adopted as two masters to manipulate co-simulations among FMUs.

- ***What techniques are required to support design automation in the MBSE tool-chains? How do system developers make use of these techniques when using the MBSE tool-chains?***

  Three aspects are considered to support design automation in the MBSE tool-chains: 1) a model-driven process management approach; 2) design ontology for supporting service orchestration; 3) automated decision-makings for parameter value selections based on service-oriented technical resources. The detailed explanations are introduced as follow:

  In the MBSE tool-chain, DSM models are first used to formalize development processes and system artifacts of the target products. Then the DSM models are transformed to the ontology which is used to generate a WPMS (Section 3.3.4.1). The WPMS supports stakeholders to manage and monitor their development process and implement their own work tasks automatically. In three case studies (**Paper D-F**), the WPMS is generated from the DSM models which support co-design of aero-engine development, automated co-simulation and automated parameter value selection for auto-braking systems.

  A scenario-based ontology design approach is used to formalize service orchestration in the MBSE tool-chains in order that the WPMS links to service-oriented technical resources for implementing stakeholders' work, as shown in Section 3.3.4.2 (see details in **Paper H**). For each case study, an ontology is designed based on OWL to represent dependencies between the development process, related design information and required technical resources. Such dependencies represent service orchestrations from WPMS to service-oriented technical resources. Then service orchestration supports automated co-simulation implementations based on these ontologies in the MBSE tool-chains.

  In the MSBE tool-chains, a decision-making algorithm is developed to support automated parameter value selection for an auto-braking system (Section 3.3.4.3). The algorithm obtains and analyzes simulation results of Simulink models in the previous phases, and makes decisions for process executions in the WPMS and parameter spaces for co-simulation models in the later phases. The parameter spaces refer to collections of the selected value for the design parameters as shown in Section 3.3.4.3 (see details in **Paper F**).

RQ3: How can MBSE tool-chains be assessed, particularly on MBSE capabilities and Levels of Information Systems Interoperability (LISI)?

Based on systems thinking, the MBSE tool-chains are first formalized using DSM models. Then the DSM models are transformed to visualizations and BN models in order that the MSBE tool-chains are analyzed using qualitative and quantitative approaches.

- ***What compositions and capabilities in the MBSE tool-chains are required to be formalized? How are such compositions and capabilities formalized?***

  In Section 3.1.4, a conceptual model is proposed to formalize MBSE tool-chains (see details in **ARPaper 11**). It defines compositions and interrelationships of MBSE tool-chains using a systems thinking approach. In order to formalize this conceptual model, a DSM approach is proposed whose meta-models are developed to represent tool-chain compositions, their interactions and tool capabilities in **Paper C**.

- ***What measurements are required to consider when assessing the MBSE tool-chains? How are such measurements formalized?***

  In Section □, measurements in Prescriptive Study and Descriptive Study II are introduced. A measurement of TMCL is defined in order to assess MBSE tool-chains based on a quantitative analysis (see details in **Paper C**), related to interoperability, social procedures, application (MBSE capability) and infrastructure. In order to formalize the measurements of TMCL, a DSM approach is adopted to describe the measurements using developed meta-models. Moreover, several measurements are defined to assess tool-chains using visualizations, such as tool-chain complexities defined in Section 2.4.4 which support qualitative analysis of MSBE tool-chains. Finally, in each case study during Descriptive Study II, specific measurements related to their target product development are defined separately in order to assess the corresponding tool-chain prototypes.

- **What techniques are used to support MBSE tool-chain assessment? How do the tool-chain developers assess the MBSE tool-chain using them?**

  A DSM approach integrated with visualization and BN models is proposed to assess MBSE tool-chains using a quantitative and qualitative approach. First, MBSE tool-chains and metrics related to TMCL measurement are formalized by DSM models. Then the DSM models are transformed to visualizations for qualitative analysis and BN models for quantitative analysis. Using qualitative and quantitative analysis, the developers understand the effects of MBSE tool-chains and make decisions for selecting MBSE tool-chain prototype candidates before developing them.

4.2 MBSE Tool-chain Prototypes

In verifications of Descriptive Study II of the third round study, three case studies are separately used to evaluate the success KPIs and application KPIs as shown in Table 3.1 (Section 3.5). In each case study during the operation phase, based on the SPIRIT framework, the MBSE tool-chains are developed using techniques described in Table 3.2 (Section 3.5). Moreover, success and application KPIs are evaluated in **Paper D-Paper F,** using the evaluation methods mentioned in Table 2.4 (Section 2.4.4). From the results, the success KPIs are satisfied by each tool-chain and related application KPIs are promoted as well. Details are seen in Table 4.1.

Table 4.1 Results of case studies

KPI type	**KPIs (**Table 3.1**)**	Case1 (Table 3.1, see details in **Paper D**)	Case2 (Table 3.1, see details in **Paper E**)	Case3 (Table 3.1, see details in **Paper F**)
Success KPIs	Co-design of aero-engine performance analysis in Case 1.	A case study is presented to illustrate the MBSE tool-chain supports a co-design process of co-simulation for aero-engine performance analysis.		
	Automated co-simulation based on HLA and FMI in Case2.		A case study is presented to illustrate the MBSE tool-chain generates java codes automatically for supporting co-simulation using jCerti and jFMI.	
	Automated value selections of design parameters for auto-braking systems in Case3.			A case study is presented to illustrate an automated decision-making algorithm for value selections of design parameters based on simulation models in Simulink for following co-simulation models.
Application KPIs	Levels of Communication between Stakeholders (Be evaluated by methods in Table 2.4. Other application KPIs are the same).	• DSM models formalize development processes and system artifacts of aero-engine development. • DSM models represent deployments of co-simulations between Simulink and FMUs.	• DSM models represent dependencies and traceability between development processes and system artifacts. • DSM models represent deployment of co-simulations based on jCerti and FMUs.	• DSM models formalize development processes and system artifacts of auto-braking system development. • DSM models formalize decision-making algorithms for automated value selections of design parameters. • DSM models represent deployment of co-simulations between Simulink, Carmaker and FMUs.
	Capability of Process Management.	WPMS generated based on BPM Camunda is used to support co-design of aero-engine performance analysis.	WPMS generated based on BPM Camunda is used to deploy Certi RTI for co-simulations among FMUs automatically.	WPMS generated based on BPM Camunda is used to select parameters for co-simulations automatically.
	Interoperability of Integrated System Simulation.	Co-simulation between Simulink and FMUs for aero-engine performance analysis.	Co-simulation between Certi RTI and FMUs in Case 2.	Co-simulation between Simulink, Carmaker and FMUs for auto-braking system development.
	Traceability among development Processes, related Information and Technical resources (models, data and tools).	OSLC services supports WPMS linked with FMUs, Simulink models and other technical resources.	OSLC services supports WPMS linked with FMUs, Certi RTI and other technical resources.	OSLC services supports WPMS linked with FMUs, Carmaker models, Simulink models and other technical resources.
	Tool Interoperability.	13 tool operations supported by OSLC services.	11 tool operations supported by OSLC services.	14 tool operations supported by OSLC services.
	Efficiency of CPS development.	Less individual efforts on manual operations for developing and configuring co-simulations.	Less individual efforts on generating codes for developing and configuring co-simulations.	Using less times of co-simulations compared with co-simulations without automated value selections of design parameters.

4.3 Threats to Validity

This section discusses the threats to the validity of all the appended papers. The research results are internally validated based on the case studies provided by partners, namely, Nanjing University of Aeronautics and Astronautics, SAFER [96], Suzhou Tongyuan Software & Control Tech. Co., Ltd and Beijing Z.K. Fengchao Tech. Ltd.

4.3.1 General threats to the validity in the thesis

The hypothesis of this thesis is to develop a framework to support CPS tool-chain development using a service-oriented and MBSE approach. Under this framework, the tool-chains aim to support CPS development more efficiently. However, there are several threats to the validity of this thesis, most importantly: 1) The general risk that the case studies are not representative enough for CPS; and 2) The specific risk that software-intensive systems are not represented well enough as part of the selected case studies for verifying MBSE tool-chains.

Firstly, three case studies from different domains are defined to verify my ideas in the internal validity. More case studies are used to assess the external validity, such as automotive and IoT systems. Currently there is another concept, called Model-based Software Engineering, which is considered as a different concept from Model-based Systems Engineering. Though software components are not treated as internal validity, an OSLC configuration tool, *Datalinks* is used to support interface (software components) development of IoT systems for the external validity. Moreover, open standards, such as OSLC and FMI, provide potentials to support design of CPS components including both software and hardware. The use of open standards is one solution for integrating these two concepts.

4.3.2 Paper A

The hypothesis of **Paper A** is that it is possible to find out the nature of MBSE tool-chains and to identify key techniques of MBSE for CPS system development using a questionnaire survey. Before designing the questionnaire, literature reviews and MBSE practices are implemented in the first and second rounds. Thus, the questionnaire is developed based on lessons-learned and literature reviews from the first two rounds.

The main threats to the validity of this work are threefold: 1) The risk that the scope of the questionnaire cannot cover all the challenges and benefits of MBSE tool-chains. 2) The risk that samples are not representative enough of general situations in industry because of the low response rate. 3) The risk that the scope of the respondents restricts the expected results if all the respondents are from Chinese industry.

In the questionnaire, multiple choice questions are mostly designed with some open answers which can add additional content and comments. Moreover, this questionnaire survey aims to achieve a broad enough coverage to reach the few engineers who understand MBSE. From the survey results, we find most of the respondents are the high-level managers or team leaders in their own areas. Therefore, it is inferred that their answers can be used to analyze the general situations of their teams, even the companies they are in. In order to analyze the survey results from an extended scope, a more comprehensive analysis of literature reviews and questionnaire is implemented in **ARPaper1**. From the comparisons between literature reviews and the questionnaire survey, it is inferred that current industrial concerns are the trends which academia directly focuses on.

4.3.3 Paper B

The hypothesis of **Paper B** is that if tool chain developers consider one MBSE tool-chain as a system, a framework can be developed to capture complete information for tool-chains based on systems thinking. Tool-chain compositions and their interactions are considered together to support MBSE tool-chain development.

The main threat to the validity of **Paper B** is the completeness of verification of the defined framework. The verification refers to if the framework enables tool-chains to support CPS development more efficiently, as we expect.

Three case studies are defined to develop, implement and evaluate MBSE tool-chain prototypes in **Paper D-F**. The effects of the target product development implemented by such prototypes are considered as the main KPIs for this framework. By implementing these prototypes, MBSE tool-chains are evaluated by comparing their efforts with the efforts of not using them.

4.3.4 Paper C

The hypothesis of **Paper C** is that if the capabilities and architectures of MBSE tool-chains are formalized and analyzed, models can be used to assess MBSE tool-chains using qualitative and quantitative approaches. In this paper, a DSM approach is proposed to support early verification of MBSE tool-chains before prototyping.

The main threats to the validity of **Paper C** include: 1) The risk of assessing KPIs using incomplete metrics based on MBSE capability [44] and Levels of Information Systems Interoperability (LISI) [15]. 2) Stakeholders may not be able to obtain the expected results from the assessments when MBSE capabilities and LISI are only used to assess individual tools without assessing the entire tool-chain. 3) The risk that tool-chain developers do not obtain the expected results using assessment based on probability.

Using the DSM approach, meta-models are used to formalize capabilities and compositions of MBSE tool-chains to assess MBSE tool-chains. If more metrics can be designed, their corresponding meta-models can be developed for the future measurements. In order to promote confidences of tool-chain developers, visualizations are also used to support qualitative analysis of MBSE tool-chains.

4.3.5 Paper D

Paper D presents the first MBSE tool-chain prototype for aero-engine performance analysis. This paper aims to evaluate the effects of MBSE tool-chains on co-design processes which are used to verify the prototype developed based on the SPIRIT framework.

The main threats to the validity of **Paper D** include: 1) The risk that the effects of the given MBSE tool-chain in the case studies cannot evaluate the framework systematically. 2) The risk that the developed service orchestration is not generalized for other specific scenarios during supporting co-simulation; and 3) The risk that measurements cannot satisfy the demand of evaluating the developed MBSE too-chain, which are implemented based on individual human behaviors. Efforts of individuals when operating the prototype are compared with their efforts without it.

The tool-chain is developed based on the SPIRIT framework. In the thesis, the efficiency of the co-design process with and without using the MBSE tool-chain is compared using quantitative and qualitative approaches. From the results in the paper, automated implementations of tool APIs have obviously promoted the efficiency. In order to evaluate this framework in a systematic way, two other case studies of co-simulation are provided in **Paper E** and **Paper F**. Moreover, an ontology-based approach is proposed to design service orchestrations for different scenarios in **Paper H**. In order to verify the prototype in a systematic way, several COTS tools are developed which are used in industry as described in **Paper G** and **ARPaper 2**.

4.3.6 Paper E

Paper E presents the second MBSE tool-chain prototype for automated co-simulation based on HLA and FMI. This paper aims to evaluate the effects of MBSE tool-chains on an automated co-simulation process which is used to illustrate the performances of the tool-chain developed based on the SPIRIT framework.

The main threats to the validity of **Paper E** include: 1) The risk that accuracy of co-simulation is not a main concern in this paper. 2) The risk that efforts on coding are mainly measured as KPIs to illustrate the performance of tool-chains when comparing with manual coding for co-simulation executions based on HLA and FMI,.

Several open source engines are used to construct the MBSE tool-chains for co-simulations. Using these techniques, errors between the results from the prototype and a commercial tool are analyzed and found to be acceptable. Moreover, a qualitative analysis of tool operations is also used to assess the prototype compared with manual operations. The implementations of HLA are executed automatically which promote the efficiency of configuring the co-simulations.

4.3.7 Paper F

Paper F presents the third MBSE tool-chain prototype supporting automated value selections of design parameters. This paper aims to evaluate the effects of MBSE tool-chains on auto-braking system development using an automated decision-making algorithm.

The main threat to the validity of **Paper F** is the risk that scope of the developed decision-making algorithm is limited to the specific cases of auto-braking system development.

The meta-models developed in **Paper F** can formalize other scenarios and are not limited to auto-braking system development. The decision-making algorithms for other scenarios can be defined as new values of the property "goal function" in the meta-models in order that the meta-models can describe more algorithms without rebuilding meta-models (see details in **Paper F**).

4.3.8 Paper G

Paper G presents the *Karma* language based on the GOPPRR meta-meta models for MBSE formalisms, such as SysML and UML. It introduces an overview of the language and the workflow to support MBSE in one COTS tool, *MetaGraph*.

The main threat to validity for **Paper G** is the risk of incompleteness of the proposed language. Currently, this language aims to support MBSE formalisms, architecture-driven, code generation, formal verification based on satisfiability modulo theories and behavior simulation based on automata theory.

The concrete syntax and abstract syntax are designed based on current practices. Several industrial evaluations are ongoing to verify this language and provide more feedbacks to promote this language. For example, meta-models of SysML, UML and BPMN are built based on *Karma* language and several domain cases are used to evaluate the corresponding aspects.

4.3.9 Paper H

Paper H presents an OWL-based approach to support ontology design for service orchestration. The service orchestration is used to deploy and to manage the technical resources from WPMS. The approach is evaluated through an MBSE tool-chain prototype for supporting one scenario of co-simulation used in **Paper D.**

The main threat to validity for **Paper H** is the potentially limited scalability of the approach.

Systems thinking is used to develop the ontology based on different scenarios. Before using OWL to design the ontology, different scenarios are defined and analyzed to understand compositions and their interrelationships. Based on systems thinking, more scenarios can be analyzed and used to develop ontology for service orchestrations in different tool-chains.

4.3.10 External Validity

Large-scale external validations by industry are supported by Beijing Z.K. Fengchao Tech. Co. Ltd[6]. They developed four COTS tools to support the SPIRIT framework: 1) a DSM tool, *MetaGraph*; 2) an OSLC configuration tool, *Datalinks*; 3) a co-simulation tool based on FMI, *Prajna*; 4) a data visualization tool supporting OSLC, *DataVis*. These four tools have been used by the industrial partners of Z.K. Fengchao who provided feedbacks for improving the future research through the development and use of four COTS tools, introduced separately below.

MetaGraph is a DSM tool based on the *Karma* language. Feedbacks from Chinese industry, suggests that this technique is very interesting and expected to be followed up on for one reason, namely that: current MBSE solutions provided by several tool suppliers are mainly focuses on SysML, UML, OPM [97] and Arcadia [98]. Moreover, some tool suppliers consider that PLM, PDM and simulation tools based on co-simulation and Modelica are also parts of the MBSE. Thus, industry is confused about what is MBSE and how to develop their target products using MBSE. Using *Karma*, such MBSE languages can be formalized. From the integration views, engineers in industry are anticipating some unified language to support their work, not only middleware or tools.

We find *MetaGraph* has been used to build meta-models of SysML, UML, BPMN and UPDM for developing the related models. Several partners of Z.K. Fengchao have adopted this DSM tool to support their work, which includes the domains of automotive and IoT. The concerns they have when using the *Karma* language are twofold: 1) The maturity level of *MetaGraph*- whether it can support engineers to build models under a friendly human-machine interface. 2) The completeness of the *Karma* language- If the syntax of the language can satisfy all the industrial requirements.

Though current partners of *MetaGraph* are mainly from China, another DSM tool-*MetaEdit+* [13] based on the GOPPRR approach is widely used in the industry in Europe and the US. From these industrial practices, we could infer that tools based on the GOPPRR approach satisfy the demands of MBSE required by a wide range of industries.

Datalinks and *DataVis* have been used to support smart city by Z.K. Fengchao. Signals and data from cameras and drones are integrated and collected through OSLC adapters developed based on *Datalinks*. Then the data is visualized in *DataVis*, which aims to support safety monitoring in the construction fields. One concern from the industry is whether OSLC specifications can be widely used in the future. From the observations, several tools supporting tool-integration based on OSLC has been developed, such as Smartfacts[7] and IBM Doors [99]. I infer OSLC, at least, has the potential to support tool integration for MBSE. From my personal opinion, the reason why OSLC has not been widely used, is that current PDM and PLM systems developed by big companies can support tool-integration

---

6 http://www.zkhoneycomb.com/, a software company for MBSE.

7 https://www.smartfacts.com/

of specific tools. Tool-integration is not the main challenge of industry when they implement their workflows, because their current concerns are mainly focused on their products.

Prajna is an integrated simulation tool to support co-simulation based FMI, distributed simulation based on HLA and fault tree analysis based on BN models. From current feedback, co-simulation tools have been widely used in different industries, such as the open source co-simulation tool provided by INTO-CPS [100]. Most of the engineers think it is an interesting topic to consider how to integrate SoS simulation and co-simulation for systems.

5 Conclusion and Future Work

In this chapter, we offer conclusions and outline possibilities for future work.

5.1 Conclusion Summary

The main goal of this thesis has been to develop a framework for constructing CPS tool-chains using a service-oriented and MBSE approach. A literature review and questionnaire survey have been adopted to investigate earlier research (how MBSE tool-chains were developed), the current state-of-the-art of techniques for developing them and the future technical trends.

Regarding the goals mentioned in Section 1.3.1, **Paper A** identifies benefits and challenges of MBSE tool-chains from the results of the questionnaire. **Paper B** proposes the SPIRIT framework for MBSE tool-chain development using systems thinking and service-oriented approaches. Using this framework, tool-chain developers are required to consider social, process, information and technical perspectives for developing their MBSE tool-chains. Moreover, a service-oriented approach is used to support tool-integration of the entire tool-chains. Furthermore, two main techniques are highlighted: 1) **Paper G** proposes a DSM approach based on GOPPRR supporting MBSE formalisms; 2) **Paper H** proposes an ontology design approach for service orchestrations in the MBSE tool-chains. In order to support early evaluation of MBSE tool-chain prototypes, **Paper C** demonstrates a DSM approach with BN models and visualizations to assess the MBSE tool-chains using metrics based on MBSE capabilities and LISI. Stakeholders adopt DSM models to formalize MBSE tool-chains and MBSE capabilities and interoperability of their tools. Then the DSM models are transformed to visualizations and BN models for qualitative and quantitative analysis in order to evaluate MSBE tool-chains before prototyping. **Paper D**, **Paper E** and **Paper F** demonstrate three prototypes for co-design of aero-engine performance analysis, automated co-simulation based on HLA and FMI and automated value selections of parameters for auto-braking system development. From the results in Section 4.2, we find the prototypes promote efficiency of the target product development compared with manual operations without using our MSBE tool-chains.

5.2 Future work

The review of research questions shows opportunities to continue the research on the SPIRIT framework for MBSE tool-chain development. The current research will be extended in the following ways:

5.2.1 A General Domain-specific Modeling Language to Support Satisfiability Checking

Satisfiability Checking is a process of checking the existence of a satisfying solution for a logical formula [101]. The *Karma* language, proposed in **Paper G**, would be expected to support satisfiability checking based on Satisfiability Modulo Theories (SMT) formulas [102]. Moreover, the *MetaGraph*[8] would be developed to support SMT solver implementations for the *Karma* language in the future.

[8] One DSM tool, MetaGraph developed by Z.K. Fengchao, http://www.zkhoneycomb.com/.

5.2.2 Reinforcement Learning to Support Decision-makings

Based on OSLC RDF representing technical resources for CPS development, reinforcement learning algorithms would be of interest to investigate for how to support decision-making in a web-based process management system. The experiential data from different domains, such as simulation results, will be considered together to train the reinforcement learning models with the goal of more complex decisions for the development processes being made automatically in the future.

6 Bibliography

[1] H. Langseth and L. Portinale, “Bayesian networks in reliability,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 1, pp. 92–108, 2007.

[2] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, “Co-Simulation,” *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–33, May 2018.

[3] S. J. I. Herzig, A. Qamar, A. Reichwein, and C. J. J. Paredis, “A Conceptual Framework for Consistency Management in Model-Based Systems Engineering,” in *31st Computers and Information in Engineering Conference*, 2011, no. July 2015, pp. 1329–1339.

[4] D. Carney, S. A. Hissam, and D. Plakosh, “Complex COTS based software systems: practical steps for their maintenance,” *J. Softw. Maint. Res. Pract.*, 2002.

[5] M. Törngren and P. Grogan, “How to Deal with the Complexity of Future Cyber-Physical Systems?,” *Designs*, vol. 2, no. 4, p. 40, Oct. 2018.

[6] E. A. Lee, “CPS foundations,” in *Proceedings of the 47th Design Automation Conference*, 2010, p. 737.

[7] S. Kelly and J.-P. P. Tolvanen, *Domain-Specific Modeling: Enabling Full Code Generation*, Wiley-IEEE., no. 3. Wiley-IEEE Computer Society Press, 2008.

[8] A. Qamar, “Model and dependency management in mechatronic design,” KTH Royal Institute of Technology, 2013.

[9] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[10] A. Geraci, F. Katki, L. McMonegal, B. Meyer, and H. Porteous, “IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries,” 1991.

[11] J. Lu, Y. Wen, Q. Liu, D. Gürdür, and M. Törngren, “MBSE Applicability Analysis in Chinese Industry,” *INCOSE Int. Symp.*, vol. 28, no. 1, pp. 1037–1051, Jul. 2018.

[12] Modelica Association Project “FMI,” “Functional Mock-up Interface for Model Exchange and Co-Simulation,” 2013.

[13] S. Kelly, K. Lyytinen, and M. Rossi, “MetaEdit+ A fully configurable multi-user and multi-tool CASE and CAME environment,” in *8th International Conference on Advanced Information Systems Engineering, CAiSE’96*, 1996, vol. 1080, pp. 1–21.

[14] K. Symington, Susan and Morse, Katherine L and Petty, “IEEE Standard for Modeling and Simulation High Level Architecture (HLA)-Federate Interface Specification,” 2001.

[15] C4ISR Architecture Working Group, “Levels of information systems interoperability (LISI),” 1998.

[16] INCOSE, *Systems Engineering Handbook 4E*. 2015.

[17] N. Phojanamongkolkij *et al.*, “Modeling to Mars: a NASA Model Based Systems Engineering Pathfinder Effort,” in *AIAA SPACE and Astronautics Forum and Exposition*, 2017, pp. 1–15.

[18] B. G. Pioso, “Middle-ware interface status tool and method for using same,” 2009.

[19] IEEE., “Towards a Definition of the Internet of Things (IoT),” 2015.

[20] K. Bauer, “KPIs - The Metrics That Drive Performance Management,” *DM Rev.*, 2004.

[21] IEEE, *ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes*. 2008.

[22] A. G. Ryman, A. J. Le Hors, and S. Speicher, “OSLC resource shape a language for defining constraints on linked data,” in *CEUR Workshop Proceedings*, 2013, vol. 996, pp. 1–5.

[23] CertiProject, “jCerti- java Open Source HLA RTI,” 2010. [Online]. Available: http://savannah.nongnu.org/projects/certi.

[24] J. Lu, J. Wang, D. Chen, J. Wang, and M. Torngren, “A Service-Oriented Tool-Chain for Model-Based Systems Engineering of Aero-Engines,” *IEEE Access*, vol. 6, pp. 50443–50458, 2018.

[25] T. Erl, *Service-Oriented Architecture:Concepts, Technology, and Design*. 2005.

[26] J. Lu, D. Chen, J. Wang, and M. Torngren, “Towards A Service-oriented Framework for MBSE Tool-chain Development,” in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, 2018, pp. 568–575.

[27] N. Kumar and S. Kumar, “Querying RDF and OWL data source using SPARQL,” in *2013*

Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–6.
[28] I. Horrocks *et al.*, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," 2004.
[29] "What is systems thinking? - Definition from WhatIs.com.," 2019. [Online]. Available: https://searchcio.techtarget.com/definition/systems-thinking.
[30] IEEE, "IEEE Std 1471-2000, Systems and software engineering —Recommended practice for architectural description of software-intensive systems," 2011.
[31] P. Benjamin, M. Patki, and R. Mayer, "Using Ontologies for Simulation Modeling," in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 1151–1159.
[32] M. Törngren, "Platforms4CPS," 2018. [Online]. Available: https://platforum.proj.kth.se/tiki-view_blog_post.php?postId=40.
[33] M. Törngren and U. Sellgren, "Complexity Challenges in Development of Cyber-Physical Systems," in *Simulation and Modeling of Systems of Systems*, 2018, pp. 478–503.
[34] J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," *Sci. Comput. Program.*, vol. 89, no. PART B, pp. 144–161, Sep. 2014.
[35] J. Heidrich, R. Van Lengen, and T. Kuhn, "Systems Engineering Study: Challenges and Best Practices," 2016.
[36] H. Andersson, "Aircraft Systems Modeling - Model Based Systems Engineering in Avionics Design and Aircraft Simulation," Linköpings universitet, Linköping, 2009.
[37] N. Subcommittee, Model-Based Engineering, "Final Report," 2011.
[38] B. Beihoff *et al.*, "A World in Motion Systems Engineering Vision 2025," in *International Council on Systems Engineering*, 2010, vol. 327, no. 5970, pp. 1183–1183.
[39] INCOSEUK, "What is Model Based Systems Engineering," 2012.
[40] J. Lu, D. Chen, D. Gürdür, and M. Törngren, "An Investigation of Functionalities of Future Tool-chain for Aerospace Industry," *INCOSE Int. Symp.*, vol. 27, no. 1, pp. 1408–1422, Jul. 2017.
[41] W. Schamai, P. Fritzson, C. J. J. Paredis, and P. Helle, "ModelicaML value bindings for automated model composition," *Proc. 2012 Symp. Theory Model. Simul. - DEVS Integr. M&S Symp.*, pp. 31:1--31:8, 2012.
[42] J. Lu, D. Gürdür, D.-J. Chen, J. Wang, and M. Törngren, "Empirical-Evolution of Frameworks Supporting Co-simulation Tool-Chain Development," in *Advances in Intelligent Systems and Computing*, vol. 745, 2018, pp. 813–828.
[43] H. G. Sillitto, "Design principles for Ultra-Large-Scale (ULS) Systems," *INCOSE Int. Symp.*, vol. 20, no. 1, pp. 63–82, Jul. 2010.
[44] B. Friedland, J. Herrold, G. Ferguson, and R. Malone, "Conducting a Model Based Systems Engineering Tool Trade Study Using a Systems Engineering Approach," *INCOSE Int. Symp.*, vol. 27, no. 1, pp. 1087–1099, Jul. 2017.
[45] A. A. Shah, A. A. Kerzhner, D. Schaefer, and C. J. J. Paredis, "Multi-view Modeling to Support Embedded Systems Engineering in SysML," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5765 LNCS, 2010, pp. 580–601.
[46] E. Syriani, H. Vangheluwe, R. Mannadiar, C. Hansen, V. Mierlo, and H. Ergin, "AToMPM:A Web-based Modeling Environment," in *Joint proceedings of MODELS'13 Invited Talks, Demonstration Session, Poster Session, and ACM Student Research Competition co-located with the 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013)*, 2013, pp. 21–25.
[47] K. G. Young, "Defense Space Application of MBSE - Closing the Culture Chasms," in *AIAA SPACE 2015 Conference and Exposition*, 2015, pp. 1–11.
[48] F. A. Díaz González, P. R. Pinzón Cabrera, and C. M. Hernández Calderón, "Design of a Nanosatellite Ground Monitoring and Control Software - a Case Study," *J. Aerosp. Technol. Manag.*, vol. 8, no. 2, pp. 211–231, May 2016.
[49] W. S. A and B. C, "BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation," Future Strategies Inc, 2010.
[50] M. Abramovici, "Future Trends in Product Lifecycle Management (PLM)," in *The Future of Product Development*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 665–674.

[51] E. J. Vidal and E. R. Villota, “SysML as a Tool for Requirements Traceability in Mechatronic Design,” in *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering - ICMRE 2018*, 2018, pp. 146–152.
[52] A. Leitner, B. Herbst, and R. Mathijssen, “Lessons Learned from Tool Integration with OSLC,” in *ICIST 2016: Information and Software Technologies*, 2012, vol. 319, pp. 242–254.
[53] G. Burgio *et al.*, “Framework for Modelling and Simulation of Multi-Physics Aircraft Systems with Distributed Electronic Controllers,” in *SAE Technical Paper Series*, 2017, vol. 1.
[54] M. Biehl, “A Modeling Language for the Description and Development of Tool Chains for Embedded Systems,” KTH Royal Institute of Technology, 2013.
[55] J. D’Ambrosio and G. Soremekun, “Systems engineering challenges and MBSE opportunities for automotive system design,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, vol. 2017-Janua, pp. 2075–2080.
[56] D. Gürdür, J. El-Khoury, T. Seceleanu, and L. Lednicki, “Making interoperability visible: Data visualization of cyber-physical systems development tool chains,” *J. Ind. Inf. Integr.*, vol. 4, pp. 26–34, Dec. 2016.
[57] T. J. Bayer, M. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins, and S. Mandutianu, “Concept of operations for integrated model-centric engineering at JPL,” *IEEE Aerosp. Conf. Proc.*, 2011.
[58] S. Nejati, M. Sabetzadeh, D. Falessi, L. Briand, and T. Coq, “A SysML-based approach to traceability management and design slicing in support of safety certification: Framework, tool support, and case studies,” *Inf. Softw. Technol.*, vol. 54, no. 6, pp. 569–590, Jun. 2012.
[59] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, “Co-simulation: State of the art,” Feb. 2017.
[60] A. DeHon and J. Wawrzynek, “Reconfigurable computing,” in *Proceedings of the 36th ACM/IEEE conference on Design automation conference - DAC ’99*, 1999, no. c, pp. 610–615.
[61] PLM4MBSE Working Group/INCOSE, “10 theses about MBSE and PLM,” 2015.
[62] A. C. Lucienne T.M. Blessing, *DRM: A Design Reseach Methodology*, Springer. London: Springer, 2009.
[63] T. Maqsood, A. D. Finegan, and D. H. T. Walker, “Five case studies applying Soft Systems Methodology to Knowledge Management,” *7th Annu. Conf. Syst. Eng. Res.*, p. 18, 2009.
[64] H. W. Lawson, *A journey through a systems landscape*. College Publications, 2010.
[65] J. Mingers, “Towards an Appropriate Social Theory for Applied Systems Thinking: Critical Theory and Soft Systems Methodology,” *J. Appl. Syst. Anal.*, vol. 7, pp. 41--50, 1980.
[66] M. Simulink and M. A. Natick, “The mathworks,” MATHWORKS, 1993.
[67] K. KELLEY, “Good practice in the conduct and reporting of survey research,” *Int. J. Qual. Heal. Care*, vol. 15, no. 3, pp. 261–266, May 2003.
[68] J. Wu, “From Wechat To We Fight: Tencent and China Mobile’s Dilemma,” in *PACIS 2014 Proceedings*, 2014, p. 265.
[69] R. J. Torraco, “Writing Integrative Literature Reviews: Guidelines and Examples,” *Hum. Resour. Dev. Rev.*, vol. 4, no. 3, pp. 356–367, Sep. 2005.
[70] B. Kitchenham *et al.*, “Systematic literature reviews in software engineering – A tertiary study,” *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, Aug. 2010.
[71] G. D. Beecroft, G. L. Duffy, and J. W. Moran, *The executive guide to improvement and change*. ASQ Quality Press, 2003.
[72] OASISOpenProject, “Open Services for Lifecycle Collaboration Core Specification Version 3.0,” 2018.
[73] T. Blochwitz *et al.*, “Functional Mockup Interface 2 . 0 : The Standard for Tool independent Exchange of Simulation Models,” *9th Int. Model. Conf.*, 2012.
[74] M. U. Awais, P. Palensky, A. Elsheikh, E. Widl, and S. Matthias, “The High Level Architecture RTI as a Master to the Functional Mock-up Interface Components,” in *2013 International Conference on Computing, Networking and Communications (ICNC)*, 2013, pp. 315–320.
[75] J. Freund and B. Rücker, *Real-Life BPMN: Using BPMN 2.0 to Analyze, Improve, and Automate Processes in Your Company*. 2012.
[76] L. Mao, B. Zheng, G. Zhao, and L. Liu, “Understanding Geographical Space with Big Data: A Network Perspective,” *J. Digit. Inf. Manag.*, vol. 13, no. 5, pp. 354–360, 2015.

[77] M. Golby, "Case study method: key issues, key texts," *Teach. Dev.*, vol. 5, no. 3, pp. 403–408, Oct. 2001.
[78] A. J. Onwuegbuzie, "Framework for Internal and External Validity," 2000.
[79] D. Broman, E. A. Lee, S. Tripakis, and M. Törngren, "Viewpoints, formalisms, languages, and tools for cyber-physical systems," in *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling - MPM '12*, 2012, vol. 1, no. 212, pp. 49–54.
[80] C. Tschirner, L. Bretz, R. Dumitrescu, and J. Gausemeier, "Applying Model-Based Systems Engineering for Product Engineering Management concepts for industrial application," in *2015 IEEE International Symposium on Systems Engineering (ISSE)*, 2015, pp. 42–49.
[81] A. Hall, "Three-Dimensional Morphology of Systems Engineering," *IEEE Trans. Syst. Sci. Cybern.*, vol. 5, no. 2, pp. 156–160, 1969.
[82] J. Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy," *J. Inf. Sci.*, vol. 33, no. 2, pp. 163–180, Apr. 2007.
[83] J. Lu, M. Törngren, D.-J. Chen, and J. Wang, "A Tool Integration Language to Formalize Co-simulation Tool-Chains for Cyber-Physical System (CPS)," *Softw. Eng. Form. Methods*, pp. 391–405, 2018.
[84] H. Kern, A. Hummel, and S. Kühne, "Towards a comparative analysis of meta-metamodels," in *Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE!'11, AOOPES'11, NEAT'11, & VMIL'11 - SPLASH '11 Workshops*, 2011, vol. 1, p. 7.
[85] Modelon, "FMI Toolbox for Matlab," 2014. .
[86] X. Chen and Z. Wei, "A New Modeling and Simulation Platform-MWorks for Electrical Machine Based on Modelica," in *International Conference onElectrical Machines and Systems, 2008. ICEMS*, 2008, pp. 4065–4067.
[87] C. Stenzel, C. Stenzel, S. Pawletta, and S. Pawletta, "CERTI - Bindings to Matlab and Fortran," *Engineering*, 2008.
[88] O. Evora, J., Hernandez, J. J., & Roncal, "JavaFmi." [Online]. Available: https://bitbucket.org/siani/javafmi/wiki/Home.
[89] R. J. Allen, D. Garlan, and J. Ivers, "Formal modeling and analysis of the HLA component integration standard," *ACM SIGSOFT Softw. Eng. Notes*, vol. 23, no. 6, pp. 70–79, Nov. 1998.
[90] M. Geiger, S. Harrer, J. Lenhard, M. Casar, A. Vorndran, and G. Wirtz, "BPMN Conformance in Open Source Engines," in *2015 IEEE Symposium on Service-Oriented System Engineering*, 2015, pp. 21–30.
[91] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, "A Comparison of a Graph Database and a Relational Database," in *Proceedings of the 48th Annual Southeast Regional Conference on - ACM SE '10*, 2010, p. 1.
[92] OMG, "Software & Systems Process Engineering Meta-Model Specification V2.0," 2008.
[93] M. Dinar and D. W. Rosen, "A Design for Additive Manufacturing Ontology," *Vol. 1B 36th Comput. Inf. Eng. Conf.*, vol. 17, no. June 2017, p. V01BT02A032, 2016.
[94] O. Community, "OSLC Primer," Dec. 2012.
[95] D. Wu, D. W. Rosen, L. Wang, and D. Schaefer, "Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation," *Comput. Des.*, vol. 59, pp. 1–14, Feb. 2015.
[96] D. Chen, "SAFER SARMITS project." [Online]. Available: https://www.saferresearch.com/projects/sarmits-systematic-approach-rish-management-its-context.
[97] Y. Mordecai, O. Orhof, and D. Dori, "Model-based interoperability engineering in systems-of-systems and civil aviation," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 4, pp. 637–648, 2018.
[98] P. Roques, P. Roques, T. European, and P. Roques, "MBSE with the ARCADIA Method and the Capella Tool," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, 2016, p. 11.
[99] J. Beatty and F. Remo, "How to Evaluate and Select a Requirements Management Tool," *Seilevel Whitepaper*, 2011.
[100] P. G. Larsen *et al.*, "Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project," in *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*, 2016, pp. 1–6.

[101] J. Li, L. Zhang, G. Pu, M. Y. Vardi, and J. He, “LTLf satisfiability checking,” in *Frontiers in Artificial Intelligence and Applications*, 2014.
[102] L. De Moura and N. Bjørner, “Z3: An efficient SMT Solver,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008.
[103] H. Wang, G. Wang, J. Lu, and C. Ma, “Ontology Supporting Model-Based Systems Engineering Based on a GOPPRR Approach,” in *WorldCist'19 - 7th World Conference on Information Systems and Technologies*, Cham: Springer International Publishing, 2019, pp. 426–436.