Nebojša Taušan

CHOREOGRAPHY MODELING IN EMBEDDED SYSTEMS DOMAIN
CHOREOGRAPHY MODELING IN EMBEDDED SYSTEMS DOMAIN

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Technology and Natural Sciences of the University of Oulu for public defence in the OP-Pohjola auditorium (L6), Linnanmaa, on 9 December 2016, at 12 noon
Choreography modelling, as a service-oriented architecture specific technique, is increasingly present in embedded systems development domain. This technique specifies a flow of interactions between participants' services from the global or neutral point of view while the specified models represent an integral part of the overall software architecture. Choreography modelling languages that are currently used in embedded systems domain, however, are not expressive enough to capture the choreography-relevant information in this domain. For this reason, choreography specifications are often lacking information or include ambiguous information. This allows misinterpretation of the specified choreography models and leads to difficulties in communication among stakeholders that use those models.

The objective of this research is to advance the design of choreography modelling languages by identifying the information content that is relevant in embedded systems domain and by designing a choreography modelling language that supports that information content. To achieve this objective, this research adopted the design science research framework and five individual studies were conducted within this framework. These studies used methods such as the interviews with practitioners, company specific documents and open standards to understand the challenges in industry, systematic literature review to collect the existing scientific knowledge about the utilization of choreography in embedded systems and the focus groups to evaluate the designed language. Based on these study results, the information content that is relevant for choreography modelling in embedded systems domain was identified and then supported with the design of choreography modelling language.

The design of the choreography modelling language is evaluated in academic and industry context. The evaluation in academic context is realized by language implementation while the evaluation in industry is realized with industry experts. Language evaluation showed increased expressiveness of the designed language and indicated on possible benefits from its use in testing and protocol development area. These benefits include the reduction of development time and errors in the testing phase while the reduction of maintenance burden and performance improvement can be expected in the protocol development area.

**Keywords:** choreography, choreography modelling language, communication, design science research, embedded software development, embedded systems, qualitative methods, service-oriented architecture, software architecture

Tämän tutkimuksen tavoitteena on edistää koreografiamallinnuksessa käytettävien kielen suunnittelua tunnistamalla ne tietosiä, jotka ovat oleellisia sulautetuille järjestelmile sekä suunnitella kielin, joka tukee oleellisia tietosisällöjä. Tavoitteessa saavutettakseen sovelletaan ”design science” (suunnittelun tutkimus) tutkimusmenetelmää, jolla toteutettiin viisi tapaustutkimusta. Näissä tutkimuksissa hyödynnettiin teollisuuden asiantuntijoiden haastatteluja, yritys-kohtaisia dokumentteja ja avoimia standardeja, joiden avulla pystyttiin ymmärtämään teollisuuden kohtalaisia haasteita. Systemaattisen kirjallisuuskatsauksen avulla kerättiin yhteen olemassa olevia tietoja koreografian käytöstä sulautetuissa järjestelmissä. Kehitetyn kielin sopivuutta teolliseen tuotantotyyliin arvioitiin asianajajaryhmille järjestelyissä työpajoissa. Saatuja tutkimustuloksia valossa koreografiamallinnuksessa tarvittavat oleelliset tietosisällöt sulautettujen järjestelmiin alueella pystyttiin määrittämään sekä kehitetään tietosisällö tukeva koreografian mallinnuskielel.


Asiakirjat: Koreografia, koreografinen mallinnus, laadulliset tutkimusmenetelmät, ohjelmistoarkkitehtuuri, palvelukeskeinen arkkitehtuuri, sulautettujen ohjelmistojen kehitys, sulautetut järjestelmät, suunnitellon tutkimus, vuorovaikutus.
Mojoj porodici...
Preface

Preparing the preface of my Ph.D. thesis made me realise that I am actually writing the last words that will be included in this book and that my journey called PhD studies has come to its end. Reaching this milestone took me six years, and I must admit that these years were equally difficult as they were inspiring. With this short note, I would like to acknowledge the people who were with me on this journey and to express to them my sincerest thank you! Without their support, guidance and friendship, this book would not have been completed.

I would like to thank my supervisors, Professor Markku Oivo, Dr Jouni Markkula and Dr Pasi Kuvaja from Oulu University. Professor Oivo, thank you for giving me the opportunity to be a member of one of the leading software engineering research groups and for your guidance during my studies. Dr Markkula, your patience during our numerous discussions, your understanding of my shortcomings and your endless support for my ideas are something I will always treasure. Dr Kuvaja, thank you for sharing with me your rich experiences in software engineering and for keeping me motivated to complete my studies.

I would also like to thank my follow-up group members, Professor Burak Turhan, Professor Jaakko Sauvola and Dr Vladimir Mandić. Our annual meetings and your feedback on my research were a valuable gauge of my progress and served as the guidepost for my future steps. Thank you for all your efforts.

This thesis was likewise evaluated by two external reviewers. I am respectfully grateful to Associate Professor Mira Kajko-Mattsson from the KHT Royal Institute of Technology, Sweden and to Professor Tommi Mikkonen from the Tampere University of Technology, Finland. Your detailed comments and suggestions allowed me to further improve the quality of the thesis.

Work on this Ph.D. project strongly relied on insights from software companies. For this reason, I would like to acknowledge the assistance of Mr Jari Lehto, Mr Jari Peltonen, Mr David Findlay and Mr Antti Välimäki. Thank you for unselfishly sharing your expertise and experiences and for allowing me to better understand the difference between the science and the ‘black art’ of software development. Additionally, I would like to acknowledge the AMALTHEA and N4S research projects that provided funding.
for my research, as well as the University of Oulu Graduate School that supported two of my conference trips with travel grants.

Special thanks go to all my colleagues from the M3S research group and friends. Sanja, thanks for all the birthdays, synchronised skating and many other events that I spent with you and your beautiful family. Markus, thank you for the ice-fishing activity and our numerous political discussions. Jarkko, I am grateful for your valuable comments and feedback on my work. Davide and Rahul, thank you, as well, for all the (science fiction) movies that we watched together and the epic meal time that followed. Ijaz, I am thankful to you for being such a good flatmate, for introducing me to your culture and for showing me how to prepare excellent Pakistani food. Thank you, Vladimir, Burak, Pilar, Ovais, Itir, Ali, Ayse, Woub, Lucy, Amine, Negar, Adrian, Ivana and Vitor, for the fun time we all had at Uniresta, at Nepalese restaurant, barbecue sessions and other social gatherings. I hope that we all stay in touch.

From time to time, my Ph.D. studies were interrupted by my visits to my hometown Subotica, where my family, friends and former colleagues constantly reminded me of where home is. Hvala Edita na tvojoj beskonačnoj brizi, ukusnoj kuhinji i što si uvek verovala u mene, čak i kada nisam to zaslužio. Tomislave, hvala na podršci koju si mi pružio i koju mi i dalje pružaš. Dragi Istok, hvala što si tokom ovog poduhvata bio uz mene svaki dan. Thank you Ana, Srđan, Tatjana, Aleksandre, Natalija, Ćonka, Ilija and Danilo, for making me feel as if I have never left. I would also like to acknowledge my former colleagues from the Faculty of Economics Subotica, University of Novi Sad. Thank you Prof. Tumbas, Peđa, Marton, Nebojša, Srđan, Vuk, Olivera, Lazar and Mirjana for all the support you provided while I was away.

Finally, I saved the most important and valuable treasures I have for the end. I would like to thank the two most important women in my life, Nada and Tea. Nada, you are the love of my life. Thank you for being my inexhaustible source of hope and my strongest ally in this journey. Tea, your smile is the cutest I have seen. Watching you smile in the cradle fill me with joy and bliss.

I wish you all good health, happiness and prosperity.

In Subotica, Serbia, 10th October 2016
Abbreviations and terminology

AMALTHEA
ITEA2 International research project goal of which was the development of open and comprehensible platform for tool integration.

BPMN
Business Process Model and Notation

Choreography
In SOA-based application, choreography captures the flow of interactions between autonomous service providers or participants form the global or neutral point of view.

DESMET
Methodology for the evaluation of software engineering methods, technologies and tools

DS
Design Science

DSR
Design Science Research

ES
Embedded system. A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function [1].

ICT
Information and Communication Technology

ITEA
The EUREKA Cluster programme supporting innovative, industry-driven, pre-competitive R&D projects in the area of software-intensive systems & Services (https://itea3.org/about-itea.html)

MDE
Model-Driven Engineering. Systematic use of models as primary artefacts during a software engineering process [2].

SA
Software Architecture. Fundamental concepts or properties of a system (or software) in its environment embodied in its elements, relationships, and in the principles of its design and evolution [3].

SLR
Systematic Literature Review

SOA
Service-Oriented Architecture. in computer software design is an architectural style where in services are provided to the other components by application components, through a communication protocol over a network1.

Tekes
The Finnish Funding Agency for Technology and Innovation

---

Orchestration

In SOA-based applications, orchestration captures the flow of service interactions from the viewpoint of an individual service provider or participant.

TAT  Thematic Analysis based on Templates (also known as Template Analysis)

UML  Unified Modelling Language
List of original publications

The publications on which this dissertation is based on are as follows.


## Contents

Abstract  
Tiivistelmä  
Preface  
Abbreviations and terminology  
List of original publications  
Contents  

1 Introduction  
   1.1 Motivation  
   1.2 Related Work  
   1.3 Problem Statement  
   1.4 Objectives and Research Questions  
   1.5 Research Scope  
   1.6 Study Outline  
   1.7 Dissertation Structure  

2 Research Approach  
   2.1 Overview of Hevner et al.’s Research Framework  
   2.2 Research Design and Implementation  

3 Publication Overview  
   3.1 Overview of PI: Customized Choreography and Requirement Template Models as a Means for Addressing Software Architects’ Challenges  
   3.2 Overview of PII: Choreography in Embedded Systems Development Domain: A Systematic Literature Review  
   3.3 Overview of PIII: Comparative Influence Evaluation of Middleware Features on Choreography DSL  
   3.4 Overview of PIV: Choreography Modelling in Embedded Systems Domain – Requirements and implementation technologies  
   3.5 Overview of PV: Choreography Modelling Language for the Embedded Systems Domain – Empirical Evaluation and Lessons Learned  

9  
11  
13  
15  
17  
18  
19  
24  
25  
27  
30  
32  
33  
34  
39  
43  
45  
47  
49  
51  
53
4 Research Contribution

4.1 Contribution from PI: Customized Choreography and Requirement Template Models as a Means for Addressing Software Architects’ Challenges ...................................................... 56

4.2 Contribution from PII: Choreography in Embedded Systems Development Domain: A Systematic Literature Review ...................... 57

4.3 Contribution from PIII: Comparative Influence Evaluation of Middleware Features on Choreography DSL .......................... 57

4.4 Contribution from PIV: Choreography Modelling in Embedded Systems Domain – Requirements and implementation technologies ...... 58

4.5 Contribution from PV: Choreography Modelling Language for the Embedded Systems Domain – Empirical Evaluation and Lessons Learned .................................................. 59

4.6 Answers to Research Questions ........................................... 59

5 Implementation of the CML and Editor ................................. 63

6 Discussion ........................................................................... 75

6.1 CML in its Scientific Context .............................................. 75

6.2 Validity of the Research Approach ...................................... 77

6.3 Trustworthiness of the Study ............................................. 80

7 Conclusion and Future Work .............................................. 85

References ........................................................................... 89

Appendices ........................................................................... 95

Original Publications ........................................................... 103
1 Introduction

The majority of devices used in everyday life are controlled by embedded systems (ES). These systems control smartphones, entertainment systems and house appliances as well as large machines, such as industrial robots, cars and aeroplanes. The growing demand for ES along with the availability of affordable hardware components continuously broadens their application areas and increases the size and complexity of the software [4], which leads to ES-specific software development challenges, such as real-time processing, platform heterogeneity, reliability and limited resources [5]. As a partial solution to these challenges, Ebert and Salecker [6] suggested enabling effective communication among stakeholders across domains, disciplines and cultures.

The establishment of effective communication among stakeholders in ES development is not an easy task. One of the main challenges is the lack of information in documents that are exchanged between stakeholder groups involved in software development. For example, communication between original equipment manufacturers and suppliers in the automotive domain is hindered by a lack of information in requirement specification [7] or a lack of technical information in the documents they use [8]. Difficulties in communication among ES practitioners were also identified by Chen and Babar [9]. One of the findings from their study revealed that documented software variability is difficult to communicate to different stakeholders due to modelling approaches that are not user-friendly. Iwai, Oohashi and Kelly [10] identified ambiguous concepts in models that were used for service modelling in vehicle-to-vehicle and vehicle-to-infrastructure systems as the source of development challenges. The findings from their study suggested that improvements were needed in the communication among companies involved in the development of these systems.

Communication challenges among stakeholders involved in ES development processes that are caused by a lack of information in a documented form was the problem addressed in the study presented in this dissertation. This study focused on software development in the ES domain and on the utilisation of software architecture (SA) [11], service-oriented architecture (SOA) [12] and SOA-specific modelling techniques for software development. SA encompasses the documentation regarding relevant viewpoints or aspects of a software system [11]. Choreography modelling, which is one of the SOA-specific modelling techniques [13, 14], was viewed as a solution approach
to provide the missing information from the SA and consequently a way to address the communication challenges.

The study presented in this dissertation was conducted as part of an AMALTHEA international research project [15]. AMALTHEA was an ITEA2 [16] project, the goal of which was to identify the relevant aspects of ES development and to support them with modelling languages and tools. Therefore, this project provided a suitable environment to explore the problem addressed in this study. This study consisted of five individual studies that were coordinated within the AMALTHEA project using the design science research framework [17]. The results of the five individual studies advanced this study toward its goal, and these results are published and presented in scientific conferences and journals. These publications are included in this dissertation.

1.1 Motivation

The development of SA is known to facilitate communication between software developers as well as among other stakeholders involved in a software development process [11, 18–21]. SOA is a concept that emerged at the beginning of the 1990s [22] and became one of the dominant approaches for the development of software and their SAs in the enterprise systems development domain [23, 24]. Several studies conducted within the enterprise systems domain indicated that the utilisation of SOA, its principles and techniques can influence communication in several ways.

Haines and Haseman [25] studied SOA adoption patterns in eight organizations with on-going SOA initiatives. Their findings revealed that SOA improves the relationship among different business units by facilitating a new way of thinking among developers and by serving as an effective communication tool.

Similar observations were made by Becker et al. [26], who investigated the perspectives of users and vendors regarding the potential value and challenges of SOA. One of the findings highlighted the possibility to increase the alignment between development and business organisations as a result of the more easily understood architecture and the established common language, which contributed to better communication.

Collaboration within and across organisational barriers based on SOA technologies was explored by Schulte et al. [27]. This study revealed that the utilisation of SOA-based technologies leads to new opportunities for collaboration, cooperation and consequently communication among inter and intra-organisations, resulting in cost reduction, increased flexibility and reduced time to market.
Two studies identified industry-accepted and standardised information exchange protocols, data formats and interfaces as the basis for enhanced communication. The first study was performed by Krishanan et al. [28], who investigated information transparency and its influence on supply chain performance. The second study was conducted by Huang et al. [29], whose focus was the investigation of the strategic benefits of the implementation of web services as the dominant implementation technology for SOA.

Literature sources also indicate that choreography modelling, which is one of the integral techniques in SOA, has the potential to facilitate communication among stakeholders in the enterprise systems development process [30]. This potential was recognised, for example, by Mahfouz et al. [31], who designed the framework for customising choreography specifications, which enabled stakeholders to collaborate and to reconcile their individual business needs. Similarly, Koehler et al. [32] indicated that the two-way (or round-trip) translation of high-level business process models to implementation-level choreography models contributes to more effective communication regarding changes among business analysts and developers. The importance of communication in choreography modelling can also be argued with explicit requirements for communication support that are imposed on choreography modelling languages, as presented in [33, 34].

The presented study results indicate that the utilisation of SOA principles and choreography modelling for SA development facilitates communication among stakeholders in the enterprise systems domain. This knowledge motivated the consideration of choreography modelling, which is an SOA-specific technique, as a suitable solution to improve the content of SA documentation and consequently to facilitate communication among stakeholders in ES development domain.

1.2 Related Work

To capture and to present the choreography viewpoint in the ES domain, a choreography modelling language was designed during the study presented in this dissertation. This language design was influenced by existing modelling languages that are recognised in scientific literature and used for choreography modelling in the ES domain.

This section introduces the existing languages used for choreography modelling in ES by providing an overview of each language and by describing how they are used in the reported studies. Existing languages and a summary of each are presented in Table 1. The descriptions of how these languages were used in the reported studies
highlights the development areas in which they were used, the development aspect they aimed to support and the purpose for which they were used. The needs of the identified development areas, the development aspects and the purpose of using the modelling language were considered during the design of the choreography modelling language in this study and influenced the definitions of its language constructs.
Table 1. Choreography modelling languages used in ES domain.

<table>
<thead>
<tr>
<th>Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Process Model and Notation (BPMN) [35]</td>
<td>The BPMN language is one of the most commonly used languages in enterprise systems development for business process modelling and execution.</td>
</tr>
<tr>
<td>Web Service-Choreography Description Language (WS-CDL) [36]</td>
<td>The WS-CDL is a choreography modelling language specifically designed for enterprise services whose interfaces are implemented using WSDL interface. [37].</td>
</tr>
<tr>
<td>Business Process Execution Language (BPEL) [38]</td>
<td>Similar to BPMN and WS-CDL, BPEL is primarily used in the enterprise systems development domain for the modelling of internal processes.</td>
</tr>
<tr>
<td>Executable Choreography Language (ECL) [39]</td>
<td>ECL allows for the modelling of choreography scenarios that can dynamically change the default process flow based on various contextual factors.</td>
</tr>
<tr>
<td>Policy Enforced State Machine (PESM) [40]</td>
<td>The PESM language allows for the flexibility and the reuse of collaborations by defining policy-ruled choreography models of collaborating components.</td>
</tr>
<tr>
<td>Simple Application Logic description using Transducers (SALT) [41]</td>
<td>The SALT language enables the description of the distributed logic of Internet of Things applications in a choreography manner.</td>
</tr>
<tr>
<td>Scribble [42]</td>
<td>Scribble is a choreography language designed to capture a communication protocol specification among concurrent components embedded in devices used for ocean observation.</td>
</tr>
<tr>
<td>Abstract Process Engine Language (APEL) [43]</td>
<td>APEL is a hybrid modelling language that merges choreography with other viewpoints to efficiently capture service compositions.</td>
</tr>
<tr>
<td>Signal Calculus-Signal Core Language (SC-SCL) [44]</td>
<td>SC-SCL is a specialised formal language that focuses on long-running transactions in SOA-based environments.</td>
</tr>
<tr>
<td>CHOREO [45]</td>
<td>The CHOREO language captures choreography models in consideration of pervasive environment requirements.</td>
</tr>
</tbody>
</table>
BPMN, WS-CDL and BPEL are languages designed for the purpose of process modelling in enterprise systems development. The remaining languages presented in Table 1 were developed with ES development in mind; however, their utilisation in other domain is not excluded. These languages are similar because each of these languages was designed to capture the choreography viewpoint of service or component interactions; however, they can also be differentiated based on the development phase or the specific development aspect they aim to support. The following overview highlights these differences.

Along with the adoption of SOA in the ES development domain, various research initiatives were conducted in which the modelling languages from enterprise systems development were reused for ES development purposes. For example, Cortes-Cornax et al. [46] merged BPMN with an ontology of recommendations and evaluations by metrics. The ontology was attached to BPMN elements and was associated with a knowledge base from which the correct recommendations were extracted and presented to developers. The purpose of this endeavour was to improve the quality of the choreography models and to reduce the cognitive burden on the developers. Cambronero et al. [47] supplemented WS-CDL with real-time information, which allowed it to capture the system’s time-related execution constraints. Capturing this particular aspect of ES is realized through the extraction of real-time data from RT-UML diagrams and adding them to WS-CDL specifications during the automated transformation process. Similarly to BPMN and WS-CDL, Zhang et al. [48] relied on reusing the BPEL in the ES domain. Since the BPEL constructs were not designed to support the choreography (or peer-to-peer) modelling, the authors of the study supplemented the language with additional attributes. These attributes can capture the location of the participant in the choreography scenario and enable choreographed interactions as one of their services.

A number of research initiatives in the ES domain resulted in the development of a language for choreography modelling. The ECL language, presented by Cottenier [39] and the PESM language, presented by Rossebø and Braek [49], are choreography modelling languages designed to support the dynamic changes of the default system behaviours. Both the ECL and PESM choreography models allow for the specification of rules that are related to choreography language constructs. During system execution, the specified rules can detect changes in the environment and can react according to those changes. As a result, the default flow of execution dynamically adapts to the emerging changes in an environment.
The SALT language was developed for Internet-of-things applications [50]. The language is part of a large, integrated development environment that implements mechanisms to cope with hardware abstraction, service discovery, deployment and execution. Within this environment, the SALT language facilitates the development aspects typical in the Internet-of-things domain, such as work with timers, cooperation among things and algorithm updates. Thus, Internet-of-things applications an benefit from improvements in flexibility and versatility, which were two of the main design goals of the SALT language.

Scribble is a choreography language designed to facilitate the design and testing of communication-based software in the distributed systems domain [51]. Scribble’s syntax was designed based on the multiparty session type theory and aims to support the rigorousness and verifiability of the communication protocol. Rigorous and verifiable specifications were considered crucial aspects and thus were the focus of the Scribble language.

APEL language specifies the orchestration viewpoint, but it also captures the distribution of system parts over the computer network as well as the dedicated devices where these system parts are executed [52]. Hence, orchestration specifications also convey choreography-relevant information. In APEL, this is achieved using the annotations of language constructs that indicate the specification parts that will execute on different devices. During execution, the distributed devices realize the choreographed interactions.

SC-SCL was designed based on two related formal calculi, which are the network coordination policy and the signal calculus [53]. One of the main reasons for establishing a formal foundation for the language was to allow for formal reasoning techniques, which were considered important for SOA-based applications. The focus of the SC-SCL language is long-running transactions that are expressed both with the choreography and the orchestration models.

The CHOREO language supports development aspects, such as a varying number of services and actors and fault tolerance [45]. These two aspects are typical for pervasive systems and thus were the focus of CHOREO. To address these aspects, the CHOREO language implements (a) a set-based invocation, which allows for the invocation of a large number of services offering the same operation, and (b) the monitoring of incoming and outgoing messages.

The presented studies revealed that in the ES domain, choreography specifications can be used in different development areas, such as system analysis, implementation and
testing, for purposes such as documentation, code generation, dynamic behaviours or run-time monitoring. These studies also suggested that stakeholders from different development areas can use the choreography specifications for their own purposes; however, the constructs of the presented choreography modelling languages are designed for the particular purpose or development aspect they aim to support. Consequently, the information conveyed in the specification is sufficient only for stakeholders engaged in a development area for which the language is designed. Once the choreography specification, as a part of the SA documentation, is utilised by stakeholders in other development areas or with different purposes in mind, the information content in this specification becomes insufficient.

1.3 Problem Statement

SA consists of documented viewpoints on a software system; these viewpoints are relevant to stakeholders interested in that software system [3, 11]. Stakeholders see documented viewpoints as a common knowledge base and as a tool to achieve a common conceptualisation about the software system [54]. Because sharing the same knowledge and having the same understanding about various concepts of the software system are pre-requisites for a successful communication process, SA is considered one of the key enablers of successful communication in software development [18–21]. Achieving successful communication, however, assumes that all relevant viewpoints are included in SA and that documented viewpoints encompass all the information relevant to stakeholders. This assumption is not always satisfied, which leads to ambiguity and misinterpretation of SA. Consequently, the lack of information content in SA can cause various communication problems among the stakeholders that use it.

Embedded systems development represents a domain in which the lack of relevant information in SA causes communication problems [7–10]. One of the reasons is that the development of SA in the ES domain is increasingly guided by the principles of SOA [55–57], whereas the majority of the tools that are used to support such principles are developed for the enterprise systems domain. For example, choreography is a SOA-specific viewpoint on the software system [14] and is a part of the overall SA. Experiences from enterprise systems development suggest that choreography facilitates communication among stakeholders [30–32], and for this reason, it can be considered to have the potential to improve SA content and thus communication in the ES domain. The scientific literature, however, indicates that the choreography specifications used in
the ES domain may lack the needed information content because of limitations in the existing choreography modelling languages that are used for their development (see Section 1.2).

Following the presented reasoning, the problem addressed in this study is derived. The communication problems among different stakeholders in ES development, which are caused by the lack of information content in SA is the problem addressed by the research conducted.

1.4 Objectives and Research Questions

The study presented in this dissertation relied on choreography modelling as a way to improve the content of SA documents and consequently facilitate communication among stakeholders in the ES development process; however, scientific literature has indicated that choreography modelling in the ES domain is inhibited by the limitations of the choreography modelling languages that are currently used.

During the study, the identified limitations in the existing choreography modelling languages were explored and addressed by designing the choreography modelling language. At the core of these limitations is insufficient information content; thus, the choreography modelling language design needs to be expressive enough to support the needs of stakeholders interested in developing or utilising choreography specifications during ES development.

Based on this need, the purpose and both short- and long-term objectives were derived. The purpose, or the immediate visible result, of this study was the design and implementation of the choreography modelling language that is applicable in the ES domain. The short-term objective was to advance the design of choreography modelling languages by identifying information content that enables the utilisation of languages throughout the ES development phases. The long-term objective was to contribute choreography language design-related knowledge to the software engineering community. To achieve the presented purpose and objectives, three research questions were developed:

*RQ1: What are the requirements of a choreography modelling language in the embedded systems domain?*

Existing choreography modelling languages are partially applicable in the ES domain. The answer to RQ1 contributed to the study’s objective by providing foun-
dational knowledge about the ES development aspects that need to be supported by a choreography modelling language and thus be applicable in an ES domain. This foundational knowledge is presented in the form of requirements, and these requirements were derived through the investigation of the industry environment and the existing scientific literature that focuses on the utilisation of choreography in ES.

**RQ2: How can the identified requirements be applied in the design and implementation of the choreography modelling language?**

Choreography modelling in the ES domain is supported by designing a choreography modelling language whose language constructs are sufficiently expressive and understandable to stakeholder groups that use the choreography specifications throughout the ES development phases. RQ2 expands on the answer to RQ1 and indicates how the identified requirements are used and how they influence the choreography language design and the identification of technologies that are suitable for language implementation. The answer to RQ2 was derived from the language design process, during which different technologies, methods and techniques were considered and used.

**RQ3: To what extent is the designed choreography modelling language applicable in the embedded systems domain from the practitioners’ point of view?**

The extent to which the designed choreography modelling language supports choreography modelling in the ES domain is the focus of RQ3. RQ3 was answered by conducting a field evaluation, during which the industry experts evaluated the language design and its applicability in various ES development phases. Consequently, the answer to RQ3 represents a practitioner’s viewpoint on the support the designed language provides during the ES development process.

The answers to the stated RQs enable the fulfilment of this study’s purpose and objectives. The study’s purpose is met as a part of the evaluation of the study results, during which the choreography modelling language is designed and its prototype editor is implemented. The fulfilment of the short-term goal facilitates the work of choreography language developers by showing how new languages can be designed or how existing languages can be adopted so that they are applicable in the ES domain. The fulfilment of the long-term goal is expected to strengthen the foundation for future research in the area of choreography in ES.
In the following sections of the dissertation, the designed choreography modelling language will be referred to as CML while the implemented editor prototype will be referred to as "the editor".

1.5 Research Scope

A lack of information in SA documentation used during the ES development process impedes communication among the involved stakeholders, and it is the problem addressed in this research. This problem is broad, or high-level, which can be researched using various research approaches and from several points of view. Consequently, if the research of this problem is to be meaningful and manageable, the scope of the research must be defined. This section first introduces the technical concepts and assumptions made during this research and then defines the research scope. Technical concepts and assumptions position the research in a concrete software development context. The scope of the research was defined using defined concepts and assumptions.

The technical concepts and assumptions are represented by rectangles in Figure 1 ES development represents an overall domain of this research; thus, the remaining concepts are arranged within its boundaries. ES includes a combination of hardware and software designed to provide a dedicated functionality for larger systems, such as cars or industrial machines [58]. This dedicated functionality is typically realised using embedded software, the development of which was the focus of this research.

Two assumptions regarding embedded software development were made in this research. These assumptions and their rationales include:

- ES software developers develop and rely on SA for documenting and communicating information for various stakeholder groups. There are two reasons for this assumption. First, SA is known to facilitate the development of non-functional (or quality) requirements, such as scalability, reliability or safety [18, 19], which are important aspects in embedded software development [6]. Second, well-defined SA encompasses the documentation that is relevant for software development and can facilitate communication among stakeholders [11, 18–21], which was found to be challenging in ES development.

- Model-driven engineering (MDE) [59] was adopted in this research as an overall engineering approach for ES development. In short, this approach identifies the relevant viewpoints of the ES and uses models and their modelling languages to
express and document those viewpoints. As a result, the majority of ES development documentation, including SA documentation, consists of various models. The reason for this assumption is that MDE is increasingly present in embedded software development [5] and thus was viewed in this research as a dominant overall approach for software engineering in the ES domain. In addition, increased presence of MDE was also confirmed in the AMALTHEA research project because the majority of the participating companies adopted this particular approach [15, 60].

Two assumptions and models, which are central elements of SA documentation, are presented within embedded software boundaries on the left-hand side of Figure 1. The right-hand side of the Figure 1 represents the solution approach for supplementing the content of SA documentation. These concepts include the principles of SOA [12], that represent an overall solution approach, the composability which is one of the key SOA principles and the modelling of choreography, which is one of the viewpoints related to the composability principle.

As discussed in the previous section, the utilisation of SOA and its principles can facilitate communication among stakeholders; however, the exploration of the influence of each SOA principle on communication exceeds the scope of this research due to significant resources needed for such an endeavour. For this reason, as well as the reasons presented in Section 1.1, this research focused on the choreography...
modelling technique. Choreography is one of the viewpoints of software systems and represents service interactions from the global or neutral point of view [14]. Along with other viewpoints on service interaction, choreography modelling is related to the composable principle, which denotes the ability to compose applications, also known as services, into a higher level of functionality that cannot be provided by individual applications alone.

In this research, the lack of information in SA documentation was addressed by supplementing the choreography models with additional information content. Choreography can be specified using free-form textual descriptions or using formal graphical or textual modelling languages specialised for this task. In this research, the supplementation of SA with choreography was realised through the specification developed using a modelling language. In Figure 1 this is represented by a full-lined arrow. The reason that documenting choreography was realised using a choreography modelling language rather than some other form of representation is the assumption regarding the increasing presence of MDE in ES. In MDE, models and modelling languages represent the principal concepts. This reasoning is illustrated by the dashed arrow in Figure 1.

In summary, the technical concepts and assumptions that were included in this research and the corresponding concepts and assumptions that were excluded are as follows:

- SA documentation of the embedded software was included in this research, while the documentation related to other ES development aspects, such as the hardware development or documentation relevant to financial or marketing aspects of ES, were excluded.
- Modelling of the choreography viewpoint was included in this research, while the remaining SOA related principles and viewpoints, such as the service contract, service reusability or orchestration, were excluded from this research.
- The research assumption was that SA documentation encompasses the choreography specifications that are developed using the modelling languages. Other forms of choreography viewpoint representations, such as textual representation, were excluded from this research.

These concepts and assumptions are presented in the highlighted, rounded rectangle in Figure 1, and they define the boundaries of the scope of this research.

---

2More details regarding the technical concepts are presented in Appendix 1
The research findings contribute new knowledge for the design of choreography modelling languages. These findings were derived as part of the language design process, which was the primary method used for knowledge generation in this research. The knowledge focus was on the information content of the choreography modelling language and its influence on communication in ES development. The influence of other factors on communication, such as the use of non-native languages or development in distributed locations, and the influence on other ES development factors, such as conformability, security or reliability, were not considered in this research.

The design process relied on five qualitative studies, which provided input from the industry, scientific literature and language evaluation with experts. A majority of the subjects who participated in these studies were involved in implementation and development tasks; thus, the generated knowledge is limited to their perspectives on ES development. A non-technical personnel perspective was not considered and thus was not used during the language design.

1.6 Study Outline

The objective of the study was to design a CML whose language constructs are expressive enough to capture the choreography viewpoint in ES development. For this purpose, the design science framework [17] was adopted to coordinate the individual research activities to achieve the study’s objective. Following this framework, individual research activities were organized in three groups: data collection, the design process and the design evaluation. These groups are presented in Figure 2 as rounded rectangles and are interconnected with full-lined arrows, indicating the flow of the research execution. The knowledge contribution of each group is indicated by dashed arrows.

**Group I: Data collection.** The research activities in this group encompass data collection activities. The data was collected from ES development companies and from scientific literature that focuses on choreography in the ES domain, and it was analysed using qualitative research methods. The analysis results represent the foundational knowledge for understanding the problem addressed in this study and for the CML design.

**Group II: Design process.** The research activities in this group encompass the actual design of the CML and the design operationalisation through the implementation of the CML and its editor. In design science studies, the design process is the centre of the research and is viewed both as an ICT artefact development process and as a method
Fig. 2. Study outline.

for generating scientific knowledge [61]. For this reason, the design activities were also considered research activities; thus, the knowledge contribution is also an output from this group of research activities.

The design process group uses the data provided from the first group to design the CML, and it forwards the design to the following group of activities for evaluation. The evaluation results are returned to the design process, and they are used for the design improvements. The design process group of research activities can also induce additional data collection and evaluation activities in cases when it is needed. This indicates that the flow of research execution is not strictly sequential but rather iterative in nature.

**Group III: Design evaluation.** The research activities in this group encompass the evaluation of the CML design. The CML design evaluation was realised both in an academic and in an industry context. In an academic context, the derived design requirements were evaluated by language implementation and with the implementation of its editor prototype. The evaluation in the industry context was realised by collecting expert opinions about the CML’s impact on SA documentation. The evaluation results revealed the extent to which the CML fulfils the identified requirements and the experts’ perspectives of the CML’s impact. These results also indicated possibilities for CML improvements and thus could induce additional design activities.

Evaluation of the design is also enabled by opening the design and editor to wider audience. The language and the language editor prototype are publicly accessible and
can be obtained from [62]. Research and experts are encouraged to evaluate the design and to provide feedback.

The scientific contribution of this study is presented in five publications that are included in this dissertation. In summary, these findings include the categorised SA development challenges, the influence of different implementations of middleware features on a choreography modelling language, an overview of the choreography use in the ES domain, design requirements for the choreography modelling in ES and experts’ opinions on the CML’s impact on ES development. In Figure 2, the knowledge contribution is presented in highlighted, rounded rectangles and as an output from the three groups of research activities.

1.7 Dissertation Structure

A detailed description of this study is presented in seven chapters and one appendix. Chapter 1 introduces the primary foundations of this study, such as the problem that is addressed, the motivation, the study objectives and the research questions. The research approach adopted in the study is presented in chapter 2. This section first introduces the research methods that were used and then explains how they were adopted as part of the overall research design. The study was documented in five scientific publications, which are included in this dissertation. Chapter 3 introduces the scientific forums in which the publications were presented and outlines these publications. The ways that the studies and their results presented in each publication contributed to the answers to the stated RQs and to the overall goal of this study are presented chapter 4. Detailed presentation of the implemented CML and editor is presented in chapter 5. The discussion of the answers to the RQs and the validity threats are presented in chapter 6. The conclusions, which consist of the summarised description of the primary study contributions, and the possibilities for future research are presented in chapter 7.

Note on Appendix: In addition to the presented chapters, the dissertation includes one appendix. Appendix 1 introduces the foundational concepts of this study. These concepts include the AMATHEA research project and the case company, which provided more insights that were relevant for the study compared to other participating companies. In addition, this appendix facilitates the understanding of this study by presenting the technical concepts on which the study was based. These concepts include SOA, choreography, orchestration and MDE.
2 Research Approach

The paradigmatic approach underlying this study was design science research. Design science research emerged in the information systems field, but according to Kuechler and Vaishnavi [61], it is naturally applicable in various computer science-related fields, such as databases, human-computer interfaces and software engineering. Two main reasons motivated the adoption of design science research as a paradigmatic approach for this study. These reasons include:

– The design science research approach is oriented towards solving practical problems in environments and organisations [17, 61]. This study was driven by the communication problems identified in ES development companies, the root-cause of which is related to the lack of information content in SA documents. These communication problems represent the practical problem addressed in this study.
– In design science research, the identified practical problems are addressed with the design of innovative ICT artefacts [17, 61]. The problem identified in this study was addressed by the design of a choreography modelling language that is applicable in the ES domain.

Scientific literature recognises several methodological guidelines for conducting design science research. Examples of methodological guidance include the framework presented by Hevner et al. [17] and the regulative cycle presented by Wieringa [63]. The research implementation of this study was guided by the design science research framework by Hevner et al. [17]. This framework was developed to conduct and evaluate design science research in information systems. Consequently, this study represents the framework’s adoption in the software engineering field.

The study consisted of five individual studies. The chosen framework coordinated these studies and ensured that their results contributed to the overall study objective. The five studies were based on empirical data that was collected from AMALTHEA industry partners. The collected empirical data was analysed using qualitative analysis methods and produced results based on which CML was designed. A qualitative analysis aims to explore a phenomenon that is not well-understood by exploring and describing the motivation, problems or reasoning behind it [64]. The choreography modelling in ES is considered a phenomenon that is still not well-understood; thus, the qualitative approach
was used in the individual studies for the analyses. Consequently, this design science study can also be viewed as an exploratory empirical software engineering study.

This chapter introduces the overall research approach and consists of two sections. The first section introduces Hevner et al.’s design science framework and the research methods used within the framework. The second section describes the research design and implementation.

2.1 Overview of Hevner et al.’s Research Framework

The information systems discipline focuses on the design of the ICT artefact\(^3\) and on the management issues within the organisation it is deployed [65]. Investigation techniques in the information systems discipline therefore comprise methodological approaches for studying both the construction, or the design, of the ICT artefact and the people who are exposed to it. With this in mind, Hevner et al. [17] presented a framework that merges two paradigmatic approaches that are dominant in information systems studies: design science and behavioural science. The purpose of this framework is to help researchers and practitioners conduct, evaluate and present design science research studies. Since the design science research approach can be used in fields such as software engineering, Hevner et al.’s framework was adopted in this study as methodological guidance.

In the following two sections, the framework is presented by introducing its basic elements and by presenting its underlying logic using regulatory circles. The final section summarises the data collection and analysis methods used during the individual studies that were conducted within the framework.

**Basic elements of the Framework**

The basic elements of the Hevner et al.’s framework include the environment, knowledge base and the design research. These elements and the ways they are related are presented in Figure 3.

---

\(^3\)Artifact related to information communication technology (ICT)
In the centre of the framework is the ICT artefact and its design. The ICT artefact provides solutions to the practical problems identified in the environment or improvements to existing solutions, and it represents the results of the design research. The design of the artefact involves a process of artefact creation; however, within design science research, it is also a method used for generating new knowledge [61, 66].

According to Hevner et al. [17], a research process is conducted in two complementary phases. The first phase includes developing and justifying the theory that explains the business need under investigation. The second phase includes building and evaluating the artefact that addresses that need. Through these phases, two paradigmatic approaches are merged in Hevner et al.’s framework. First, the behavioural science paradigm explains the business need with a theory. Second, design science addresses that need with the designed artefact.

Business needs originate from the environment, which is comprised of people, organisations and technology. These needs can be understood as the people’s perception of different goals, problems and tasks that are related to their organisations and the technology they are using. For researchers, business needs represent a source of practical problems that can be addressed through research. Guiding the research activities to address the practical problems that originate from business needs assures the relevance of the research.

Knowledge that is used during the research process originates from the knowledge base, which is comprised of the foundations and methodologies. Foundations comprise theories, frameworks, instruments, constructs, models, methods and instantiations,
which are used during the develop/build phase of the study. Methodologies comprise data collection and data analysis techniques, measures, validation criteria and formalism as well as provide steps for the justify/evaluate phase of the study. Applying foundations and methodologies correctly provide the needed rigor for the research process.

Hevner et al.’s framework recognises two types of research contributions. First, it considers the application of the artefact in the environment. This contribution represents the utility that is gained from the instantiation of the designed artefact in an environment. Second, it considers the artefact design. This contribution represents the new knowledge regarding how to design the ICT artefacts, and this knowledge is added to the knowledge base.

Regulatory Cycles as the Underlying Logic of the Framework

The framework for design science research can be characterised by three related cycles of activities that explain the framework’s underlying logic [67]. These cycles include the relevance, rigor and the design cycle, and in Figure 4, they are outlined with the three basic elements of the framework. Each circle is: (a) presented with distinguishable upper and lower parts, which highlight the direction of the information flow and (b) positioned between the framework’s elements related to the specific information flow.

Fig. 4. Regulatory cycles (adopted from [67]).
The relevance cycle connects the environment to the design research. The upper part of the cycle, represented by the full-line arrow, denotes the input of information about the identified practical problem and the data related to that problem that are needed for the artefact design. The lower part of the cycle, represented by the dashed-line arrow, denotes the field studies in which the designed artefact is evaluated in an environment. The field study results determine whether the additional iteration is necessary or the design satisfies the business needs of the environment.

The rigor cycle connects the knowledge base to the design research. The upper part of the cycle denotes the input from the existing knowledge base about the identified problem and about the design. The lower part of the cycle denotes the new knowledge in the form of extensions to existing theories and methods that were produced during the artefact design.

The design cycle iterates between the construction and evaluation of the artefact. The upper part of the cycle denotes the construction of the artefact, while the lower denotes the evaluation. The construction of the artefact relies on the data and knowledge delivered through the upper parts of the relevance and rigor cycles. The evaluation of the design in this cycle was realised in an academic, or laboratory, context and provides additional knowledge about the design properties. Based on this knowledge, additional design cycle iterations could be induced.

It should be emphasised that the presented regulatory cycles are also related to one another but have different iteration dynamics. Compared with the relevance and rigor cycles, the design cycle is more dynamic, or faster. This is explained by the number of design alternatives that can be developed and evaluated based on the inputs from the relevance and rigor cycles. Based on these evaluations, only the most promising alternatives were evaluated in the field, which were released along the relevance cycle, and had the potential to contribute to the knowledge base, which were released along the rigor cycle.

Research Methods

The five studies that were conducted and coordinated using Hevner et al.’s framework relied on empirical data collected in an industry context and on qualitative research methods to produce the results. These studies and their results were published in various scientific forums, and these publications are included in this dissertation. Table 2 summarises the data collection and the analysis methods, the publications that presented
the study in which the methods were used and the positions of the study results in Hevner et al.’s framework, which was adopted for the overall research study.

Table 2. Research methods used during the study.

<table>
<thead>
<tr>
<th>Data collection method</th>
<th>Data analysis method</th>
<th>Publication</th>
<th>Position in the framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interview [68]</td>
<td>Thematic analysis [69] and qualitative coding [64, 68]</td>
<td>P I</td>
<td>Input of business needs from the environment into the design research.</td>
</tr>
<tr>
<td>Systematic literature review [70]</td>
<td>Thematic synthesis [71] and qualitative coding [64, 68]</td>
<td>P II</td>
<td>Input of the applicable knowledge from the knowledge base into the design research.</td>
</tr>
<tr>
<td>Company specific documents and standards</td>
<td>DESMET [72] and the Goal Question Metric [73]</td>
<td>P III</td>
<td>Input of business needs from the environment into the design research.</td>
</tr>
<tr>
<td>Data from previous studies (PI – PIII)</td>
<td>Thematic analysis [69] and qualitative coding [64, 68]</td>
<td>P IV</td>
<td>CML design and the evaluation by implementation in an academic setting.</td>
</tr>
<tr>
<td>Focus group [74]</td>
<td>Thematic analysis with templates [75] and qualitative coding [64, 68]</td>
<td>P V</td>
<td>CML evaluation in the environment.</td>
</tr>
</tbody>
</table>

In addition to the data that was collected using the methods presented in Table 2, this study also relied on data sources such as company specific documents, which included the development process descriptions, various templates, software architecture specifications and standards and technical reports from the AMALTHEA research project. Similarly, along with the presented analysis techniques, some of the decisions that were made in this study were based on parts of the discussions with industry experts during numerous workshops. Finally, the study presented in PI was conducted as a case study [76]. In PI, a case study was seen as an overall research approach that guided the data collection and analysis activities. For this reason, the case study approach was not presented in Table 2.

4The term "qualitative" is used to differentiate the coding, as an analysis technique, from the coding in as programming source code.
2.2 Research Design and Implementation

The implementation of the overall research is explained by introducing individual studies within Hevner et al.’s framework and by describing their roles in the context of the framework’s logic. In addition, this section briefly presents the adoption of data collection and analysis methods used during the individual research activities and the rationales behind the selections.

The implementation of research activities within Hevner et al.’s framework is outlined in Figure 5. The framework’s primary elements, which are the environment, knowledge base and the design, are represented by pale grey lines and used as guiding templates or as blueprints for arranging and presenting the individual research activities. The research activities are represented by rectangles either within the environment or the body of knowledge depending on the focus of a particular activity. The results from these activities are also represented by rectangles and are positioned in the centre of the figure. The arrows connecting the research activities with the results describe which research methods were adopted to provide a particular result. Two types of arrows were used for this purpose. The first are the full-lined arrows, which denote the adoption of a concrete research method, such as an interview or a thematic analysis. The second are the dotted-lined arrows that denote a less formal inquiry, such as a non-systematic literature analysis or workshop discussions with industry experts.

The research implementation was realised in two phases: the preliminary investigation and the main research. The preliminary investigation denotes the activities that were conducted to identify the practical problem of interest, the study objective and the RQs. The AMALTHEA research project provided the high-level topics relevant from an industry point of view, while the preliminary discussions with industry experts indicated the concrete practical problems in ES development within those topics. The concrete practical problems were discussed from a scientific and literature point of view during the numerous discussions with senior researchers and resulted in the initial definitions of the problem, objective and research questions. These activities and their results are presented above the dashed line in Figure 5, and they are described in the introduction of this dissertation.
The main research is based on the preliminary investigation. In Figure 5 this phase is presented below the dashed line. The main research began with the investigation of the state of the practice that focused on SA development challenges in companies.
and potential approaches to address these challenges. This study is presented in PI, and interviews with experts were used as a primary data collection method; however, company specific documents, such as the specifications, process definitions and various guidelines, were used to supplement the interview data. A thematic analysis [69] was used for the interview analysis and resulted in empirical insights into SA design challenges and related choreography modelling with a sub-set of those challenges. These findings confirmed the prevalence of the communication among stakeholders, as a problem in practice that is identified in this research as well as the solution chosen to address this problem.

The research continued with the exploration of the body of knowledge by means of a systematic-literature review [77] and the thematic synthesis [71] methods. The purpose of this step was to collect evidence regarding the utilisation of choreography in ES as well as to learn about the existing choreography modelling languages, technologies and benefits from their use. A systematic literature review method is a well-accepted method used for collecting relevant scientific literature in the software engineering field; thus, it was chosen for this study. The thematic synthesis method was chosen because its aim is to synthesise knowledge from existing scientific sources. This study is presented in PII.

The design of the ICT artefact was the focus of the framework that guided this research. The artefact was the CML, the design of which is represented by the shaded rectangle in Figure 5. According to [17], the ICT artefact design is a cyclic activity that iterates between development and evaluation activities until a satisfactory design is achieved.

The development activities were based on the knowledge provided by the exploration of the practice and the scientific literature (PI and PII) as well as on two additional studies.

– The first study is presented in PIII and focused on the comparison of two middleware products and the analysis of their influence on choreography language design. This study was motivated by the analysis of scientific literature, which revealed that choreography is frequently studied in the context of middleware products. Since the study approach was comparative, the DESMET method [72] was adopted, and the GQM [73] method was used for the derivation of metrics that were used by industry experts for comparison. As the main source of data, company specific documents and standards were used to compare and to identify which middleware features need to be supported by choreography modelling in the ES domain. This study also provided
insights into how to identify and support the information content with the modelling language.

- The second study is presented in PIV and its focus was the identification of technologies for language implementation. This study relied on data collected during previous studies and on insights from cooperation with industry experts. This resulted in the selection of the language that would be redesigned and the implementation technologies that would be used during the language implementation. The thematic analysis approach was adopted to identify the requirements for the CML design.

The evaluation activities were conducted both within academic settings and with industry experts. The evaluation in the academic setting was conducted by implementing the CML into an existing modelling language and by the development of its editor using the proposed implementation technologies. This study is presented in PIV. The evaluation with industry experts is presented in PV. The evaluation study was performed by collecting the industry experts’ opinions on the effects that can be expected from the utilisation of the designed modelling language. These opinions were collected using the focus group approach [78] and analysed using a thematic analysis with templates [75]. These methods of data collection and analysis were chosen based on the study by Tremblay [79], which recommended these methods as suitable for the evaluation of ICT artefacts in design science studies.

Design science studies contribute to both environments and to the body of knowledge [17]. In this study, the contribution to the environment, or to practice, is the improvement of communication caused by the increased informative nature of SA documentation. The contribution to the knowledge base consists of several intermediate findings that are published in scientific literature. These findings are presented in publications included in this dissertation (PI–PV) and include the categorised SA challenges, the influence of middleware features on choreography language, language implementation technologies and the evaluation of the designed CML.
3 Publication Overview

This dissertation includes five publications that report the individual studies conducted as part of the overall research and published in international peer-reviewed scientific forums that focus on software engineering topics. Four publications are published in conferences, and these conferences are: The International Conference on Software Engineering Advances (ICSEA), International Conference on Model-Driven Engineering and Software Development (MODELSWARD) and International Conference on Product-Focused Software Process Improvement (PROFES). One publication is published in a scientific journal, which is the Journal of Information and Software Technology (JIST). All publication forums are ranked in julkaisufoorumi\textsuperscript{5}, which is a classification of publication channels created to support the quality assessment of academic research.

Table 3 presents the authors of the publications, the year and the venue in which the publications are published. This table also summarises the study’s topic and the responsibilities of the author of this dissertation in these studies.

\textsuperscript{5}Julkaisufoorumi – Classification of the scholarly publication channels developed and maintained by The Federation of Finnish Learned Societies. Website: http://www.julkaisufoorumi.fi/en

43
<table>
<thead>
<tr>
<th>ID</th>
<th>Publication, topic and the responsibilities of the author of this dissertation</th>
</tr>
</thead>
</table>
Topic: Identification of SA development challenges and two models as a means for addressing these challenges based on interviews with industry experts. The study findings confirmed the prevalence of communication problems in ES development and indicated how to overcome them. Author’s responsibilities: Preparation of the questionnaire, conducting the interviews, analysis and the derivation of findings. |
Topic: Systematic analysis of the scientific literature that focuses on choreography in ES. The study finding provided a scientific context in the form of existing languages, technologies and industries in which the choreography was utilised in the ES domain. Author’s responsibilities: Design and implementation of search protocol, selection of the analysis method, publication analysis and the derivation of results. |
Topic: Comparison of middleware products and the demonstration of how to capture middleware specifics with choreography language. Based on the knowledge from this study, various other choreography-relevant aspects were supported by the CML design. Author’s responsibilities: Comparison of middleware products, derivation of the comparison metrics, middleware product comparison and the analysis of their influence on the choreography modelling language. |
Topic: Derivation of design requirements and implementation technologies for the CML design based on the data collected throughout the dissertation research. The feasibility of the requirements and technologies was evaluated by implementing the CML and its editor. Author’s responsibilities: Collection of data that were relevant to the CML design, the analysis and derivation of findings. |
Topic: Evaluation of the designed CML by industry experts. The study findings indicated the impact of the CML on ES development. Author’s responsibilities: Selection of the focus group as a method for the CML evaluation, the implementation of two focus groups with industry experts and the analysis of the collected data. |
In addition to the responsibilities of conducting the studies that are presented in Table 3, the author of this dissertation also played major role in writing each publication. These responsibilities included structuring, writing and formatting the publications and the preparation of figures and tables. The author of this dissertation is the first author of each publication.

The overview of the publications consists of four sections. These sections introduce (a) the goal and research questions that guided the study presented in the publications, (b) research methods and how they were adopted, (c) the research results and (d) detailed description of responsibilities of this dissertation author in the studies.

3.1 Overview of PI: Customized Choreography and Requirement Template Models as a Means for Addressing Software Architects’ Challenges

Goal and research questions. The study presented in this publication explored the SA development practices of software architects from ES development companies. The goal of this research was to identify the challenges related to SA development and to propose solutions to address the challenges. The two RQs were:

– RQ1: What challenges do software architects face during the development and maintenance of software architecture design?
– RQ2: How to address the identified challenges with domain-specific models?

Research methods. Interviews with experts and company specific documents were used as data sources, while the thematic analysis method [64] was used to analyse the collected data. As supplementation for these methods, a number of workshops with industry experts were held to identify the challenging areas that needed to be addressed by the domain-specific models.

The primary source of data in this research was the interviews, which were conducted with five software architects from a large embedded systems development company during the first quarter of 2012. These software architects had between 10 and 26 years of experience in software development, and the interviews lasted between 1 and 2 hours. As a secondary data source, company specific documents were analysed alongside the interviews. These documents included various templates, process and work descriptions, example requirements and test specifications. The collected data was analysed using the thematic analysis approach, and coding was used as the core analysis technique
in the thematic synthesis. The thematic analysis and coding technique resulted in the identification of various patterns in data sources based on which the SA challenges were derived.

**Research results.** This study resulted in (a) the identification of the SA development challenges organised into four categories and their interpretations and (b) two domain-specific models, which were derived based on the identified challenges and aimed to address two subsets of those challenges. The four categories identified are:

- **Knowledge** - includes five challenges whose causes are related to the lack of both theoretical and practical knowledge regarding how to design SA.
- **Global software development** - includes four challenges that are related to geographically distributed development, multicultural development teams, different time zones and communication using non-native languages.
- **System size and complexity** - includes seven challenges that were related to the variety of implementation technologies, software platforms, development teams and the complexity of features that the system supports.
- **Architectural viewpoints** - includes six challenges that were related to the existence of different architectural views and viewpoints. For example, there are features, component, maintenance or performance viewpoints.

Two domain-specific models were developed as a means to address two subsets of the identified challenges. These models were derived through the workshops that were held with industry experts, during which the identified challenges were analysed and discussed. The developed models are:

- **The choreography based agreements model** aims to increase the developers’ awareness regarding which development team is responsible for which task and in which order those tasks need to be completed. A lack of awareness regarding these issues leads to work duplication, frequent delays and the loss of opportunities from the parallelisation of work. This model proposes choreography modelling using the customised modelling language.
- **The dynamic requirement template** aims to facilitate the collection and the communication of the needed technical information between the customers and software developers. A lack of technical information in the requirements leads to inefficient use of the software architects’ time due to additional communication with customers. This model proposes dividing the requirement template into a common part, which contains
data that is common to each requirement, and a specific part, which contains data that is specific to each part of the system that is maintained by a specific development team.

**Responsibilities of the dissertation author.** The dissertation author was responsible for the following tasks: preparation of the questionnaire, conduct of the interviews, and analysis and derivation of the findings. The questionnaire was prepared on the basis of scientific literature and in cooperation with the industry partner, who was also the third author of this publication. The dissertation author conducted interviews through telephone calls and face-to-face sessions. The analysis of the collected data and the derivation of the results were conducted by two researchers, one of whom was the author. The second researcher who contributed to the analysis and derivation of results also contributed to the publication writing and was the second author of the publication. The third author was the industry partner whose responsibilities were to review the paper from the industry’s point of view, ensure that company data were accurately interpreted and that sensitive company data were not revealed. The remaining authors facilitated the selection of the research methods, supervised all of the presented tasks and reviewed the text of the manuscript.

### 3.2 Overview of PII: Choreography in Embedded Systems Development Domain: A Systematic Literature Review

**Goal and research questions.** This study continued the research presented in PI by exploring the choreography modelling practices in ES development in more detail. The goal of this research was to provide a broad characterisation of the utilisation of choreography in the ES development domain based on what is known in scientific literature. For this purpose, the following RQs were answered:

- **RQ1:** What are the current developments in scientific literature that focus on choreography in ES?
- **RQ2:** How is choreography adapted in ES domain?
- **RQ3:** How is a choreography specification used in ES domain?
- **RQ4:** How is a choreography specification realized during the ES execution?
- **RQ5:** What implicit assumptions regarding choreography should be considered when utilizing choreography in ES domain?
Research methods. The answers to the stated RQs were based on the existing scientific literature, and for this purpose, the systematic literature review method [77] was adopted. This method is an accepted method in the software engineering field for synthesising evidence-based knowledge based on the published literature and was therefore selected for this research. The implementation of this method consisted of two major stages. During the first stage, the search protocol was planned and executed, while during the second stage, the relevant studies were analysed using the thematic synthesis approach and the custom-developed analytical tool.

The search protocol was implemented by searching eight relevant databases, which resulted in the initial list of 6823 relevant scientific publications. After the systematic screening of these publications, 48 were selected as suitable for the research purposes. The selected studies were analysed using the thematic synthesis method [71] and using the custom-made analytical framework [80]. The thematic synthesis method was selected because it provides guidance regarding how to produce new knowledge in systematic literature review studies, while the custom-made analytical framework facilitated the analysis of choreography adaption in the ES domain.

Research results. This research resulted in the derivation of five themes that provided answers to the stated RQs. The derived themes are as follows:

- The first theme presents the current developments in studies that focused on choreography in ES. These developments were presented as classifications of aspects, such as the study context, motivation and impact.
- The second theme presents the choreography adaption strategies. The analysis of the adaption strategies included the organisation of the studies according to the custom-developed analytical framework and the interpretation of the studies encompassed within the framework.
- The third theme explains how a choreography specification is used during ES development. Three types of choreography specifications use were identified: descriptive, parsed and executable choreography specifications.
- The fourth theme describes how a choreography specification is realised during ES execution. Three types of realisations were identified: participant driven, deployed and piggybacked choreography realisation types.
- The fifth theme presents two implicit assumptions for the utilisation of choreography in the ES development domain. These assumptions are participant autonomy and network stability. Participant autonomy reveals differences in how the participant
concept is interpreted in the ES and non-ES domains, while the network stability reveals the volatile nature of the network formed by the participants in a choreography scenario.

Based on the presented results, several study findings were derived. These findings were categorised as research-oriented and practice-oriented findings. The research-oriented findings included choreography as a pervasive specification, a lack of expressiveness in current choreography modelling languages used in ES, insufficiently explored adaption strategies and the lack of consideration for synthesis in current studies that focused on choreography in ES. The practice-relevant findings included the specific interpretation of participants in the ES domain, performance improvements and technological heterogeneity as an aspect worth considering when using choreography in the ES domain.

**Responsibilities of the dissertation author.** The dissertation author was responsible for the design and implementation of the search protocol, analysis of the identified publications and derivation of the results. During the design of the search protocol, the author defined the RQs, selected the publication sources, keywords and criteria for publication inclusion and exclusion and defined the quality assessment procedures. All authors of this publication were involved in the piloting of the criteria for publication inclusion and exclusion and in the implementation of the search protocol. The author led the protocol implementation, and, together with the other authors, he evaluated the collected studies. The methods used during the analysis were proposed by the author, and on the basis of the adoption of these methods, the study results were derived. The remaining authors of this publication supervised the analysis, contributed to the derivation of the results and reviewed the publication text.

### 3.3 Overview of PIII: Comparative Influence Evaluation of Middleware Features on Choreography DSL

**Goal and research questions.** The research reported in this study focused on the comparison of middleware products that are used in ES development and explored the influence of implementation differences on choreography modelling languages. The goal defined for this research was to learn about the middleware features that are relevant to service interactions and their influence on the development of choreography DSL. This goal was achieved by answering the following research question:
RQ: How do variations in the implementation of middleware features influence the implementation of the DSL for choreography modelling?

Research method. The RQ was answered by adopting two research methods. The first research method was DESMET [72] which proposes guidance regarding how to conduct a structural comparison in the software engineering field. In this research, the features of the two middleware products were the focus of the comparison. The second research method was the Goal Question Metric (GQM) [73], and this method was used to identify the middleware features and measurement scales that indicate the level of support these features provide to developers. The feature evaluation, the analysis of the evaluation results and the influence of the implementation differences from choreography DSL were conducted through numerous discussions with industry experts during workshops.

The middleware products that were compared were LISA and AUOTSAR. LISA is a proprietary middleware product used in the telecom domain, while AUTOSAR is an open middleware product used in the automotive domain. The primary data source used during this research were the company specific documentation for LISA and the publicly available specifications for AUTOSAR. Following the DESMET and the GQM, the researchers familiarised themselves with the two middleware products, identified the eight relevant features and derived the evaluation scales. These scales were used during the workshops to evaluate the identified features of the middleware products and to identify how the differences in the feature implementation influenced the choreography DSL.

Research results. The results consisted of two parts. The first part was the interpretation of the evaluation of middleware features using the derived measurements, while the second part described how the variations in feature implementation influence the development of choreography DSL. The middleware features evaluated during this research include: functionality access, location transparency, state information, message format, message transformation, interaction scenario, protocol support and protocol translation. The evaluation of these features revealed that the AUTOSAR features are more mature in comparison with LISA’s features, meaning that they offer more support to developers. The analysis of the influence on choreography DSL identified six language constructs that are affected by the variation in feature implementation. These language constructs are the participant, role, interaction, message content type, state variable and channel variable. To address the differences in the implementation of
middleware features, the language constructs were customised by adding additional classes and attributes or by modifying their relations.

**Responsibilities of the dissertation author.** The dissertation author was responsible for the derivation of the middleware comparison metrics, the comparison of the features of the two middleware products and the analysis of the influence these features have on the choreography modelling language. The author derived the metrics and then analysed and presented the influence of the features on the choreography modelling language. The comparison of middleware products was conducted in cooperation with the industry partner, who was also the second author of this publication. The contributions of the remaining authors included the preparation of the proposals regarding the selection of the research methods, supervision of the research process and revision of the publication text.

3.4 Overview of PIV: Choreography Modelling in Embedded Systems Domain – Requirements and implementation technologies

**Goal and research questions.** The study presented in this publication indicated the foundations on which the language for the choreography modelling in ES is designed. This foundation consists of design requirements for the choreography modelling language, technologies for language implementation and the feasibility evaluation of these proposals. The goal of the research was to produce knowledge regarding the design of a comprehensive choreography modelling language capable of supporting the choreography-relevant development aspects that are typical in ES development. The research question derived based on this goal was:

– **RQ: How to increase the expressiveness of a choreography modelling language to enable its applicability in the embedded systems domain?**

**Research method.** The research methods used in this research include the qualitative coding technique [64] and the practice of collaborative decision making. The qualitative coding technique was selected for this analysis because it enables the organisation of the data sources in a way that is most suitable for researchers to derive the results. The practice of collaborative decision making involves joint discussions, during which industry experts and researchers discuss the various questions that are relevant to
a certain task. These discussions are based on participants’ personal experiences and knowledge.

The primary sources of data for this research were the aggregated data sources and results from the previous research that were presented in PI, PII and PIII. These data included interviews, company specific documents, open specifications and the existing scientific literature that focuses on choreography modelling in the ES domain. These data sources were analysed using the qualitative coding technique, which allowed the researchers to efficiently organise the data and to derive the design requirements. In addition to coding, collaborative decision making was applied to select the technologies for the implementation of the choreography modelling language and to select the existing language on which the design requirements would be implemented.

Research results. This research resulted in the derivation of six design requirements for the choreography modelling language and in proposals regarding which implementation technology could be used for the implementation of the language. The derived design requirements include: constraint based access, ad-hoc networks, technical and technological heterogeneity, service invocation variants, real-time execution and supplementary information of language constructs. Each of the derived requirements was presented by describing the context in which the requirement is identified, the problem in ES development that it addresses and the knowledge source from which the requirement was derived.

The implementation technologies that were proposed for the implementation of the choreography modelling language include the business process model and notation (BPMN), eclipse modelling framework (EMF) and Sirius. BPMN is a standardised language for modelling business processes, and in this research, it was selected to be the base language on which the derived design requirements were implemented. EMF is a modelling and code generation facility for building tools on structured data models. Sirius is an EMF-based technology that facilitates the development of tools for the representation of the structured data model.

Responsibilities of the dissertation author. The dissertation author was responsible for the collection of the knowledge sources, the analysis, the derivation of the findings and the conduct of the feasibility evaluation. In this study, the author used knowledge sources that consisted of data collected in our previous studies (PI–PIII) and various technical reports produced in the AMALTHEA research project. The author selected the analysis method, conducted the analysis and derived the results. To evaluate the feasibility of the results, the author used the findings to design the CML and implement...
the prototype language editor. The most significant contributions from the remaining authors of this publication were the suggestion to use the language design and prototype editor implementation as a feasibility evaluation and to structure the research method section of the publication. Aside from these, the remaining authors supervised the research process and revised the publication text.

3.5 Overview of PV: Choreography Modelling Language for the Embedded Systems Domain – Empirical Evaluation and Lessons Learned

**Goal and research questions.** The study presented in this publication focused on the evaluation of the designed CML. This evaluation was realised by eliciting industry experts’ opinions about the impact of the CML on the ES development process. Consequently, the stated goal of the presented study was to learn about the applicability of the designed CML in the ES domain using the experts’ evaluations. Based on this goal, the following RQ was answered:

– *To what extent is the designed CML applicable in the ES domain from experts’ point of view?*

**Research method.** The answer to the stated RQ was provided by adopting the focus group approach [78] to collect the data and by using the thematic analysis with templates [75] to analyse the data. Both the data collection and the analysis were supported by an additional technique. The affinity grouping technique was used to support the data collection during the focus group session, while coding was used as part of the thematic analysis with templates. A focus group was used because it is a suitable technique for collecting experts’ opinions about newly developed ICT artefacts. A thematic analysis with templates was selected because it is a recommended approach for the analysis of data that is collected during a focus group in the software engineering field.

Two focus groups were conducted, during which nine participants from three countries commented on the choreography modelling language and the prototype editor developed during this dissertation work. Both focus group sessions were conducted in the first half of 2015 and lasted 80 and 50 minutes. The affinity grouping technique was used as part of the focus group sessions to facilitate brainstorming about the potential problems in using and understanding the designed choreography language. The thematic analysis with templates began with a predefined set of codes that reflected
the phenomena of interests. During the coding process, the predefined set of codes evolved by adding new codes and omitting useless codes.

**Research results.** The research results consisted of three parts. These parts include: (a) the application areas that denote ES development areas in which the language can be used, (b) design requirements that indicate how to improve the existing version of the language and (c) practical challenges regarding the use of the language in practice that revealed which needs should be considered to facilitate the deployment of the language in practical use. Based on these results, the lessons learned were derived.

The first lesson learned was that choreography modelling can be used to capture the interactions between verification steps in the testing phase. Choreography models, as defined in the scientific community, capture the interaction of participants’ services from a global, or neutral, point of view. To the authors’ best knowledge, this is the first report on the use of choreography in this context. The second lesson learned confirmed the use of choreography modelling for protocol development. The third lesson learned included the four design requirements that were used for guidance in the development of choreography modelling languages. Finally, the fourth lesson learned highlighted a user’s level of expertise and the presence of proper guidance regarding how to use CML during the introduction of the language into working practice.

**Responsibilities of the dissertation author.** The dissertation author was responsible for planning and conducting the data collection activities, analysing the data and deriving the results. Data collection planning included identifying the research problem, selecting the participants and structuring the data collection sessions. The author conducted two data collection sessions with industry experts to evaluate the designed CML. The work on this task was supported by the industry partner, who was also the second author of this publication. After the data collection, the author recommended the analysis method, analysed the data and derived the results. These results were reviewed by the industry partner to ensure the accuracy of the results. The remaining authors of this publication supervised the research process and reviewed the publication text.
4  Research Contribution

The individual studies presented in the five scientific publications (PI–PV) contributed to the overall study through the results based on which the stated RQs were answered. This section presents details about how the individual studies contributed to the study’s goal and how their results formed the answers to the stated RQs. This presentation begins with the summary of the common contribution of all individual studies, continues with the detailed description of their individual contributions and ends with the answers to the stated RQs. Table 4 illustrates the relation of the RQs with the individual studies whose results contributed to their answers.

Table 4. Contribution of the publications to RQs.

<table>
<thead>
<tr>
<th>RQs</th>
<th>PI</th>
<th>PII</th>
<th>PIII</th>
<th>PIV</th>
<th>PV</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1: What are the requirements for choreography modelling language in embedded systems domain?</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RQ2: How can the identified requirements be applied in the design and implementation of the choreography modelling language?</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ3: To what extent is the designed choreography modelling language applicable in embedded systems domain from the practitioners’ point of view?</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

The study presented in this dissertation is a design science research study, the objective of which is to advance the design of a choreography modelling language by identifying information content that enables its use throughout ES development phases. Each of the individual studies (PI–PV) contributed to the identification of the relevant information content that needs to be supported by the choreography modelling language design. This content is embodied in ten design requirements that were derived based on knowledge from the (a) analysis of both practice and the scientific literature (PI–PIII) and the (b) evaluation in academic and industry settings (PVI and PV). In addition, knowledge gained from the design itself was used to derive the design requirements. During the CML design process, numerous design alternatives were checked against the acquired knowledge to identify a satisfactory design. The selected alternatives were
built into the CML design. This ensured that the design requirements could be supported by the CML implementation and the applicability of the implemented CML in practice.

The ten design requirements for choreography modelling languages that were evaluated by implementation and by industry experts are the main knowledge contributions provided by this study. This knowledge is recommended to be considered during the design of choreography modelling languages that will be used in the ES domain or as part of any initiative that aims to support choreography modelling in the ES domain.

4.1 Contribution from PI: Customized Choreography and Requirement Template Models as a Means for Addressing Software Architects’ Challenges

The contribution of the research presented in PI to the overall study is twofold. First, the analysis identified communication problems among software architects in ES domain and related these problems with different understanding and misinterpretation of information in SA documents. This result underpins the relevance of the communication problems in ES by indicating on their presence in today’s ES development companies and by confirming similar findings from the literature, such as [7–9]. Second, a subset of SA development challenges, including communication, was identified and related to the choreography modelling that is proposed as a potential solution to these challenges. This result supported the indications from the literature [31–34] that as an integral SOA practice, choreography modelling is a potential solution to address challenges related to communication.

The results from this research contribute to RQ1, which is: what are the requirements for choreography modelling in ES domain? This contribution includes a set of SA development challenges that: (a) influence stakeholder communication in the ES domain and (b) can be addressed by choreography modelling. The derived SA challenges were considered a starting point for deriving the requirements for the CML design, while the choreography modelling was confirmed to be suitable in addressing SA challenges that are related to communication.
4.2 Contribution from PII: Choreography in Embedded Systems Development Domain: A Systematic Literature Review

The contribution of the research presented in PII continued the work presented in PI and contributed to the overall study by providing existing scientific knowledge about choreography in the ES domain. This knowledge revealed various ways that choreography is used in ES development and indicated knowledge gaps that can be addressed. The systematic exploration of the existing knowledge also underpins the rigorousness of the study and positions the research in its scientific context.

The analysis results provided by this research, along with the results presented in PI, revealed various aspects and phases in ES development that can be supported by the information content of the choreography modelling language. In addition, the results identified the technologies that were most commonly used for choreography language design and implementation in the ES domain. The revealed aspects and phases contributed to RQ1 because they provide valuable areas that can be supported by the CML design and thus contribute to the derivation of the design requirements. Knowledge of implementation technologies contributed to RQ2, which is: how can the identified requirements be applied in the design and implementation of the choreography modelling language? The identified technologies were considered a starting point for selecting the CML design and an implementation approach that could satisfy the identified design requirements.

4.3 Contribution from PIII: Comparative Influence Evaluation of Middleware Features on Choreography DSL

The research presented in PIII contributed to the overall study by presenting how a middleware product, which is a concrete ES development aspect, influences a choreography modelling language design. The research showed how information content is extracted based on the analysis of two middleware products and how an existing choreography modelling language is supplemented with that content. Since middleware products were identified as an aspect relevant to choreography in previous studies (PI and PII), the added information content was viewed as an improvement of the language expressiveness and as a potential facilitator of its adoption by the ES development community.
The research results presented in this publication partially answered RQ1 and RQ2. A detailed comparison of middleware products revealed the features that are relevant to choreography and thus were considered requirements for the CML design and a contribution to RQ1. The contribution to RQ2 consists of the extraction of information content from the choreography-relevant middleware features and its incorporation into the syntax and semantics of an existing choreography modelling language. The contribution to RQ2 suggests that extracting information content from the choreography-relevant aspects and incorporating the content into an existing language is a suitable approach for supporting various ES development aspects and phases with the modelling language. This approach was used for the CML designed in this study.

4.4 Contribution from PIV: Choreography Modelling in Embedded Systems Domain – Requirements and implementation technologies

The contribution of the research presented in PIV continued the work presented in the previous publications, PI–PIII. This research focused on the initial evaluation of the CML design that is based on the (a) derived design requirements and (b) implementation technologies proposals. The initial evaluation was realised through the implementation of the CML and its editor following the approach presented in PIII. The evaluation suggested that the derived design requirements can be implemented following this approach and using suggested technologies.

This research contributed to RQ1 due to the analysis of the knowledge collected throughout the studies presented in PI–PIII, which resulted in the design requirements for the CML. Along with the research presented in PII and PIII, PIV also contributed to RQ2 by presenting (a) the information content that is incorporated into the CML design, (b) suggesting the concrete implementation technologies that are suitable for the CML design and editor implementation and (c) the evaluation of the derived requirements and technologies by means of language and editor implementation.
4.5 Contribution from PV: Choreography Modelling Language for the Embedded Systems Domain – Empirical Evaluation and Lessons Learned

The contribution of the research presented in PV is the evaluation of the CML based on the analysis of experts’ opinions. This evaluation revealed (a) the SA specifications that include choreography models built using the designed CML contribute to SA informative improvement in testing and protocol development phases, (b) insights based on which additional improvements can be made and (c) suggestions regarding how to facilitate its integration into the existing ES development process.

The results from the research presented in this publication answer RQ3, which is: to what extent is the designed choreography modelling language applicable in the embedded systems domain from the practitioners’ point of view? The analysis of practitioners’ viewpoints revealed that the CML specifications can complement SA documentation and lead to information improvements in testing and protocol development phases. For testing, the added choreography specifications can reduce testing time by reducing the number of test cases needed for the verification of software functionality. For protocol development, the choreography specification can reduce the maintenance burden and improve the performance of the components. In addition, the analysis indicated additional ES development aspects that can be supported by CML and thus contributed to the findings presented in PIV and to the answer to RQ2.

4.6 Answers to Research Questions

The answers to three RQs that guided this study were derived based on the results from the individual studies. These studies and their results were described in five publications that are included in this dissertation (PI–PV), and their overviews are presented in a previous section. The ways that the individual contributions from these studies formed the answers to three RQs is presented in this section.

Answer to RQ1: What are the requirements for choreography modelling language in embedded systems domain? The answer to RQ1 provides the foundational knowledge needed for the CML design and thus for supporting choreography modelling in the ES domain. This knowledge consists of ten design requirements that were derived based on the analysis of the data collected throughout the studies presented in the included publications. Table 5 presents these requirements.
<table>
<thead>
<tr>
<th>Id</th>
<th>Design requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The CML should capture the format of participant’s access constraints in a structured way</td>
</tr>
<tr>
<td>2</td>
<td>The CML should differentiate the participants which expresses ad-hoc behaviour on the network</td>
</tr>
<tr>
<td>3</td>
<td>The CML should capture participants’ heterogeneous technical and technological details</td>
</tr>
<tr>
<td>4</td>
<td>The CML should support the description of various service invocation mechanisms</td>
</tr>
<tr>
<td>5</td>
<td>The CML should capture the real-time information needed for executing the choreography scenario in ES domain</td>
</tr>
<tr>
<td>6</td>
<td>The CML should support a structured way of describing its language construct</td>
</tr>
<tr>
<td>7</td>
<td>The CML should differentiate the communication layer on top of which the interactions being modelled are occurring</td>
</tr>
<tr>
<td>8</td>
<td>The CML should capture the real-time information that is specific to the application domain or industry branch</td>
</tr>
<tr>
<td>9</td>
<td>The CML should include the information needed for the visualisation of data that is carried by the message</td>
</tr>
<tr>
<td>10</td>
<td>The CML should support the definition of criteria for clustering the interactions</td>
</tr>
</tbody>
</table>

Each design requirement denotes a particular ES development aspect that was identified as choreography relevant and thus needed to be supported by the choreography modelling language design. Detailed descriptions of the design requirements are presented in PIV and PV, while the description of the requirements’ implementation is presented in Chapter 5.

**Answer to RQ2: How can the identified requirements be applied in the design and implementation of the choreography modelling language?** The identified design requirements can be applied as guidelines for the design of the choreography modelling language. These requirements suggest which choreography-relevant aspects and phases need to be supported so that the language is applicable throughout the ES development lifecycle.

In this study, the identified design requirements were incorporated into the design of the BPMN [81]. The incorporation of the requirements was realised by supplementing BPMN’s meta-model with the information content that supports the requirements. Consequently, the expressiveness of the BPMN was increased in a way that enables its applicability in the ES development domain. Four steps guided the supplementation of the BPMN with additional information content. These steps are presented in Table 6.
Table 6. Design steps.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Familiarization with the BPMN meta-model</td>
<td>Detailed inspection of the BPMN's meta-model for the purpose of understanding how its constructs can be supplemented with additional information.</td>
</tr>
<tr>
<td>2. Identification of the classes</td>
<td>Examination of the BPMN's meta-model and the design requirements for the purpose of defining the information content.</td>
</tr>
<tr>
<td>3. Design</td>
<td>Supplementation of the BPMN's meta-model with the information content.</td>
</tr>
<tr>
<td>4. Summarization</td>
<td>The documentation of what is done during the CML design.</td>
</tr>
</tbody>
</table>

The description of these steps is presented in PIV, while the exact information content in the form of classes and attributes that were derived is presented in PIII and PIV. The implementation of the CML and its editor and how they support the design requirements are presented in Chapter 5.

**Answer to RQ3: To what extent is the designed choreography modelling language applicable in embedded systems domain from the practitioners' point of view?** The answer to RQ3 is based on an evaluation of the CML in the field. The analysis of the evaluation data suggests that the CML is applicable in the ES domain and thus can advance the informative aspect of SA documentation. This suggestion was observed in the ES testing phase and in protocol development practice in which choreography modelling was viewed as a way to highlight the interactions between specifications and components developed by autonomous development teams. The advanced information content of SA documentation improves communication among stakeholders, consequently leading to reduced development time, numbers of errors and maintenance burdens.

The description of the evaluation with industry experts based on which the answer to RQ3 was derived is presented in PV.
The CML and its editor were implemented during the study, and their implementation was based on ten design requirements that were derived from the analysis of the industry partners’ practices and the literature that focuses on the utilisation of choreography in the ES domain. This chapter presents details about the CML and its editor as well as the ways they address the identified requirements with their features. Details about the implementation technologies and the rationales behind the selections are presented in PIV.

The language editor allows for the development and maintenance of the choreography scenarios using the CML that was designed in this study. This editor was created to be a visual modelling environment whose main parts include the main canvas, palette, language construct properties and tree navigation. The design of the choreography scenario is presented on the main canvas. The language constructs available to users are listed in the pallet, from which the constructs are dragged and dropped on the main canvas. Each language construct has its own set of properties, and these properties are listed in the language construct properties area once they are selected. Tree navigation facilitates navigation through the designed choreography scenario by presenting it in the form of tree. The overview of the language editor with the main parts labelled in red is presented in Figure 6.

The choreography language was designed based on the requirements that were derived from the analysis of the case company practices, scientific literature and the feedback from the industry experts who evaluated the language. The analysis and the evaluation were published in two publications that are included in this dissertation (Publication IV and V). The explanations of how these requirements are addressed by the language design and supported by the tool are aligned with the explanations in these publications and consist of two parts. The first part describes the requirements by presenting the context and the problem based on which the requirements were derived. The second part describes how the requirement is addressed by the language and supported by the tool.
Constraint Based Access

Context: The number of participants within an ES can be large. These participants are not involved in all choreography scenarios, nor do they need to be able to interact with all other participants. To control the access and the utilisation of participants’ resources, participants need to be aware of each other and need to be enabled to grant or deny access based on agreements. Problem: There are different technical solutions that can control the access to participants’ resources. Each solution has its own format to structurally describe the constraints. The information about the format should be related to the corresponding participants that use the format. To address this problem, this design requirement was derived:

**DR1:** The language should capture the format of participants’ access constraints in a structured way

DR1 is implemented by adding an attribute to the participant language construct and by relating it to the class in which the details about the constraints can be recorded. Using the tool, users can select the participant whose access constraint they wish to edit.
and add the format of the constraint in the corresponding property field. As an example, in Figure 7, the Lock controller participant uses ECL rules to describe the access rules to its resources. The participant and the property are indicated by red rectangles.

**Ad-Hoc Networks in Choreography Scenario**

Context: Participants engaged in choreography can be considered nodes in the network. These nodes are assumed to always be connected, available and ready to process an incoming message. Problem: The participants in a choreography scenario can form an ad-hoc network in the ES domain. Ad-hoc networks denote networks in which the participating nodes can freely join (connect) or leave (disconnect) the network at any point during scenario execution. The reason for this behaviour is that participants are often mapped to mobile devices, which freely move in and out of network coverage areas. Based on this, the following design requirement was derived:
DR2: The language should differentiate the participants which express ad-hoc behaviour on the network

DR2 is implemented by adding an attribute to a participant language construct that indicates whether the participant expresses the ad-hoc behaviour or not and by relating it to a class that captures the details regarding the ad-hoc behaviour. Depending on the attribute setting, the visualisation of the participant in the main canvas will differ. In Figure 8, the unlock processor participant’s ad-hoc attribute was set to be true, which caused the colour change in the language construct.

Fig. 8. DR 2 Ad-hoc networks in choreography scenario.

Technical and Technological Heterogeneity

Context: Large and networked ES encompasses participants that are built on heterogeneous technical and technological platforms. This heterogeneity denotes different hardware, processing capabilities, communication technology, interfaces, message formats and implementation technology. Problem: Choreography specification is primarily used as an analytical tool and for documentation purposes. In an MDE context, the specified choreography scenarios are also parsed by tools to extract the information for the subsequent development phase or code generation or to manage the traceability. Including the technical and technological data in the language is necessary because it
enables seamless code generations for various platforms as well as the deployment of the generated code. The following design requirement addresses this need:

**DR3: The language should capture participants’ heterogeneous technical and technological details**

DR3 is implemented by adding an attribute to a participant language construct that indicates the target platform and by relating it to additional classes that capture different aspects of heterogeneity. The classes capture aspects such as the central processing unit type, number of processing cores, amount of random access memory, network type and data transfer rate. The target platform attribute of the selected participant is presented in Figure 9.

![Figure 9. DR 3 Technical and technological heterogeneity.](image)

**Service Invocation Variants**

Context: To exchange messages, participants in the choreography scenario invoke one another’s services. To invoke a service, a participant who is sending a message must know various details, such as the location of the recipient and the invocation strategy. Problem: Technical and technological heterogeneity have led to various service invocation mechanisms. Each of these mechanisms relies on its own set of data to resolve the service call and to invoke the service. Capturing the data needed for the invocation should be supported by the language, and based on this need, the following design requirement was derived:
DR4: The language should support the description of various service invocation mechanisms

DR4 is implemented by adding two attributes to a message flow language construct. The message flow language construct relates the message to the participant who sends the message. Two attributes allow users to record whether the invocation strategy should push or pull as well as the type of invocation. Based on the invocation strategy and type, the editor allows for the recording of additional data for invocation and visually differentiates the message flows on the main canvas. Figure 10 presents these attributes and two different visual representations of message flow language constructs based on the values of those attributes.

Fig. 10. DR4 Service invocation variants.

Generic Real-time Execution Information

Context: Choreography scenarios focus on interactions between services owned by autonomous participants. In the ES domain, the execution of services is often constrained by different real-time constraints. These constraints impose strict duration times for
services to process the incoming messages and to deliver the response. Problem: Current languages for choreography offer only partial support for real-time requirements. This support is also heavily tailored to the particular area of application and thus is not applicable in other areas. To address this problem, the language should enable a generic way of capturing real-time requirements for choreography, but it should also allow users to define attributes for real-time descriptions that are specific in their own areas of application. Hence, this design requirement was derived:

**DR5: The languages should capture the real-time information needed for executing the choreography scenario in ES domain**

DR5 is implemented by relating the participant language construct with the class that captures the real-time execution data, such as the execution start, end, maximum duration and delay between executions. Users can record and edit these data through the properties of the language constructs. Figure 11 presents the participant language and its real-time execution details.

![Fig. 11. DR5 Real-time execution.](image)

**Supplementary Information of the Language Constructs**

Context: Language constructs that are used to describe choreography scenarios often require additional descriptions. These descriptions are used as an additional means to
document the language construct as well as an additional source of data that facilitates the execution or translation to the subsequent specification. Problem: Current choreography languages allow only free-form descriptions of language constructs. To facilitate the execution or translation of the specification, the language constructs must be described in a structural way that is known to the tool that parses that specification. To address this need, the following design requirement was derived:

**DR6: The language should support a structured way of describing its language constructs**

DR6 is implemented by extending each language construct to a class that enables the additional description of the language construct. The additional descriptions are captured in the key-value form in which both the keys and the values can be simple variables or specific commands that are interpreted by the tool. Figure 12 presents an example of the way message flow is supplemented with an additional description.

![Fig. 12. DR6 Supplementary information of language constructs.](image)

**Differentiation of the Communication Layers**

Context: Embedded systems tend to encompass a large number of participants that communicate by exchanging an even larger number of messages. These messages are
exchanged for different purposes, so the communications can be differentiated and organised in structures called layers. For example, at least three communication layers can be identified in the telecommunication domain. The first is the end-user data delivery layer, which is also known as the data plane and includes messages that carry data for end-users. The second is the route definition layer, or the control plane, which includes messages that define the routes through which the end-user data is delivered. The third is the system configuration layer, which is also known as the management plane and includes messages for system management, maintenance and upgrades. Problem: Current languages for choreography modelling do not differentiate between message exchanges that occur on different communication layers. Thus, the following design requirement was derived:

**DR7: The language should differentiate the communication layer on top of which the interactions being modelled are occurring**

DR7 is implemented by adding an attribute to a choreography scenario that captures the communication layer. Depending on the value of the attribute, the visualisation of the sequence flow language construct will differ in the main canvas by presenting arrows that are related to the choreography tasks with different colours. Figure 13 presents the ways in which the colours of the sequence flow change based on the value of the attribute.

**Domain-Specific Real-Time Execution Information**

Context: Embedded systems that are developed in different application domain can have different needs regarding the specifications of real-time executions. As indicated by experts, some ES can execute continuously, while some execute in discrete time paradigms. Problem: Real-time information provided by DR5 is seen as too generic and thus insufficient for practical use. Consequently, the generic real-time information highlighted in DR5 needs to be supplemented with information that is content-specific to different application domain.

**DR8: The CML should capture the real-time information that is specific to the application domain or industry branch**

DR8 complements the DR5 by adding more details to the information content that supports the real-time execution. These details are merged with the content provided
based on DR5 and can be accessed through the properties of the language constructs. Figure 11 demonstrates real-time execution details.

Visualization of Data

Context: In today’s ES, the communication between different participants is realised using different communication technologies. These technologies rely on their own message formats and data types that define the payload that is carried within the message. Problem: Current languages for choreography modelling do not support differentiation of the payload that is carried by the message that is exchanged between participants. This motivated the following design requirement:

**DR9:** The language should include the information needed for the visualization of data that is carried by the message

DR8 is implemented by allowing users to record the payload type for the message language construct. Based on the values of the message payload attribute, the visualisation of the message on the main canvas will differ by displaying the values of the attribute, thus allowing users to differentiate between what the message that is being exchanged contains. Figure 14 presents the visualization of the message data.
Visualization of Interaction

Context: The number of interactions in the ES domain can increase until the visualisation of these interactions becomes difficult or impossible. This makes choreography scenarios large and complex, while the overall value of the specification decreases. The main reason for the value decrease is that the large and complex specifications contribute to the increase (rather than decrease) of the developers’ cognitive burdens. Problem: Current languages for choreography modelling do not support the definition of criteria for clustering the interactions. Based on this need, the design requirement was derived:

**DR10: The language should support the definition of criteria for clustering the interactions**

DR10 is implemented by extending each language construct with the class that captures the criteria for clustering or categorising the interactions. Once the criteria are defined and the language construct is categorised using those criteria, the editor allows the two types of visualisations of the categorised language constructs. The first type is the tabular visualisation in which the constructs are presented in rows, and the categories are presented in columns. An example of tabular visualisation is presented in Figure 15 and the participants in the choreography are presented based on different criteria.
The second type of visualisation is the hierarchical representation of the choreography scenario. The hierarchical representation allows users to organise the scenario on several hierarchical levels in which each level captures the interactions based on some of the defined criteria. Hierarchies are realised using the sub-choreography language construct that encompasses the choreography scenario on the lower hierarchical level. This construct is presented in Figure 16.

Readers interested in contributing, testing or evaluating the presented language and editor can freely download the environment from the web location given in [62].
6 Discussion

This chapter discusses the study results in a scientific context and the validity of the research conducted. The scientific context consists of the studies that motivated the CML design and influenced various design decisions made during the CML design. The validity of the research is discussed on two levels. The first level focuses on the validity of the overall research design. The validity of the research design was assessed by discussing the study in the context of seven guidelines for performing and evaluating design science research, which were proposed in [17]. The second level focuses on the trustworthiness of the qualitative research approach that was used during the study. The trustworthiness of the approach was assessed following the strategies proposed by Shenton [82].

6.1 CML in its Scientific Context

This study addresses the identified communication challenges in the ES domain caused by the lack of information content in SA documentation by enabling the specification of the choreography viewpoint. Specifying the choreography viewpoint in the ES domain was motivated by the studies conducted in the enterprise systems domain, which indicated on the potential of choreography to facilitate the communication among involved stakeholders [30–32]. Following these indications, during this study, a CML that is applicable in the ES domain was designed. This design was based on existing choreography related design knowledge and consolidated various ES development aspects that are supported by the existing languages used for choreography modelling in the ES domain.

The CML was based on the BPMN language and extends its design to additional information content. Several reasons motivated the reuse of the BPMN’s design, most relevant of which are its status of the de-facto standard for process modelling in the enterprise systems domain [81, 83] and the suitability of its language constructs for capturing the choreography viewpoint [33]. Designing on top of the BPMN compared with designing from scratch can facilitate a seamless and quick adoption. The main reasons for this include a large community of developers that are familiar with the BPMN and can easily master the redesigned language and the presence of BPMN as a de-facto standard in university level curricula. The suitability of BPMN’s language
construct in capturing the choreography viewpoint was another reason for its selection. This was revealed by Cortes-Cornax et al. [33], who evaluated BPMN using the extended quality framework. The evaluation results showed that aside from minor limitations, BPMN is a suitable language for choreography modelling and is therefore a suitable foundation for the CML design.

The CML extends BPMN with information content that supports various choreography relevant ES development aspects. These aspects were consolidated based on the analysis of existing languages and the data collected in an industry context. The designed CML thus supports aspects such as real-time execution, constraint-based access, service invocation variants, ad-hoc networking and technical and technological heterogeneity (see PIV and PV for details). Due to the design being built on existing knowledge, which is embodied by BPMN, and the support for the identified choreography relevant ES development aspects, the CML can capture the choreography viewpoint in a way that provides sufficient information for various stakeholders involved in the ES development process.

Communicating the needed information to stakeholders involved in ES development by developing and exchanging specifications is a prerequisite for successful communication. The choreography specification, which is included in the designed CML, contains information content that is relevant in ES development and consequently supports communication among stakeholders. Hence, the CML design fulfils one of the explicit requirements for choreography modelling languages, which is to support communication among its users [33, 34].

Another feature of the CML design can also be viewed as support for communication in ES development. The CML is designed with MDE in mind and allows for the translation and relations of choreography specifications to other specifications used during ES development. Study results presented by Koehler et al. [32] showed that the ability to translate business, or higher-level specifications, to implementation, or lower-level specifications, can facilitate communication of changes made during the development. Consequently, the CML feature that it can be translated to other specifications can also be viewed as communication support for stakeholders involved in ES development.
6.2 Validity of the Research Approach

To support the adoption of the design science framework, Hevner et al. [17] proposed seven guidelines for the realisation and evaluation of design science studies. To demonstrate the validity of the research approach adopted in this study, this section first introduces the guidelines and then presents how these guidelines were adopted. Hevner et al.’s guidelines are shown in Table 7.

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Design as an artifact</td>
<td>Design science research must produce a viable artifact in the form of a construct, a model, a method or an instantiation.</td>
</tr>
<tr>
<td>2: Problem relevance</td>
<td>The objective of design science research is to develop technology-based solutions to important and relevant business problems.</td>
</tr>
<tr>
<td>3: Design evaluation</td>
<td>The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.</td>
</tr>
<tr>
<td>4: Research contribution</td>
<td>Effective design science research must provided clear and verifiable contribution in the area of the design artifact, design foundation, and/or methodologies.</td>
</tr>
<tr>
<td>5: Research rigor</td>
<td>Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.</td>
</tr>
<tr>
<td>6: Design as a search process</td>
<td>The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.</td>
</tr>
<tr>
<td>7: Communication of research</td>
<td>Design science research must be presented effectively to both technology-oriented and management-oriented audiences.</td>
</tr>
</tbody>
</table>

1: Design as an artifact. The ICT artefact designed in this study is the CML for service choreography in the ES domain. This design was realised as a customisation or redesign of an existing language that is used for service composition modelling in the enterprise systems domain. The redesign was guided by ten design requirements that were derived based on the analysis of the (a) interviews with ES practitioners, (b) company specific data and (c) relevant scientific literature.
2: Problem relevance. The problem regarding communication among different stakeholder groups caused by the lack of information in SA documents was resolved in the study. To ensure the relevance, this problem was defined based on the cooperation with industry experts and by relating it to the existing scientific literature. The cooperation with industry experts ensured the business need to address this problem. The relation to scientific literature ensured that the problem was interesting from a scientific point of view and thus worthy of scientific investigation.

The cooperation with industry experts was based on high-level and industry-relevant topics that were established by the AMALTHEA research project [15] and consisted of a number of workshops. During these workshops, industry experts presented ES development problems they have experienced in their companies. Afterwards, the presented problems were reviewed in the context of the existing scientific literature that focuses on ES development. This review was conducted with senior researchers and resulted in several studies [7–10] to which a subset of the presented problems could be related. This subset of problems confirmed the indications in these studies, according to which the communication among stakeholders is often hampered by the lack of information in SA documents. Consequently, the defined problem is underpinned with the existing scientific literature and addresses the business needs of ES development companies.

3: Design evaluation. The ICT artefact designed in this study was evaluated by adopting the informed argument approach [17]. This approach is a descriptive method for evaluating artefacts in design science research. The informed argument approach relies on information from the knowledge base to build a convincing argument that supports the artefact. In this study, this approach was extended by eliciting experts’ opinions about the usefulness of the artefact in the ES development domain. Thus, the informed argument was strengthened by information from the environment, which contributes to the soundness of the evaluation.

The arguments from the knowledge base supported the selection of the BPMN as a base language on which the CML was designed and the technologies used for the implementation of the language and editor. These arguments were retrieved by means of a literature review and were presented in PIV. The arguments based on experts’ opinions were elicited by means of the focus group method, which is a recommended method for artefact evaluation in design science research [79]. These arguments were described in PV.
4: Research contribution. The research contribution of this study is twofold. The first contribution is the ten design requirements that are presented in PIV and PV and in Chapter 5. These requirements guided the design of the ICT artefact but can also be seen as generic requirements for the development of future choreography modelling languages in the ES domain. The second contribution is the empirical evidence that emerged from the language evaluation. This evidence revealed ES development areas in which the language can be used to improve the informative aspect of SA documents and consequently to improve communication among stakeholders.

5: Research rigor. Research rigor is derived from the effective utilisation of the knowledge base to methodologically design and evaluate an ICT artefact, which is the CML [65, 67]. This study consisted of five research activities that were coordinated with the design science research framework toward a common goal. As explained in Section 2.2, each study addressed a specific area of the framework and adopted a specific research method to produce a partial study result. The reasons behind the selection of each research method and how they were adopted were described in the publications that are included in this dissertation.

6: Design as a search process. Studies guided by the design science research framework can be characterised by three regulatory circles, which are explained in Section 2.1. Each circle consists of several iterations that seek to provide data from the environment and knowledge base and to utilise the data during the design and evaluation of the ICT artefact. These circles can also be seen as search activities that identify the practical problem and constraints from the environment, search for suitable technologies, tools and methods from the knowledge base and search for satisfactory design alternatives based on the outputs from the previous circles. Therefore, during the study, the search of the environment was realised by exploring (a) the challenges in SA development, (b) the influence of the middleware features on choreography modelling language and (c) the opinions of the industry experts regarding CML use. Additional details and results from these explorations are presented in PI, PIII and PV, respectively. The search of the applicable knowledge was conducted by exploring the scientific literature, and the results from this exploration are presented in PII. The search for the satisfactory CML design was conducted in close collaboration with industry experts and was based on the data collected from the environment and knowledge base. The satisfactory design was achieved by trying various design alternatives and by checking them against the derived design requirements. These requirements and their evaluation by CML implementation are presented in PIV.
7: Communication of research. This dissertation was developed based on a compilation of five scientific publications. These publications have been presented to wider audiences in various international scientific forums, such as conferences and journals. Additional details about these forums and publications are presented in Chapter 3.

6.3 Trustworthiness of the Study

The individual research efforts conducted within this study relied on a number of qualitative research methods [64, 68] to derive the results. Compared with quantitative studies that rely on the assessment of reliability and internal, external and construct validity to ensure the validity of the study, qualitative studies rely on credibility, transferability, dependability and confirmability criteria to ensure the trustworthiness of the study [84]. Trustworthiness criteria and their descriptions are presented in Table 8.

Table 8. Trustworthiness criteria for qualitative studies (adopted from [84]).

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I: Credibility</td>
<td>Ensures that the study results are accurate from the perspective of the participants who provided the data. In quantitative studies, this criterion corresponds to internal validity.</td>
</tr>
<tr>
<td>II: Transferability</td>
<td>Represents the degree to which the study results can be generalised. In quantitative studies, this criterion corresponds to external validity.</td>
</tr>
<tr>
<td>III: Dependability</td>
<td>Denotes the stability of the study results in changing contexts in which the research was conducted. In quantitative studies, this criterion corresponds to reliability.</td>
</tr>
<tr>
<td>IV: Confirmability</td>
<td>Refers to the extent to which other researchers or participants confirm and corroborate the study results. In quantitative studies, this criterion corresponds to objectivity.</td>
</tr>
</tbody>
</table>

The criteria presented in Table 8 were addressed by adopting the strategies proposed by Shenton [82]. These strategies, which were grouped according to their corresponding criterion, and how they were adopted throughout this study are as follows.

I: Strategies for promoting the credibility of the study include:
The adoption of research methods that are well established in the field. The adopted research methods include interviews, focus groups, systematic literature reviews, a thematic analysis and DESMET, which are well-established and accepted research methods in the software engineering field.

The development of an early familiarity with the culture of participating organizations. The University of Oulu and the companies involved in this research collaborated on a number of workshops and meetings during the AMALTHEA project preparation in which they became familiar with the organisational rules and practices and built a mutual trust. This collaboration also contributed to the relations with the Case Company, which was the main partner in this study. Researchers and experts from this company already collaborated on a number of research projects and were consequently familiar with mutually beneficial ways to work together.

Tactics to help ensure the honesty in informants. The long-term cooperation between the university and the partner companies also contributed to this strategy. In addition, experts who participated in the study were aware of the opportunity to refuse participation, the importance of the accuracy of their responses, the freedom to express their opinions and their rights to withdraw from the study at any time. The researchers worked under a non-disclosure agreement, which provided additional guarantees related to sensitive information that was collected and used.

Sampling. This study’s focus was SA documentation. To ensure that the collected data was relevant, the subjects who provided the data were software architects or experts closely related to SA and ES design. This method of sampling can be characterised as a purposive sampling, which is often adopted in qualitative studies.

Frequent debriefing sessions. A large number of meetings, workshops and demonstrations was conducted during this study. These events enabled researchers and industry experts to cooperate, exchange opinions about various aspects of the artefact design and define directions regarding how the research work should continue.

Background, qualification and experience of the investigator. This study is a doctoral dissertation project. The principal investigator was a junior researcher with limited research experience who was supervised by three senior researchers throughout the entire study. This allowed the principal investigator to gain valuable experience and to improve research-related expertise.

Detailed description of the phenomenon under scrutiny. The phenomenon under scrutiny is the choreography modelling language that is supported by an editor. The rationale behind the design was best explained in PIV and PV, which described the
design requirements of the CML and the motivation behind the requirements. A detailed description of the CML and how it addresses the identified design requirements are described in Chapter 5.

**Examination of the existing research findings.** Existing knowledge about the utilisation of choreography in the ES development domain was examined by means of a systematic literature review. The results of this review ensured the alignment of the work in this study with the existing research efforts. This study is presented in PII, while the study overview is presented in section 3.2.

II. Strategies for promoting the transferability of the study include:

**Detailed description of the study context.** Details about the context of each individual study such as the number of participants, their roles in ES development process and years of experience are presented in publications that are included in the dissertation. Details about the environment within which the study was conducted include the AMALTHEA research project, the case company and the MDE as an overall engineering approach adopted in partner companies. More details about the environment are presented in Appendix 1.

**Detailed description of the phenomenon under study.** The concept of the choreography as an integral part of SOA and its influence on SA documentation represent the phenomenon under study. Detailed descriptions of these concepts and how they are defined in this study are presented in Appendix 1 and in publications included in this dissertation. Readers interested in additional details regarding these concepts can consult the literature sources listed in the reference section.

III. Strategies for promoting the dependability of the study include:

**Detailed description of the research design and its implementation.** The research design implemented in this dissertation was organised according to the design science framework. Chapter 2 presents the description of the design, the research methods that were used and the implementation details of the study as a whole.

**Operational details of data gathering.** This dissertation consists of five publications that report studies whose results contributed to the fulfilment of the overall study goal. The data collection methods used in these studies include interviews, focus groups, a systematic literature review and company specific documents. These methods and their adoptions for individual study purposes are described in publications (PI–PV).

IV. Strategies for promoting the confirmability of the study include:
**Detailed description of the research design and its implementation.** As was the case with the dependability criteria, the descriptions of the research design, methods and research implementation contribute to the conformability of the study. Additional details are presented in Section 2.2.

**Recognition of the study limitations.** The design requirements based on which the CML was developed were derived based on information from interviews with software architects from different companies, existing literature that focuses on the topic, middleware analysis and close cooperation with industry experts from one of the partner companies (see appendix 1 for more details about this company). The close cooperation between researchers and the company led to a stronger influence from its experts on the CML design compared with the influence of experts from other companies that participated in the AMALTHEA project. Due to this influence, the CML design may better support the need within this specific case company and its industry branch.

Another limitation stems from the MDE approach. The CML was designed based on the assumption that the companies interested in its deployment adopted the MDE approach for ES development. This assumption influenced the selection of (a) the approach to design the CML as a way to support choreography modelling in ES and (b) implementation technologies that enabled properties that are relevant in an MDE context. These properties include possibilities to translate the specification to another specification, traceability, code generation and executing various automatic operations in addition to the specification. Although these properties are not exclusive to MDE, the feasibility of deploying and using the CML in a non-MDE context was not examined, which may limit its adoption in some companies.

Finally, the evaluation of the CML was conducted using data collected from the literature and focus group sessions, and a thematic analysis was used as a qualitative method for deriving the evaluation results. The qualitative methods approach was considered suitable for the evaluation because the phenomenon of interest is new and unexplored in the ES domain. This method resulted in argumentative descriptions 6 regarding the ways experts perceive the CML in practice and in clear indications that benefits such as the reduction of testing time and maintenance burdens can be expected of its use; however, the approximate rates of the claimed reductions were not analysed due to limited time and resources.

---

6See also the adoption of Hevner et al.’s guideline No.3 that is presented in section 6.2
7 Conclusion and Future Work

The communication challenges among stakeholders involved in ES development that are caused by the lack of information in SA documentation represent the practical problem that was the focus of the study presented in this dissertation. As a solution to this practical problem, choreography modelling was utilised, which is an SOA-specific technique; however, the choreography modelling languages currently used were found to be insufficiently expressive in capturing the choreography relevant aspects that are typical in an ES domain. The specifications designed using these languages therefore cannot communicate the information needed by different stakeholder groups. Thus, the objective of this study was to advance the design of choreography modelling languages by adding information content that enables their use throughout ES development phases. To achieve this objective, a comprehensive choreography modelling language was designed.

The CML designed during this study is a product of design science research. The ICT artefact that was designed is the CML that is applicable in the ES domain. During the study, various research efforts aimed to collect the knowledge needed for the designed CML. These research efforts and their results were presented in five scientific publications that are included in this dissertation. The summary of these results include the categorised SA development challenges, an overview of choreography use in the ES domain, the influence of middleware features on choreography modelling languages, design requirements and the CML evaluation; however, the main result of this study offers new design-related knowledge. This knowledge consists of ten design requirements for choreography modelling languages that should be considered if the language is to be used in the ES development process. Ten design requirements were evaluated in an academic context, while the CML and its editor designed based on these requirements were evaluated in an industry context.

The evaluation of requirements in an academic context was realised through the implementation of the CML and its editor in consideration of their future adoption in an industry context. This was achieved by (a) the utilisation of implementation technologies that allow easy integration into the MDE development process and (b) designing the CML based on an existing and mature modelling language that is commonly used in the industry. These two areas were viewed as important due to the trend that indicates a
growing adoption of MDE in the ES domain and due to the opportunity to reuse the knowledge experts already have about the existing language in a new context. The evaluation indicated that the requirements can be implemented in the currently used technologies and that they can be supported by the choreography modelling language.

The evaluation of the CML and its editor in an industry context was realised through the collection and the analysis of the industry experts’ opinions about their impact on the ES development process. The analysis results revealed that choreography modelling that uses CML can improve the information content of SA documentation in testing and protocol development areas and thus can facilitate communication among stakeholders. In testing, the additional choreography specifications can reduce the testing time by reducing the number of test cases needed for the verification of software functionality. In protocol development, it reduces the maintenance burden and contributes to system performance.

In conclusion, this study presents new knowledge regarding the design of choreography modelling languages that are applicable in the ES domain. This knowledge includes ten design requirements whose feasibilities were evaluated based on CML implementation and its evaluation by industry experts. The evaluation results showed that choreography models specified using the designed CML improve the information content of SA and consequently can address the communication challenges among the stakeholders involved in the ES development process.

Future work. Three topics were identified to complement the knowledge provided by this study. The first topic is related to further customisation of the language. This topic addresses the study limitation caused by the stronger influence of experts from the case company on the CML design (see section 6.3). The language and editor can be further enriched by additional data and functionality so that it more accurately meets the needs of experts working in other industry branches. The second topic is related to an in-depth analysis of the CML’s impact on the ES development process. This topic also addresses the study limitation that was caused by the lack of the clear metrics regarding the claimed benefits. The lack of these metrics can hinder the deployment of CML because they represent the important component of calculating the return on investment, which often serves as an indicator when deciding whether or not a company should deploy new technology. The future work on this topic should rely on quantitative methods to provide accurate metrics that will facilitate companies deciding whether to deploy CML (or choreography modelling) in their specific context. Finally, the third topic is related to the visualisation of the model. Similarly, the way as the participants in
a choreography scenario can be presented with the language constructs using the tabular representation; this topic requires additional exploration of alternative visual solutions that can illustrate different aspects of the model within a choreography scenario. For example, improvements in defining and maintaining rules for complex gateways are a potential starting point for researching this topic.
References

64. Miles MB & Huberman AM (1994) Qualitative data analysis: An expanded sourcebook. Sage.
Appendix 1 Outline of the Foundational Concepts

AMALTHEA was an ITEA2 founded international research project [15] whose goal was to reduce the effort that is needed to exchange data between development tools. This goal was achieved by developing an open and extendable tool integration platform and a set of tools for ES development. The integration platform and tool assumed that model-driven engineering was an overall ES engineering approach, and this assumption positioned models, modelling languages and model transformation as the main development principles. The main implementation technology is Eclipse technology, and the developed platform and tools are distributed under the Eclipse public licence.

The AMALTHEA project lasted between July 2011 and April 2014. Nine industry partners and five research institutions from Germany, Finland and Turkey participated in the project. The Finnish consortium consisted of two large ES development companies and one research institution (University of Oulu). Throughout this dissertation, researchers from Oulu University cooperated with most of the partners involved in the project; however, the majority of the research during this dissertation was realised in cooperation with a single company from Finland. This company is referred to as the Case Company in this text.

The Case Company is a global information and communication technology (ICT) equipment manufacturer headquartered in Finland. It employs more than 60,000 employees across more than 100 countries and conducts sales in more than 150 countries worldwide. Its main product is a large system that includes hardware equipment, embedded and non-embedded software and services related to the product. The Case Company joined the AMALTHEA research project to improve its practices related to the software architecture of ES. Hence, ES development experts cooperated with researchers from Oulu University for the solution proposal and development.

To ensure the alignment of the solution proposal and the Case Company’s ES development practices, two characteristics were considered relevant. The first characteristic was the software architecture development approach. The Case Company adopted model-drive engineering practices as an overall approach to design and to specify software architecture. The model and the corresponding modelling language used is UML 2.4 and the specifications are primarily used to communicate the architecture
to different development teams worldwide. The second characteristic focused on the structure of ES implementation. The Case Company relies on components as the main building block of the ES and utilises a middleware product as an execution environment for the components. Along with these two characteristics, supplementing SA documentation with information content that supports the utilisation of SOA was considered a promising way to improve practices for software architecture design.

Service-oriented architecture (SOA) originated from the telecommunication systems development domain [85], but its popularity and wider acceptance is more commonly related to the development of enterprise (or information) systems. Providing a comprehensive definition of SOA is difficult because it has different meanings to different stakeholders in software development. For example, according to [86], SOA is defined as (a) a set of services that are exposed to internal and external partners of the organisation to a business analyst, (b) an architectural style that is based on various architectural patterns and principles to a software architect, and (c) a programming and deployment model realised by standard tools and technologies to programmers. In this dissertation, SOA is defined as a concrete architectural style that is a concrete structure of the ES being developed. According to Erl [87], this structure is governed by eight design principles, which include:

- Standardised service contract: emphasises the standardised specification of a service’s public interface.
- Service loose coupling: emphasises the reduction of dependencies between a service contract, its implementation and service consumers.
- Service abstraction: denotes the need to hide as much service details as possible.
- Service reusability: emphasises the service design in an agnostic functional context.
- Service autonomy: denotes service independence from its execution environment and evolves without influencing its clients.
- Statelessness: emphasises the separation of service and state data.
- Discoverability: emphasises a service’s ability to be searched and found in service registries.
- Composability: denotes the ability to develop new services by composing existing services.

ES was developed following the principles of SOA and shares the same underlying structure. To highlight the logic behind this structure, a reference model for SOA-based systems is proposed in [88]. This reference model uses layers as a means to separate
various concerns regarding the system. The reference model and its layers are presented in Figure 17.

In Figure 17, five layers are presented as rectangles, while the sixth, the integration layer, is presented as a background canvas that integrates them into an SOA-based system. The two bottom layers, which are operational applications and components, represent the service providers because they include the legacy applications and components that implement the services. These applications and components are grouped in dotted rectangles to denote the different application domains in which they are used. The two top layers, which are consumers and compositions, represent the service users because they invoke the services to complete their tasks. The service layer, which is presented in the highlighted rectangle, represents the services that are the main building blocks of the SOA system. The service layer homogenises the applications and components and offers unified and standardised interfaces for their users. Additional details regarding the layers include:
Operational application. This layer represents all applications within an SOA-based system. These applications fulfil a dedicated task within their own application domain, and in most cases, they are implemented using heterogeneous technologies. Applications in this layer include the existing or legacy applications that are reused in an SOA-based solution and the applications specifically built for the purpose of the solution.

Components. This layer consists of components that implement services. These components combine the functionalities of an application or applications that are used within a particular domain and implement them as a service offered by that domain.

Service. This layer consists of services that denote the standardised descriptions of functionalities that are exposed to clients within or outside an SOA system. Exposed service descriptions represent an interface for accessing the functionality provided by service components that reside in the component layer. Service interfaces are stored in specialised registries from which they can be searched, found and invoked through the network.

Composition. The services exposed in the service layer can be composed into high-level structures called processes. Compared with a service that exposes the functionality of a particular domain, a process corresponds to a functionality or a goal of a system as a whole and uses services from different system domains to realise the functionality or goal.

Consumers. This layer consists of applications that access compositions and services to fulfil their tasks. This layer is also known as the presentation layer because it is viewed as an interface for end-users.

Integration. This layer represents the capabilities that enable the integration of all layers into a holistic SOA system by providing various capabilities, such as routing, protocol translation, service repository and message transformation. In most cases, these capabilities are delivered by a middleware product.

The specifications that capture service compositions within the service composition layer represent an important software development artefact. These specifications capture the realisation of system level functionalities or goals in SOA-based systems and thus summarise information about different services developed by different development teams in different application domains. Service compositions are specified based on at least two viewpoints, which are choreography and orchestration [14].
Choreography represents a global, or a neutral point of view of service composition [13]. It focuses on the sequence of interactions that occur between different application domains in an SOA-based system, their role in the composition and the messages they exchange. The application domain denotes the identifiable and logically encircled part of a system that is managed and maintained by an autonomous development team within the company. In choreography and orchestration contexts, an application domain is referred to as the participant, and throughout the dissertation, these terms are used interchangeably.

Orchestration. Orchestration is a service composition from the perspective of a single participant [13]. Compared with choreography, orchestration is a local viewpoint and focuses on the sequence of services that fulfil the functionality of a particular participant. As a local viewpoint of a single participant, service orchestrations are specified, maintained and managed by an autonomous development team within an organisation. Choreography and orchestration are presented in Figure 18.

In Figure 18 choreography is labelled 'A', while the orchestrations are labelled 'B'. The side-by-side comparison shows the basic elements that are used in both specifications and the differences in the ways the two specifications capture service compositions. Choreography focuses on interactions between participants, and in Figure 18, two interactions are presented. Each interaction presents two participants (PX and PY), the role they play in the interaction (Roles a, b, c and d) and the message...
they exchange (M1 and M2). Once the choreography is specified, each participant specifies its own orchestration. The orchestrations are presented on the right side of the page and labelled 'PX' and 'PY' to denote the participant. Orchestrations specify the order of service invocations that are necessary for realising a functionality offered by a participant. To realise the functionality of PX, four services (S1, 2, 3 and 4) were orchestrated, and to realise PY’s functionality, three services (S5, 6 and 7) were orchestrated. The two interactions presented in the choreography were realised as a message exchange between S1 and S5 and between S4 and S7 in the orchestration.

**Model-driven engineering** (MDE) is used as an overall engineering approach for software development. The key characteristic of MDE is the shift of focus from coding to the development of SA. In MDE, SA consists of structured data models and relies on the automatic translation of models and the generation of a program code [59]. Focusing on SA and model development aims to raise the level of abstraction from the program code and to position it closer to the problem domain. This means that models are less burdened by implementation technology details expressed using domain concepts and therefore are more easily understood by domain experts. The translation of models is enabled by specialised tools that are capable of reading the models and translating their constructs into the construct of another model that is used in a subsequent development phase. This characteristic enables traceability between the developed models and concurrent work on design and verification, for example. Automatic code generation aims to produce the working program code based on the information conveyed by the developed models. A high-level view of MDE using the presented characteristics is shown in Figure 19.

![Fig. 19. MDE and its characteristics.](image-url)
Two models and a source code, which are the final results of the software development, are presented in rectangles in Figure 19. Models 1 and 2 capture two software aspects that are relevant to the development. The translation of concepts from one model to another and the traceability between them is represented by the arrow. Based on the information in the models, code generators can automatically produce the source code. Automatic source code generation is represented by dashed arrows in Figure 19.

MDE implies the utilisation of modelling languages and the consideration of middleware products that are used as part of the developed application. Modelling languages consist of concepts defined in a model, and they are used to specify the aspect or the domain of a system for which the model was developed. Since each model in MDE addresses one particular system’s aspect or domain, these modelling languages are often referred to as domain-specific languages or domain-specific modelling languages. For example, both models in Figure 19 would have corresponding modelling languages that are used to develop specifications. A middleware product is the execution environment for applications. These products provide various generic services to the application but also set various constraints that need to be considered during development. Within the MDE approach, the information about the middleware imposed constraints can be recorded in some of the models or can be a part of the code generator.
Original Publications

Following original publications are included in this dissertation:


Publications are included in printed version of the dissertation with permissions from IARIA XPS press (Publication I and III), Elsevier (Publication II), Scitepress (Publication IV) and Springer (Publication V).

Publications are not included in the electronic version of the dissertation.
672. Manninen, Outi (2016) The resilience of understorey vegetation and soil to increasing nitrogen and disturbances in boreal forests and the subarctic ecosystem

673. Pentinsaari, Mikko (2016) Utility of DNA barcodes in identification and delimitation of beetle species, with insights into COI protein structure across the animal kingdom

674. Lassila, Toni (2016) In vitro methods in the study of reactive drug metabolites with liquid chromatography / mass spectrometry

675. Koskimäki, Janne (2016) The interaction between the intracellular endophytic bacterium, Methylobacterium extorquens DSM13060, and Scots pine (Pinus sylvestris L.)

676. Ronkainen, Katri (2016) Polyandry, multiple mating and sexual conflict in a water strider, Aquarius paludum


679. Suoranta, Terhi (2016) Advanced analytical methods for platinum group elements: applications in the research of catalyst materials, recycling and environmental issues

680. Pesonen, Janne (2016) Physicochemical studies regarding the utilization of wood- and peat-based fly ash


682. Ahmad, Muhammad Ovais (2016) Exploring Kanban in software engineering

683. Mustonen, Kaissa-Riikka (2016) Climate change and boreal rivers: Predicting present-day patterns and future changes in hydrological regime and its effects on river communities

684. Karppinen, Pasi (2016) Studying user experience of health behavior change support systems: A qualitative approach to individuals’ perceptions of web-based interventions


Book orders:
Granum: Virtual book store
http://granum.uta.fi/granum/
Nebojša Taušan

CHOREOGRAPHY MODELING IN EMBEDDED SYSTEMS DOMAIN