Process Modeling and Execution in Non-Enterprise System Integration

Miao Fang

This thesis is presented as part of Degree of European Master in Software Engineering
This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 25 weeks of full time studies.

Contact Information:
Author:
Miao Fang
E-mail (UKL): miao.fang@cs.uni-kl.de
E-mail (BTH): mifc09@student.bth.se (841018P724)
E-mail (Siemens): miao.fang@siemens.com

University advisor:
Prof. Dr. Dr. h.c. H. Dieter Rombach
Technische Universität Kaiserslautern

Co-advisor:
Dr. Ludwik Kuzniarz
Blekinge Institute of Technology

Industry advisor:
Georg Leyh, M. Sc.
Corporate Technology, Siemens

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Context. Manufacturing Execution Systems (MES) are information systems that manage manufacturing processes in factories. The execution of MES processes requires non-enterprise integration, which integrates MES applications, services, and automation systems within the factories.

Objectives. We aim at developing a modeling approach that can be used to represent and execute MES processes. Having such an approach would help MES vendors to reduce the development cost to reconfigure systems, in order to achieve better business flexibility.

Methods. In order to understand the state of the art of manufacturing modeling techniques, we perform a systematic literature review (SLR) in scientific article sources, including IEEE Xplore, ACM Digital Library, Compendex, Inspec, and Springer Link. In consideration of the criteria in modeling and executing MES processes, we evaluate the selected process modeling techniques. Based on the result of evaluation, we propose a three-view-based approach to support process execution. We develop a prototype to prove that an MES process can be executed by following our approach. We also conduct semi-structured interviews in industry to validate whether our proposed approach achieves the objectives.

Results. In the SLR, 24 primary studies are selected. Our analysis reveals that existing modeling techniques have limitations to enable process execution. To overcome the limitation, we propose a three-view-based approach, which has an MES process view, an abstract plant view to represent the structure of technical systems, and a mapping view to enable the communication between MES tasks and the technical systems. We develop a prototype as the implementation of our approach, which comprises: a graphical editor for the abstract plant view, a generator of message routes for the mapping view, and a typical MES process to be executed in the context of a warehouse management system. The semi-structured interviews we conducted with three industrial experts show positive feedback to use and generalize our approach in industry, in case comprehensive tools can be established.

Conclusions. Compared to the existing modeling techniques, the three-view-based approach is specifically tailored toward process execution. Based on the feedback from industry, we conclude that applying our approach provides the possibility to achieve better reconfigurability and flexibility of MES.

Keywords: System Integration, Process Modeling, Process Execution, Manufacturing Execution Systems (MES)
I am very grateful to my thesis supervisor in TU Kaiserslautern, Prof. Dr. Dieter Rombach, for giving me the opportunity to write my Master thesis under his supervision. A very special thank goes to my advisor Adrien Mouaffo for the fruitful discussions and suggestions to help me complete this thesis.

Furthermore, I want to thank my co-supervisor in Blekinge Tekniska Högskola (BTH), Dr. Ludwik Kuzniarz. Despite his tight schedule, he carefully commented on my proposal and thesis report. His suggestions helped me improve this thesis, especially in the early phase of my research.

Additionally, special acknowledgments go to the other professors and lecturers in BTH, including Prof. Tony Gorschek, Dr. Darja Smite, Dr. Richard Torkar, Dr. Mikael Svalnberg and Dr. Cigdem Gencel. This is not only because of their help with my Master thesis, but a more important reason is that the knowledge I learned from the lectures in BTH gave me a comprehensive view on software engineering. This has already had significant influence on improving my understanding of software and the research in software engineering.

I highly appreciate the support of my industrial supervisor Georg Leyh at Siemens Cooperate Technology. He provided me with industrial insights that are deeply reflected in the contribution of this thesis. He was always patient and supportive, and encouraged me to realize my idea throughout the research of my thesis. Many thanks go to the other colleagues in my team for supporting me and providing a pleasant working environment.

Many friends helped me during the period of writing this Master thesis. I would like to thank them for helping, reviewing, and suggesting improvements for the thesis report.

Last but not least, I give endless gratitude to my parents. They always believed in me and supported me in all my decisions. Without their constant backup this thesis would never have happened.
# Contents

Abstract i  
Acknowledgments iii  

1 Introduction 1  
1.1 Scope ............................................ 3  
1.2 Aims and Objectives ................................ 3  
1.3 Research Process .................................... 4  
1.4 Outcomes and Contributions .......................... 5  
1.5 Outline .............................................. 6  

2 Problem Formulation 7  
2.1 Warehouse Management System ........................ 7  
2.1.1 Warehouse Domain Model ............................. 8  
2.1.2 A Warehouse Example ............................... 9  
2.2 Limitations of the Existing Solution ..................... 10  
2.3 Research Questions ................................... 11  
2.4 Research Methodologies ................................ 13  

3 The State of the Art 15  
3.1 Planning the Review .................................. 15  
3.1.1 Inclusion and Exclusion Criteria ..................... 16  
3.1.2 Quality Assessment .................................. 16  
3.2 Conducting the Review ................................ 17  
3.3 Reporting on the Review ............................... 17  
3.3.1 Overview of Primary Studies ......................... 18  
3.3.2 Modeling Techniques in MES ......................... 21  
3.3.3 Evaluation ......................................... 26  
3.3.4 Suitability and Applicability ......................... 28  
3.4 Summary ............................................. 30
List of Figures

1.1 The Automation Pyramid .............................. 1
1.2 Non-Enterprise System Integration (MES) ............ 2
1.3 The Process of Industrial Motivated Research [20] .... 4

2.1 The Processes in a Warehouse ......................... 8
2.2 Warehouse Elements of an AKL ....................... 9
2.3 A Warehouse Example ................................ 9
2.4 The MES Process and its Runtime Environment ....... 12

3.1 Conducting the Literature Review ..................... 18
3.2 Notations Proposed to Use in Business/Manufacturing Process Modeling ......................... 19
3.3 MES Specification Framework Models [40] .......... 29
3.4 Applying SpeziMES for the Order Picking Process ... 30

4.1 Enterprise Architecture by Zachman [59] ............... 32
4.2 Necessary Information to Model Vertical Integration . 36
4.3 Meta Model of Abstract Plant View .................... 37
4.4 Sample Model of Abstract Plant View .................. 38
4.5 Communication Between MES and PLCs ............... 40
4.6 Event Exchange in the Order Picking Process ........ 40
4.7 The Three-View-Based Integration Approach ........ 41
4.8 Order Picking Process Implemented in Activiti BPM . 43
4.9 The Pseudo Real-World Model of A Warehouse in EMF 44
4.10 Using Design-Time Elements for Abstract Plant View 45
4.11 Implementation of the Abstract Plant View .......... 46
4.12 The Implemented Generator for Mapping View ........ 48
4.13 Implementation of EIPs in Apache Camel ............ 49
4.14 The Mapping of the Stacker Crane’s Bring Function 50

A.1 Combination of Search Terms in Each Database ....... 67
List of Tables

1.1 Outcome Artifacts and Contribution ....................................... 5
2.1 Research Questions ................................................................. 12
3.1 Search Terms ........................................................................... 16
3.2 Inclusion and Exclusion Criteria ................................................ 16
3.3 Quality Assessment ................................................................ 17
3.4 Selected Studies ...................................................................... 20
3.5 Summary of Papers Applying Multiple Views and Hybrid Techniques 25
3.6 Existing Models with MES Modeling Criteria ............................... 28
4.1 Necessary Information for Modeling and Executing MES Processes 34
4.2 Missing Information When Using Existing Models from SLR ........... 35
4.3 Elements of the Mapping View .................................................... 42
4.4 From Event to Messaging ............................................................ 47
4.5 Views and Prototyping ............................................................... 53
5.1 Ordinal Measures ..................................................................... 56
5.2 General Information about the Interviews ..................................... 56
5.3 The Results of Specific Questions ............................................... 58
Chapter 1
Introduction

In industrial settings, enterprise architecture is commonly modeled as an automation pyramid to structure different applications. Enterprise Resource Planning (ERP) appears at the top layer of the pyramid providing solutions for business process integration within or across organizations. Supply Chain Management system and Customer Relationship Management system are in this layer, for instance. On the next layer, Manufacturing Execution Systems (MES) control relatively short-term manufacturing and production processes. Below MES, Supervision Control and Data Acquisition (SCADA) systems manage and monitor specific manufacturing tasks. SCADA systems inter-operate with Programmable Logic Controllers (PLCs) to perform physical production tasks. The lowest layer contains sensors, actuators, embedded systems, and devices, which actually execute the production tasks. In Figure 1.1, the pyramid on the left presents this five-layer pyramid [28]. From a software application point of view, a three-layer hierarchy presents the contemporary situation of manufacturing automation, even though different models or frameworks distinguish more levels in the control layer [44].

![Figure 1.1: The Automation Pyramid](image)

Enterprise integration, within this pyramid, is considered as the methods, models and tools that can be used to analyze, to design and to continually maintain the applications in different layers [3]. Nowadays, more and more corpo-
rations realize that enterprise system integration gives the opportunities to do business re-engineering to achieve better business flexibility. For this reason, enterprise integration helps corporations on reducing the response time to the changing market and technological evolution, in both industrial and economic context [39, 9].

Non-enterprise integration focuses on the integration of the MES layer and the Control layer. From this perspective, MES need to integrate with the applications, services, and systems within these two layers. This means that the integration of the ERP and MES layers is out of the scope of non-enterprise integration. Figure 1.2 illustrates the integration of the MES process and its execution environment. As can be seen from the MES layer, MES are process-oriented software systems [40]. MES receive queries from the upper layer or external systems. To execute manufacturing processes, MES require the horizontal integration of services in the MES layer. Meanwhile, MES require the vertical integration to the systems and devices in the Control layer. Human actors might also need to participate into certain tasks in manufacturing processes.

![Figure 1.2: Non-Enterprise System Integration (MES)](image)

In order to achieve the integration, analyzing and modeling the MES processes are necessary, but not enough. It is also necessary to look for the solutions about how to use the process model to control its physical execution environment, and how to involve human interaction to the process. In this Master thesis, we plan to conduct research on understanding the integration challenges of existing MES in industry, and look for the possible solution to overcome the challenges from an academic perspective.
1.1 Scope


With the limited time and resources of a Master thesis, it is not possible to fully cover all the function areas mentioned above. Considering the heterogeneity of MES and the execution environment, it is good to take one concrete example of MES to conduct this research. We take a warehouse management system in this Master thesis. A warehouse management system does not cover all the 11 function areas of MES, but analyzing it will help us to understand the real situation in manufacturing plants.

A warehouse management system plays an important role within supply chains and production processes [60]. A warehouse receives and buffers materials. When a picking order arrives, materials are picked from the warehouse, and shipped to the next step. In general, a warehouse management system manages operations, such as receiving, storage, order picking, and shipping. Section 2 explains more details about warehouse management systems, as well as an analysis of limitations and challenges of current system design in industry.

1.2 Aims and Objectives

The aim of this Master thesis is to develop a modeling approach that can be used to represent and execute MES processes. We expect that this approach would help MES vendors to reduce development cost, when facing the changes of requirements from customers, to achieve better reconfigurability and flexibility of MES.

1. Understand the state of the art of existing modeling techniques in MES integration.
2. Analyze the limitations and challenges of applying existing techniques.
3. Propose possible extensions as a candidate solution to overcome the challenges, develop a prototype, and validate the idea of the solution.
1.3 Research Process

This Master thesis is under the co-supervision of professors, researchers, and engineers, from TU Kaiserslautern (Germany), Blekinge Tekniska Hgskola (Sweden), and Siemens Corporate Technology (Erlangen, Germany). This research topic comes from industry. The original problems and scope were relatively wide, compared to academic research topics. This required additional efforts at the beginning phase to narrow down the research scope, and diagnose the real problems. Hence, this Master thesis follows an industrial motivated research process, which is shown in Figure 1.3 [20]. From Figure 1.3, we can also see the knowledge transferred between industry and academia. This process provides a systematic way for conducting research and structuring this Master thesis.

![Figure 1.3: The Process of Industrial Motivated Research [20]](image)

*Step 1: Identify industrial problems.* We conducted several discussions with domain experts and tried to understand the real problems and narrow down the research scope. Several documents were reviewed to understand the functions and the architecture of existing warehouse management systems.

*Step 2: Formulate the research topic in academia.* The challenges of the existing MES systems in industry were identified. A systematic literature review (SLR) was conducted, in order to understand the state of the art of MES modeling and to find whether it is possible to use the knowledge from the existing research to tackle the challenges.

*Step 3: Propose the candidate solution(s).* Learning from the SLR, the limitation of using existing process modeling techniques was analyzed. Then, an improved approach was proposed as a candidate solution to overcome the limitation.
Step 4: Validate in academia. As a proof-of-concept, a prototype of the proposed approach was implemented to prove the modeled process can be executed. This prototype has a graphical editor to model the structure of manufacturing plant, a generator of message routes within the manufacturing plant, and an MES process modeled in BPMN to be executed.

Step 5: Validate the proposed solution in industry (statically). Semi-structured interviews were conducted, in order to get feedback about the solution from domain experts in industry.

Step 6-7: Validate the proposed solution in industry (dynamically), and release the solution. Due to the limited time and research scope, these two steps are not involved in this Master thesis, but can be considered as future work.

1.4 Outcomes and Contributions

The outcomes of this Master thesis are listed as follows, and Table 1.1 explains the expected outcomes and the potential contributions to stakeholders:

1. An evaluation of prevailing modeling techniques in non-enterprise system integration.
2. A three-view-based integration approach, as an extension of the existing modeling techniques, to tackle the integration challenges.
3. A prototype of the candidate solution.
4. A summary of semi-structured interviews as industrial validation.

<table>
<thead>
<tr>
<th>Outcomes</th>
<th>Potential Contributions</th>
</tr>
</thead>
</table>
| The evaluation | For system developers and architects:  
- Understand the state of the art of process modeling notations and techniques from academic perspective.  
- Provide criteria and corresponding analysis to choose suitable modeling framework to apply. |
| The approach | By applying the approach in industry, the system could provide manufacturers (MES customers):  
- More run-time information with improved understandability of operations.  
- Flexibility to change the manufacturing processes. |
| The prototype | For system architects:  
- Give the possibility to improve the changeability or reconfigurability of MES.  
- Give the possibility to improve the flexibility in development. |

Table 1.1: Outcome Artifacts and Contribution
1.5 Outline

This thesis is structured according to the research process introduced in Section 1.3. Section 2 analyzes domain problems from industry, and describes the problems as research questions. Section 3 documents the process of conducting the systematic literature review and its results. Section 4 proposes a three-view-based integration approach as a candidate solution, and explains how we implement the prototype. Section 5 reports on the feedback of interviews as a validation from industry. Section 6 analyzes the validity threats of this thesis. Section 7, as the last section, provide a conclusion and gives an outlook on possible future work.
Chapter 2

Problem Formulation

MES (Manufacturing Execution Systems) are process-oriented systems, that manage real-time behaviors of manufacturing plants and provide manufacturing information from both management and pure operation perspectives [39, 40]. The management parts of MES are normally applications and services in information systems, however, the operational parts are PLCs (Programmable Logic Controllers) that control the manufacturing processes within factories. MES solutions require the non-enterprise integration among the applications, services, and PLCs.

The warehouse management system, as one type of MES, is chosen to better understand the integration challenges of MES. The reason is that warehouse management systems cover all the elements that are relevant to process execution, including pre-defined processes at design-time, and distributed PLCs (or technical systems) within warehouses to be integrated at run-time. The decision of using it was taken under the discussion with our industry partner to ensure the thesis is in a manageable scope.

2.1 Warehouse Management System

A warehouse management system, facilitates the registration, planning, and controlling of warehouse processes. The operations of a warehouse include four major processes: goods-in, storage, order-picking, and shipping [6]. Figure 2.1 demonstrates these four processes inside a warehouse.

1) Goods-in: When new materials or goods arrive to warehouses, the warehouse management system starts a goods-in process. The activities include: register materials, determine the storage location, transport materials, etc.

2) Storage: It refers to the activities in the storage area. For example, put materials into racks, or take them from racks or some other special storage location for heavier and larger materials.
3) Order-picking: The system starts an order-picking process when a new picking order arrives. The involved activities are: find locations of materials, transport materials to workers, pick up materials, etc.

4) Shipping: This process starts after the order-picking process. When all materials of an order are ready, they should be sent out.

During the discussions with domain experts from industry, we identified that order-picking has the most variants, and consequently becomes the most complicated one among the four processes. A study from the United Kingdom also revealed that order-picking is the most costly in comparison with the other processes. More than 60% of all operating costs in a typical warehouse are closely correlated to order-picking [14]. For this reason, the order picking process is used to drive this research.

2.1.1 Warehouse Domain Model

In MES domain, ”Automatisches Kleinteilelager” (AKL, as the acronym of this German term) stands for automatic warehouse for small goods. Figure 2.2 presents the entities within an AKL.

A warehouse normally has a number of Racks. Each Rack contains some Storage Bins. A Transportation Unit (TU) holds a number of Materials as one group to be stored in a Storage Bin. Stacker Cranes and Conveyors perform the automatic transportation tasks. Stacker Cranes transport TUs from Racks to Conveyors. Then, Conveyors transport TUs between storage locations to the Workstation. At a Workstation, when workers receive Picking Orders, they pick Materials out from TUs to Picking Boxes; when workers receive Storage Orders, they put Materials into TUs to store into the warehouse. After the worker perform their tasks, Conveyors transport TUs back to storage locations, and Stacker Cranes transport TUs back to the Storage Bins on the Racks.
2.1. Warehouse Management System

2.1.2 A Warehouse Example

Warehouse management systems in Siemens can be considered as product families that hold similar features and functions. According to customer needs, each warehouse at customer side could have a different configuration. A concrete example of a running warehouse from industry is taken to help understanding the problems. It is illustrated in Figure 2.3.

![Figure 2.2: Warehouse Elements of an AKL](image)

This warehouse has six racks for storage, numbered from 1 to 6. On each rack, there are five storage bins. Moreover, for each storage bin, there is a tray (or “Tablar” in German) with 100 spaces (10×10). This tray becomes a TU during transportation, because all materials on one tray are transported together as one group. This warehouse has three stacker cranes and three workstations. For example, Stacker Crane 1 works for Rack 1 and Rack 2, moving TUs in between racks and the conveyor system. The conveyor system brings TUs to the
three workstations on demand.

In this particular warehouse, we assume that Rack 1 and 2 store packages of Coffee; Rack 3 and 4 store sugar; Rack 5 and 6 store milk. In this case, each TU could carry a maximum 100 packages of these materials to workstations.

In addition to the above warehouse configuration, we introduce a typical order-picking use case for analyzing the challenges, sketching the solutions, and prototyping. It is described as follows:

- **Title:** Order picking in AKL
- **Description:** The order requires 80 packages of coffee, 50 packages of sugar, and 200 packages of milk. (Each TU can contain 100 packages of each material.)
- **Pre-Condition:** A warehouse manager starts the execution of this order.
- **Post-Condition:** The goods are ready for shipping.
- **Main Scenario:**
  1. The system locates the storage of coffee, sugar and milk in the racks.
  2. Stacker Crane 1, 2, 3 bring 1 TU of coffee, 1 TU of sugar, and 2 TUs of milk to the Conveyor System, respectively.
  3. The Conveyor System brings coffee to Workstation 1; sugar to Workstation 2; and milk to Workstation 3.
  4. User tasks at workstations:
     4.1 At Workstation 1, Worker A picks 80 packages of coffee to a picking box.
     4.2 At Workstation 2, Worker B picks 50 packages of sugar to a picking box.
     4.3 At Workstation 3, Worker C picks 100 packages of milk to a picking box from the first TU; and from the second TU, Worker C picks 100 packages of milk.
  5. The Conveyor System brings TUs with the remaining materials back to Stacker Cranes.
  6. Each stacker crane moves the respective TU back to its original place on the racks.
  7. The system confirms that the order-picking process stops.

### 2.2 Limitations of the Existing Solution

Warehouse systems are already in the market and follow a component-based architecture. These components include: Storage Management, Material Management,
2.3 Research Questions

Forklift Management, PLC Controller, Quality Assurance Management, Production Monitoring, etc. However, based on a market-wide observation and feedback from customers, the following limitations of the existing system are identified.

The current warehouse management system has an almost hard-coded procedure, similar to a state-machine. This procedure is neither visible to customers, nor to business analysts. The feedback from customers implies that customers prefer to see this manufacturing procedure explicitly to understand the system’s operations. By having a visible procedure (or process) at runtime, they would be able to acquire more production information.

To further understand the variants of warehouse management systems, we realize that the manufacturers (the customers of MES) require a flexible manufacturing process for producing different products. The current hard-coded process is difficult to be changed both in design-time and run-time. In addition, the configuration of the manufacturing systems could be very different in the real world. For example, in the warehouse example in Section 2.2, there are 6 racks for storage, and 3 workstations, but another warehouse could have 20 racks and different types of workstations. When facing these configuration variants, the current solution requires, to a certain extent, code level copy-and-paste. This introduces extra cost to response to new requirements from customers.

To sum up, in this Master thesis, we expect to look for the possible improvement that helps MES vendors to support better understandability of the MES process to the customers, and that provides better reconfigurability and flexibility to MES processes and systems.

2.3 Research Questions

After having understood the limitation of the existing system, we further analyze the challenges to overcome the limitation of the existing system, and formulate research questions. Figure 2.4 presents an order-picking process and its execution environment at runtime. The identified challenges are marked as A, B, and C in this figure.

Challenge A – Process modeling techniques: In order to have an explicit process model in MES, it is necessary to know the state of the art of process modeling in academia and look for a suitable one to model MES processes.

Challenge B – Using only process models is not sufficient: The process model contains only the information of an order. It is important to have an additional model to represent the physical world elements, such as racks, conveyors, and workstations (in other words, the PLCs) in a warehouse.
Chapter 2. Problem Formulation

Figure 2.4: The MES Process and its Runtime Environment

Challenge C – Mapping between processes and its execution environment:
Some tasks in MES are performed at a certain location in the warehouse. For instance, in Figure 2.4, the “Pick” task is done by the worker manually at the workstation, and the “Transport Materials” task is done by the conveyor system. For process execution, both the information from the process model and information from physical world elements are needed. Thus, a linkage between these two worlds is necessary, in order to use the MES process to control the PLCs for execution.

From the challenges, the research questions are formulated in Table 2.1:

<table>
<thead>
<tr>
<th>RQ1.</th>
<th>What is the state of the art of modeling techniques in MES integration from academia perspective?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.1 What are the advantages and disadvantages of existing modeling techniques?</td>
</tr>
<tr>
<td></td>
<td>1.2 Which of the existing modeling techniques (models and views) is suitable to use to satisfy MES integration?</td>
</tr>
<tr>
<td>Rationale:</td>
<td>The answer to RQ1.2 will help us to understand and tackle challenge A and B.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RQ2.</th>
<th>What could be the improvement of the selected process modeling techniques to support process execution?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.1 What is the necessary information in the process model for execution in a real-world environment?</td>
</tr>
<tr>
<td></td>
<td>2.2 What could be an additional view or model applicable for representing the physical environment of MES?</td>
</tr>
<tr>
<td></td>
<td>2.3 How to bridge the process model and the additional view?</td>
</tr>
<tr>
<td>Rationale:</td>
<td>To answer RQ2.2, we plan to tailor or extend existing modeling techniques for challenge B. RQ2.3 is aiming at finding the possible integration between different models or views for Challenge C.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RQ3.</th>
<th>How does the proposed improvement overcome the limitations of the existing solutions?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.1 How does the improvement compare to the existing modeling techniques?</td>
</tr>
<tr>
<td></td>
<td>3.2 How does the approach of improvement solve the limitation of the existing warehouse solution?</td>
</tr>
<tr>
<td>Rationale:</td>
<td>RQ3 will be the validation step of the proposed approach.</td>
</tr>
</tbody>
</table>

Table 2.1: Research Questions
2.4 Research Methodologies

- **Systematic Literature Review (SLR).** For RQ1, a SLR has been conducted to understand the state of the art of process modeling techniques from an academic perspective. An evaluation has been performed to find the suitable techniques to use in MES integration. Thus, the result of the SLR provided the foundation to build up the possible solution.

- **Prototyping.** RQ2 aimed at improving existing techniques to tackle the challenges and overcome the limitations that have been identified in Section 2.2. In this case, we developed a prototype of our proposed approach as the proof-of-concept, to make sure that the proposed solution is practical and feasible to realize from an academic perspective.

- **Semi-structured Interviews.** To answer RQ3, semi-structured interviews, as a qualitative research method, have been planned and conducted with engineers in Siemens. During the interviews, both the idea of the candidate solution and the prototype were provided to domain experts. This was the industrial validation to receive practical and direct feedback regarding to the proposed solution.
Chapter 3

The State of the Art

According to [29, 30], a systematic literature review in software engineering has three major steps: planning, conducting, and reporting. Accordingly, we follow these three steps to conduct our review and structure this chapter. This literature review is a stepping stone to look for the possible improvement for solving the problems identified in industry.

We already have proposed our research questions in Section 2.3. This literature review is specifically targeting at RQ1. We first collect and summarize the existing process modeling techniques from electronic databases. Taking into account the requirements in MES integration, we conduct evaluation of these techniques as the answer to RQ1.1. Then, we choose the suitable techniques for process modeling and execution. This suitability analysis becomes the answer to RQ1.2.

3.1 Planning the Review

The review protocol has been designed following the strategies suggested in [51]. Table 3.1 presents the search terms that are used during the literature review. These terms are derived from the research questions, and they include alternative terms and synonyms as well. “OR” is used to incorporate alternative spelling and synonyms; “AND” is used for combining major terms.

As shown in the following list, five electronic databases are used in this literature review. Besides these five, Google Scholar is also involved for snowball searching, in order to find other relevant papers.

- IEEE Xplore
- ACM Digital Library
- Springer Link
- Engineering Village (Compedex & Inspec)
- Science Direct


3.1.1 Inclusion and Exclusion Criteria

As a part of the review protocol, Table 3.2 presents the inclusion and exclusion criteria of this literature review. These criteria are applied for the selection of primary studies. The fourth inclusion criterion indicates that we expect to include existing systematic literature reviews in manufacturing process modeling. However, during the execution, no systematic literature review could be found. Hence, we refine our inclusion criteria to add the fifth one, to include those papers that compare modeling techniques.

<table>
<thead>
<tr>
<th><strong>Inclusion Criteria</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The article is peer reviewed.</td>
</tr>
<tr>
<td>2 The article is written in English.</td>
</tr>
<tr>
<td>3 The article proposes or analyzes a modeling language, notation, method, view, technique, or framework for manufacturing processes.</td>
</tr>
<tr>
<td>4 The article gives an evaluation or review about different MES modeling techniques.</td>
</tr>
<tr>
<td>5 The article compares two or more modeling techniques.</td>
</tr>
<tr>
<td>6 The article provides the process model and its enactment and execution in MES.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Exclusion Criteria</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The article proposes an extension to a process model that narrows to a very specific manufacturing process.</td>
</tr>
</tbody>
</table>

Table 3.2: Inclusion and Exclusion Criteria

3.1.2 Quality Assessment

According to [29, 51], quality assessment criteria are important to analyze the selected studies and to perform data extraction. With the special MES process execution concerns, we define our quality assessment as shown in Table 3.3. These criteria work as further exclusion criteria to analyze the importance of primary
3.2 Conducting the Review

Additionally, these criteria provide further guidance for our interpretation of findings and recommendation to formulate our idea of improvement in the next research step [29, 30].

<table>
<thead>
<tr>
<th>Quality Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Is the paper proposing a well-defined model that is appropriate for MES processes?</td>
</tr>
<tr>
<td>2. Is the paper providing an approach that is an extension of other approaches?</td>
</tr>
<tr>
<td>3. Is the proposed modeling approach concerning process execution?</td>
</tr>
<tr>
<td>4. Is there any limitation or negative comments reported on the model?</td>
</tr>
</tbody>
</table>

Table 3.3: Quality Assessment

3.2 Conducting the Review

Figure 3.1 explains the execution of this literature review. With the search terms displayed in Table 3.1, 4985 results were retrieved from the selected electronic databases. (More details about the search terms can be found in Appendix A, Figure A.1.) By reading the title and abstract, 128 primary studies were selected according to the inclusion and exclusion criteria in Table 3.2. Furthermore, the duplicate studies were excluded from our selection. Considering the quality assessment criteria, we did not always select the most recent publications. Instead, we included studies that better satisfy our assessment criteria. Finally, we conducted a snowball search, resulting in one additional article included in the final collection. In total, 24 primary studies are found, including 19 process modeling papers and 5 review papers.

3.3 Reporting on the Review

One of our expectations when conducting this literature review was to find existing systematic literature review papers in MES process modeling, for choosing the suitable modeling techniques to tackle our integration challenges. By following the review protocols, we found five review papers that compare two or several modeling techniques, but none of them is systematic literature review. Among the selected five review papers, one of them makes the comparison between two modeling languages, and the other four review papers provide road-maps and evaluation more into the ERP layer in the automaton pyramid. Although they focus more on business process in the ERP layer, they are included as our primary studies, because they provide the criteria that help us to better analyze
Chapter 3. The State of the Art

Figure 3.1: Conducting the Literature Review

In this section, the result of this literature review is presented in four steps:

- **An overview of primary studies of MES process modeling will be given in the first place.**
- **The 19 selected studies are categorized, and analyzed.** This part gives the answer to RQ1, as the state of the art of current research from academia.
- **In order to evaluate the benefits and drawbacks of the existing models, we summarize the specific requirement of process modeling and execution in MES. Based on these requirements, we perform an evaluation and an assessment of our selected studies, as the answer to RQ1.1.**
- **After the evaluation, the suitability of modeling techniques is discussed to tackle the identified challenges.** This part targets RQ1.2, and therefore builds the basis for research question RQ2.2.

### 3.3.1 Overview of Primary Studies

The word “process” is defined in the dictionary as a series of actions, changes, or functions bringing about a result [36]. Considering the automation pyramid introduced in Chapter 1, the idea of modeling MES processes comes from business processes in the ERP layer, which is the top most layer in the automation pyramid. Hammer and Champy define “business processes” as “a set of activities that, together, produce a result of value to the customers” [22]. According to this
3.3. Reporting on the Review

definition, we define an “MES process” as a set of activities or operations, which produces a valuable result to manufacturers. The difference between activities and operations: activities are more human-related; operations are more automated. Based on this definition, modeling MES processes requires the notations that represent activities and operations within the processes. Executing the process models requires the models, manufacturing devices and machines to be integrated.

From our systematic literature review, it is obvious that almost all MES process modeling techniques were proposed originally for business processes. Figure 3.2 illustrates roughly the year that researchers propose to use these notations in business and manufacturing process modeling.

![Figure 3.2: Notations Proposed to Use in Business/Manufacturing Process Modeling](image)

The complete list of all selected studies is presented in Table 3.4. In Figure 3.2, we only present six modeling languages. The reason is that MES processes have quite different features and requirements, in comparison with business processes (We will analyze the differences in Section 3.3.3). We actually could find more studies about using hybrid models in understanding and analyzing MES processes. They suggest combining different modeling techniques to model different information. This combination is not easy to present in a two-dimensional diagram. This finding from literature help us to understand that using multiple views or models are needed for modeling MES processes.
<table>
<thead>
<tr>
<th>No.</th>
<th>Year</th>
<th>Title</th>
<th>Database</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1997</td>
<td>Generic modeling of manufacturing processes using Petri net representations</td>
<td>IEEE</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>1998</td>
<td>A formal modelling of control processes</td>
<td>Science Direct</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>A Model-based approach for Component Simulation Development</td>
<td>ACM</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>Integrating UML Diagrams for Production Control Systems</td>
<td>ACM</td>
<td>31</td>
</tr>
<tr>
<td>5</td>
<td>2002</td>
<td>Modelling of complex automation systems using colored state charts</td>
<td>Compendex</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>2003</td>
<td>Task-based modelling and configuration of assembly workstations</td>
<td>IEEE</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>2003</td>
<td>Evaluation of Modeling Notations for Basic Software Engineering in Process Control</td>
<td>IEEE</td>
<td>55</td>
</tr>
<tr>
<td>8</td>
<td>2003</td>
<td>Business-process modelling and simulation for manufacturing management – a practical way forward</td>
<td>Inspec</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2005</td>
<td>A Role-Based Framework for Business Process Modeling</td>
<td>IEEE</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>2006</td>
<td>Process modeling for simulation</td>
<td>Science Direct</td>
<td>41</td>
</tr>
<tr>
<td>11</td>
<td>2006</td>
<td>CPM: A collaborative process modeling for cooperative manufacturers</td>
<td>Science Direct</td>
<td>43</td>
</tr>
<tr>
<td>12</td>
<td>2006</td>
<td>A three-layered method for business processes discovery and its application in manufacturing industry</td>
<td>Science Direct</td>
<td>58</td>
</tr>
<tr>
<td>14</td>
<td>2008</td>
<td>Metamodelling Techniques Applied to the Design of Reconfigurable Control Applications</td>
<td>ACM</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>2008</td>
<td>Combined use of modeling techniques for the development of the conceptual model in simulation projects</td>
<td>ACM</td>
<td>37</td>
</tr>
<tr>
<td>16</td>
<td>2009</td>
<td>Enabling Flexible Manufacturing Systems by Using Level of Automation As Design Parameter</td>
<td>ACM</td>
<td>27</td>
</tr>
<tr>
<td>17</td>
<td>2009</td>
<td>Viewpoints in complex event processing: industrial experience report</td>
<td>ACM</td>
<td>28</td>
</tr>
<tr>
<td>18</td>
<td>2009</td>
<td>Research on Reconfigurability of Service-Oriented Manufacturing Execution System</td>
<td>IEEE</td>
<td>21</td>
</tr>
<tr>
<td>19</td>
<td>2010</td>
<td>Modeling of Manufacturing Execution Systems: an Interdisciplinary Challenge</td>
<td>IEEE</td>
<td>40</td>
</tr>
<tr>
<td>20</td>
<td>2010</td>
<td>3+1 SysML view model for IEC61499 Function Block control systems</td>
<td>Inspec</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>2010</td>
<td>A discussion of object-oriented process modeling approaches for discrete manufacturing on the examp...</td>
<td>IEEE</td>
<td>16</td>
</tr>
<tr>
<td>23</td>
<td>2010</td>
<td>Business process modeling languages: Sorting through the alphabet soup</td>
<td>ACM</td>
<td>36</td>
</tr>
<tr>
<td>24</td>
<td>2011</td>
<td>A SysML-Based Methodology for Manufacturing Machinery Modeling and Design</td>
<td>IEEE</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.4: Selected Studies
3.3.2 Modeling Techniques in MES

In business process modeling, Mili et al. categorize modeling techniques into four groups: traditional process modeling languages, object-oriented languages, dynamic process modeling languages, and process integration languages [36].

We tried to categorize the 19 selected studies (without counting the 5 review papers) into these four groups and realized that the latter two groups are not suitable to be used in MES modeling. We renamed the dynamic process modeling group to multiple views & hybrid techniques. The reason for renaming the dynamic process modeling group to multiple views & hybrid techniques is that in our selected papers, all the 11 papers that use dynamic modeling techniques also have approaches that propose hybrid modeling techniques or multiple views for different types of information within a process. As having different views is a crucial aspect in this thesis to enable MES process execution, the group has been renamed accordingly. We decided to remove the process integration group, because MES have a relatively homogeneous execution environment, normally at a local site. No process integration language for MES processes was found from the SLR. Besides the categorization suggested by Mili et al., we added one additional group named process decomposition, as there are three articles reduce the complexity of MES processes by decomposing them.

Therefore, the primary studies are categorized into these four groups: 1) traditional models, 2) object-oriented models, 3) process decomposition, and 4) multiple views & hybrid techniques.

1. Traditional Models: Modeling languages in this category include Petri nets [24], the IDEF family [3] as well as extensions and transformations of them. Some other models and notations were found, such as Markov-chain-based decision processes [7, 49] and event process chains [47]. After taking into account our exclusion criteria, these models were not included in our final selected studies.

- Petri nets: Murata defines a Petri net as a special kind of graph aimed at representing the behavior of dynamic systems [38]. It contains three constructs: tokens, places, and transitions. With these fairly simple constructs, a Petri net provides relatively powerful support as a formal modeling language that can be validated and executed [38, 24]. Horvath et al. propose using Petri nets to link production process, set-ups, operations, and tool sequences [24]. From our SLR, quite a lot of articles found in the SLR propose to use Petri-net-related techniques,
such as colored Petri nets or stochastic Petri nets for modeling MES processes [42, 50].

- IDEF family: IDEF refers to Integrated computer-aided manufacturing DEFINition. It was developed by United States Air Force to model software and information systems [35]. Up to now, IDEF family already contains a bunch of different types of models (IDEF0, IDEF1....IDEF14) for different purposes: functions, information, processes, user interface, etc. Specifically for MES, Benjamin et al. apply IDEF3 (process description model) and IDEF5 (ontology description model) for simulation modeling and analysis specification. Based on these two models, an executable component-based simulation model could be generated, which becomes reusable libraries to support rapid system re-configuration [3].

2. Object-Oriented Models: This category includes UML-related languages and their extensions.

- Köhler et al. propose an approach for production control systems, with several integrated UML diagrams. This approach integrates Specification and Description Language (SDL) block diagrams, state charts and collaboration diagrams to form an executable specification language to analyze the behavior of production systems. In combination with the SDL block diagram, UML class diagrams are used as well to generate executable Java code for process simulation [31].

- Fengler et al. use colored state charts as a solution to model complex automation systems. The state diagram describes the dynamic behavior of one or several objects with similar behaviors, and each object is distinguished by a different color. Transitions in the diagram change the state of objects. The benefit of using such a colored state chart is that it can be easily transformed into a colored Petri net for analysis or verification purposes [17].

- Most MES could be considered as discrete event systems, due to MES' nature. Ryan and Heavey propose simulation activity diagrams (SAD), aiming at providing models with good understandability to both technical experts and system users without technical background. SAD introduces an action list, which consists of actions that could trigger events to change the system states. These events are modeled as SAD primitives, which enable the interaction between the controlling system and the physical resources under control [41].
3. **Process Decomposition:** To reduce the complexity of modeling different kinds of information in MES processes, the decomposition of MES processes is another promising approach. From our SLR, three primary studies are categorized in this group [18, 10, 33]:

- **Agent-based decomposition:** Ferrarini et al. propose an agent-based architecture of manufacturing automation. Agents in such a system work autonomously, and communicate with each other to control the functions of the MES. Since agents are individual software components that control the PLCs to perform certain tasks, they are especially good to apply for distributed production environment [18]. With the concerns of non-enterprise integration in the automation pyramid, an agent-based architecture provides good support for hierarchical decomposition of MES processes to functions, as well as data or events acquisition in between MES and control layer.

- **Role-based decomposition:** Within a manufacturing process, actors with different roles might take part in the same process but in different activities. Caetano et al. provide their solution to model processes based on actors’ role and the objects with which the role should interact [10]. It is possible to decompose MES systems in this way, as it helps to analyze the process from users’ perspectives.

- **Task-based decomposition:** Recurring to the definition of manufacturing process that we defined at the beginning of this section, a process consists of a set of human or machinery activities. The idea of task-based decomposition naturally breaks down MES processes in this way. In [33], activities are linked to actions in a pre-identified action list. One additional workstation configuration model describes the physical equipments of workstations. Finally, the linkage among the process, actions and workstations becomes an overview model. By using task-based decomposition, users and system designers are able to better map functional requirements to system functions with such an action list, and then, further link the list to the physical configuration.

4. **Multiple Views & Hybrid Techniques:** There are 11 primary studies in total in this category. Various modeling languages and notations are being applied. Besides the traditional models and object-oriented models, more dynamic modeling languages are proposed and used by researchers, such as Business Process Modeling Notation (BPMN), Business Process Execution Language (BPEL), and Systems Modeling Language (SysML). Before we go
on to analyze the combination of them in our primary studies, we introduce these three dynamic languages in the following paragraphs.

BPMN is designed by the Business Process Management Initiative (BPMI) work group. The first version of BPMN 1.0 specification was released in 2004. The original goal was to create a process modeling language that is understandable for business users, such as business analysts or administrators [56]. Currently, the latest version of BPMN is 2.0. BPMN contains four categories of elements. Flow objects provide the notations of business events, activities, and gateways. Connecting objects provide the representation of connection mechanisms, such as the sequence flow, the message flow, and associations. The swimlanes have two sub-elements: a pool is used to separate the activities of different participants within one process, whereas a lane separates activities of organizational functions and roles. Artifacts in BPMN contain different notations of data objects, element groups, and annotations [56, 36].

BPEL is an executable language for orchestration of web services in a business process [12]. It was known as BPEL4WS. BPEL provides very precise semantics to support process execution. Some BPMN engines support informal transformation and mapping between BPMN and BPEL, which provides the chance to implement and execute BPMN processes, but this transformation is not standardized [11, 36]. One limitation of using BPEL in MES integration is that, BPEL supports mainly horizontal integration in the sense of orchestrating web services. However, MES demand vertical integration, and generally require big effort to adapt to web service protocols, for example, by adding wrappers for existing MES applications.

SysML is developed by the Object Management Group (OMG). It is an extension of UML, and is adapted for system engineering [19]. In comparison to UML, SysML simplifies some UML diagrams. For example, it uses blocks as units to do system modeling. SysML also provides the requirement modeling concepts [19]. Since there is a widespread usage of UML, SysML as the extension of UML can provide usability and understandability for developers in the area of dynamic activity modeling.

After having introduced these three dynamic modeling languages, Table 3.5 summarizes the 11 primary studies, and how they combine traditional models, object-oriented models, and the dynamical models. From this table, it is easy to see the trend of using multiple views and hybrid techniques in MES process modeling, which is consistent with the challenges we identified in our problem formulation section. The researchers in this area have
already proposed various combinations of models and techniques.

<table>
<thead>
<tr>
<th>No.</th>
<th>Year</th>
<th>Ref.</th>
<th>Views, Layers with Techniques</th>
</tr>
</thead>
</table>
| 1   | 1998 | [13] | - Operational process layer: [54]  
|- Control process layer: [54] |
| 2   | 2006 | [43] | UML extension, IDEF, Petri nets  
|     |      |      | - Component layer  
|     |      |      | - Operation layer: UML, BPEL  
|     |      |      | - Operation Integration layer |
| 3   | 2006 | [58] | - Component layer  
|     |      |      | - Operation layer: UML, BPEL  
|     |      |      | - Operation Integration layer |
| 4   | 2008 | [37] | SIPOC\textsuperscript{1}, Flowchart, IDEF0  
|     |      |      | - Component layer  
|     |      |      | - Operation layer: UML, BPEL  
|     |      |      | - Operation Integration layer |
| 5   | 2009 | [27] | Layout, Part Information, Support, Resource Information, Production Operations, Production Planning (discrete event-driven, UML)  
|     |      |      | - Function view: BPMN  
|     |      |      | - Data view  
|     |      |      | - Process view: BPMN, BPEL |
| 6   | 2009 | [28] | - Contextual Layer: BMM\textsuperscript{2}  
|     |      |      | - Conceptual Layer: BPMM, BEMN\textsuperscript{3} (event-driven)  
|     |      |      | - Logical Layer: UML  
|     |      |      | - Physical Layer: RAPIDE [34]  
|     |      |      | - Component Layer |
| 7   | 2009 | [21] | - Function view: BPMN  
|     |      |      | - Data view  
|     |      |      | - Process view: BPMN, BPEL |
| 8   | 2010 | [40] | BPMN extension:  
|     |      |      | - MES Functional Model  
|     |      |      | - Production Process Model  
|     |      |      | - Technical System Model |
| 9   | 2010 | [52] | - MTS View: SysML  
|     |      |      | - Mechanical engineer view: SysML  
|     |      |      | - Electronics engineer view: SysML  
|     |      |      | - Software engineer view: IEC61499\textsuperscript{4} |
|     |      |      | - Execute process: BPEL |
|     |      |      | - Global layer: SysML  
|     |      |      | - High-level layer: SysML |

\textbf{Table 3.5:} Summary of Papers Applying Multiple Views and Hybrid Techniques

\textsuperscript{1}SIPOC stands for Supplier, Input, Process, Output, and Customer. The developed tool provides a high-level model for the development team to identify relevant elements of a process improvement project to start working.

\textsuperscript{2}BMM refers to Business Motivation Model specification. It is developed by OMG. It focuses on business objectives and goals, in order to analyze the impact of business models [28].

\textsuperscript{3}BEMN is a BPMN extension proposed in [11]. It is a graphical language for modeling composite events in business processes.

\textsuperscript{4}International Electrotechnical Commission (IEC) develops a wide range of standards in the area of electric, electronic and automation. In the context of this thesis, it was not possible to get access to IEC standards.
During the snowball searching phase, there are some other modeling techniques have been found. However, some of them are industrial standards that would need to be purchased, and some others are not available in English. Therefore, they are not included in this research, but we list them as follows: IEC family (IEC 61131, IEC-PAS 62424, IEC 61499), ARIS, ISO 3511, VDI/VDE 3682.

### 3.3.3 Evaluation

With regard to the research questions in Section 2.3, this section provides the answer to RQ1.1: What are the advantages and disadvantages of existing modeling techniques. Taking into account one of the identified challenges that one or several additional models is needed besides the process model, this evaluation only considers the modeling techniques in the *Multiple Views & Hybrid Techniques* group.

According to [36, 8], there are in general three aspects of process modeling. **Describe process:** The model should be conceptual that provides the representation of the functions of the system. **Analyze process:** The model should help system designers to understand and analyze the existing systems, for example, for process re-engineering purposes. **Execute process:** This refers to the implementation of the process model. For this purpose, models should support simulation, executing, or enactment to a certain satisfaction level. In consideration of these three aspects, we suggest the following criteria to conduct the evaluation in this section.

#### Describe process

- **Understandability:** The models should be easy to understand and use by users with and without technical background. The technical users of the models could be designers and developers of warehouse management systems. They use the models during design time to build and implement manufacturing systems. The non-technical users of the models could be business analysts, administrators, or managers of MES. They use the models mainly during runtime to understand and control the behaviors of MES. Especially for non-technical users, this understandability becomes very important, because they are the end users and customers, who frequently work with the models in run-time.

- **Representation power:** The models should be sufficient to present and express the MES process. In [53], important modeling elements in process modeling are identified as work flow patterns. The most relevant
3.3. Reporting on the Review

patterns regarding process execution in MES are, for example, *synchronizing merge* and *multiple instances*.

**Analyze process**

- **Support for real-world view(s):** The models should have the capability to present the elements that are being manipulated by the process [36]. Examples of such elements in typical warehouses include: conveyors, stacker cranes, racks, and materials.

- **Support for event analysis:** For non-enterprise system integration, processes can be considered as the interaction of discrete, asynchronous, and concurrent events with the concern of resource allocation [5]. This is because the communication between the MES layer and the Control layer is driven by events. Hence, for modeling MES, it is necessary to have discrete event concerns. The modeling language should provide support to analyze the exchanged events between these two layers.

**Execute process**

Taking into account Börger’s suggestion for process modeling [8], we consider the following criteria are important for process execution:

- **Support for faithful implementation:** The models should support systematic, controlled refinement. This requires the models to have relatively precise semantics, so that they can support faithful implementation.

- **Support for effective management:** The models or the modeling techniques should provide the possibility of monitoring and evaluating the system, because during process execution, the warehouse manager or administrator needs to know the status of the running processes, and manage them. For example, in case the workstation is broken when order picking process is already started, the warehouse manager should be informed to stop the process, and start a new process assigned to another workstation.

- **Coherence of different views:** The models work as the abstraction of different parts of the system. It should be possible to integrate the models for execution (e.g., the control-flow view and the resource-flow view).

- **Support for integrability:** The models or the modeling approach should provide the possibility to communicate and inter-operate with technical systems in real manufacturing plants. (e.g., the transformation of exchanged information.)
Based on these criteria, Table 3.6 presents a comparison of the 11 modeling techniques. ‘+’ means positive support. ‘-’ means lacking of support. ‘±’ means limited support.

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Criteria</th>
<th>[13]</th>
<th>[43]</th>
<th>[58]</th>
<th>[37]</th>
<th>[27]</th>
<th>[28]</th>
<th>[21]</th>
<th>[40]</th>
<th>[52]</th>
<th>[15]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe</td>
<td>Understandability</td>
<td>±</td>
<td>+</td>
<td>±</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Representation power</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Analyse</td>
<td>Support for real-world views</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>Support for event analysis</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Execute</td>
<td>Support for faithful impl.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>±</td>
<td>±</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Support for effective management</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Coherence of different views</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Support for integrability</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>±</td>
<td>±</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.6: Existing Models with MES Modeling Criteria

The five evaluation papers [32, 40, 8, 55, 2] from our literature review have already done comprehensive evaluation work, especially regarding the representation power (in the describing aspect) of the individual modeling languages. On one hand, Table 3.6 shows that the existing techniques provides fairly good support to fulfill the describing aspect of process modeling. Since we choose only the techniques support multiple views, most of them also support the first criterion of the analyzing aspect.

On the other hand, Table 3.6 clearly shows the lack of consideration of the last five criteria: concerns of event analysis, faithful implementation, effective management, coherence for different views, and integrability, especially the criteria from the “execute” aspect.

3.3.4 Suitability and Applicability

Based on the result of the evaluation, this section answers RQ1.2: Which of the existing modeling technique is suitable to use for integration between MES layer and control layer. By checking Table 3.6 carefully, we see that, SpeziMES [40] is the one that satisfies our criteria the best among the others.

The strength of [40] in describing and analyzing MES processes stems from the fact that it uses BPMN and BPMN extensions to model MES processes and
3.3. Reporting on the Review

events. The benefit of BPMN is clear that it provides good understandability to non-technical users or stakeholders. BPMN is relatively easy to learn by technical engineers or developers, so that they can model the behaviors of MES. SpeziMES tailors and extends the existing BPMN notations of events and tasks to the needs of MES.

SpeziMES proposes a graphical MES modeling approach. It comprises a technical system model, a production process model and a MES functional model, as shown in Figure 3.3 [40]. The technical system model represents the static technical systems that perform MES functions. This model could either be on the abstraction level of an entire plant, or detailed to atomic function units depending on needs. The second model is the production process model, which represents the production processes in MES. The authors propose to use UML activity diagram or flow chart as the modeling notation here. Lastly, the MES functional model represents the MES function processes. The author propose to use two different swim lanes in BPMN for MES functional model and technical system model.

In Figure 3.4\(^1\), we apply SpeziMES to model the order picking process introduced in Section 2.1.2. SpeziMES separates the production process model and the technical system model in different swim lanes. When a task in the MES function process is performed in a technical system, both the task and the technical system are marked in grey, and linked with a message flow notation. As illustrated in Figure 3.4, the ‘Transport TU’ task and the ‘Return TU’ task are marked in grey, and they are connected to the ‘Transporting Process’ in the technical system model. Here, the transporting process represents the function of transporting TU between storage and workstations.

As mentioned in the research question section (Section 2.3), for process execution, it is necessary to have a model that represents real-world elements. We

\(^1\)As the modeling tool that the authors of [40] used is not available. We create this SpeziMES model by using an on-line Business Process Modeling tool (http://www.gliffy.com/uses/business-process-modeling-software/).
consider the technical system model in SpeziMES as promising for this purpose. For proper representation, it is enough to use a message flow notation to connect tasks (in the MES function process) and the related technical system.

However, if we check the technical system model in Figure 3.4 with regard to process execution concerns, we realize that the linkage between tasks and its technical systems becomes ambiguous. In the order picking use case from Section 2.1.2, the system splits up one order into 4 picking sub-processes running in parallel: 1) picking 80 packages of coffee, 2) picking 50 packages of sugar, 3) picking 100 packages of milk, 4) picking 100 packages of milk. These four processes are executed at three workstations in the warehouse. To execute these 4 processes in run-time, the ‘Transport TU’ task, in each of these processes, needs to communicate with different transporting processes involving different stacker cranes, conveyors, and workstations. In this case, a single message flow in SpeziMES becomes insufficient to represent this many-to-many relationship.

3.4 Summary

In conclusion, from our selected primary studies, we found that SpeziMES, as a MES modeling framework, efficiently helps the stakeholders to understand MES functions, and reduces the difficulty of understanding MES processes and plant operations. According to our evaluation criteria, SpeziMES seems to be the most applicable approach to be applied. However, a detailed analysis in this section has revealed the limitations of directly using SpeziMES for process execution, considering realistic execution scenarios, a many-to-many relationship from the MES tasks to its technical systems should be addressed. In the next chapter, we plan to develop an improved approach for MES process execution.
Chapter 4

A Three-View-Based Integration Approach

Chapter 3 describes how the systematic literature review (SLR) was conducted. The evaluation of the primary studies has shown that most of existing modeling techniques have relatively good support for describing and analyzing purposes of MES. The evaluation has exposed the lack of emphasis on process execution. Moreover, the evaluation also revealed that SpeziMES, compared to the other proposed techniques, satisfies most integration criteria, albeit with some limitations in representing technical systems for executing processes.

In this chapter, we plan to propose our process modeling approach to overcome the limitations to achieve vertical integration. This will be the answer to our research question 2. We revisit our RQ2 and sub-questions as follows:

- **RQ2. What could be the improvement of the selected process modeling techniques to support process execution?**
  - 2.1 What are the necessary information in the process model and its execution in a real-world environment?
  - 2.2 What could be an additional view or model suitable for representing the physical environment of MES?
  - 2.3 How to bridge the process model and the additional view?

The answer to RQ2 is based on what we learned from our SLR. For RQ2.1, we expect to understand what are the data or information that should be modeled. For RQ2.2, we expect to establish our real-world model for representing MES, which possibly could be considered as an extension of the technical system model in SpeziMES. For RQ2.3, we plan to provide the solution to connect the process model to the real-world model. The answers to these three sub-questions are given in Section 4.1.1, Section 4.1.2, Section 4.1.3 respectively.
Chapter 4. A Three-View-Based Integration Approach

4.1 Solution Outline

4.1.1 Necessary Information

Learning from our SLR, besides SpeziMES, there are some other modeling techniques that provide valuable insights regarding MES integration. In [28], Kellner and Fiege propose to use different modeling notations for complex events processing in manufacturing and automation systems. They follow Zachman’s framework of enterprise architecture to analyze different dimensions and perspectives of MES [59]. Figure 4.1 presents Zachman’s framework. It provides six abstractions (as the columns) and six perspectives for different stakeholders (as the rows) of an enterprise system [59]. The better these views are integrated, the better the enterprise integration can be achieved.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>List of things important in the business</td>
<td>List of core business processes</td>
<td>List of business location</td>
<td>List of important organizations</td>
<td>List of events</td>
<td>List of business goals and strategies</td>
<td></td>
</tr>
<tr>
<td>Enterprise Model (Owner)</td>
<td>Conceptual data/object model</td>
<td>Business process model (BPIM)</td>
<td>Business logistics system</td>
<td>Work flow model</td>
<td>Master schedule</td>
<td>Business plan</td>
</tr>
<tr>
<td>System Model (Designer)</td>
<td>Logical data model</td>
<td>System architecture model</td>
<td>Distributed system architecture</td>
<td>Human interface architecture</td>
<td>Processing structure</td>
<td>Business rule model</td>
</tr>
<tr>
<td>Technology Model (Builder)</td>
<td>Physical data/class model</td>
<td>Technology design model</td>
<td>Technology architecture</td>
<td>Presentation architecture</td>
<td>Control structure</td>
<td>Rule design</td>
</tr>
<tr>
<td>Detailed Representation (sub-contractor)</td>
<td>Data definition</td>
<td>Program</td>
<td>Network architecture</td>
<td>Security architecture</td>
<td>Timing definition</td>
<td>Rule Specification</td>
</tr>
<tr>
<td>Functioning Enterprise</td>
<td>Usable data</td>
<td>Working Function</td>
<td>Usable network</td>
<td>Functioning organization</td>
<td>Implemented schedule</td>
<td>Working strategy</td>
</tr>
</tbody>
</table>

Figure 4.1: Enterprise Architecture by Zachman [59]

Zachman’s framework provides a very comprehensive way to analyze enterprise architecture. There are in total 6×6 cells, and each cell points out an abstraction, which potentially could become a model for stakeholders. We realize that having so many abstractions and perspectives is the reason why we found so many different models, views, and layers proposed in our SLR.

Narrowing down the focus to non-enterprise system integration, we try to analyze the necessary information for MES process modeling and execution with Zachman’s framework. Firstly, we examine the abstractions in the columns:

- The Data (What) column refers to the information related to the order.
  
  For example, in our order picking use case, the data include the required
4.1. Solution Outline

materials, the amount of materials, where the materials are stored in the warehouse, etc.

- The *Process (How)* column describes the steps or tasks required to achieve certain results in manufacturing. The simplest case could be a sequential process, but some scenarios might require a loop or multiple processes running in parallel. For us, the process clearly constitutes the main scenario in the order picking use case.

- The *Network (Where)* column refers to where the tasks are executed and performed within the system. In MES processes, this network column is important, because some tasks can only be performed at a specific location. A typical example is that workers do picking tasks only at the workstation. This is different from the tasks in business processes, where most of the tasks are related to documentation and databases. There is less restriction to the location to update the documents in business processes, as long as people can access the documentation or databases with computers or any web-based technologies.

- The *People (Who)* column refers to the actors who perform concrete tasks. In business process management, this perspective normally is managed by an organizational view (sub-system), in which roles and authorities are defined and controlled. However, in MES, workers normally have direct interaction with machines and devices. They perform more physical tasks compared to tasks, related to business processes, such as updating documentation. Hence, the systems for performing the tasks are usually less restrict for authentication and authorization.

- The *Time (When)* column is about the timing to start a task. For automated manufacturing, for instance, MES tasks are triggered by events. For this reason, we have also included “support for event analysis” as one important criterion in our evaluation. Considering that event processing is the basic communication mechanism in MES [28] has its specific concerns in this perspective that can be related to this column.

- The *Motivation (Why)* column relates to the strategic goals of organizations. We consider this column as important. But it will probably be addressed in some other requirement model and linked to MES, so we do not address such information in this research.

After having analyzed MES with the columns in Zachman’s framework, we examine the perspectives as the rows in this framework. Zachman identifies these
Chapter 4. A Three-View-Based Integration Approach

perspectives based on the stakeholders’ views over the process of engineering and complex manufacturing products [59].

The objective perspective (Row 1) is on a very high level, representing the general scope or context of business. The owner perspective (Row 2) belongs to the customers (in this Master thesis, customers are manufacturers) and users of the end products. The designer perspective (Row 3) provides the view to engineers, architects, etc. In this row, models work as the intermediary between what is desirable (Row 2) and what is physically and technically possible (Row 4) [59]. Row 4 is for the builders of the end products. They usually know the exact technical capacity and constraints to produce the end system$^1$. Row 5 is defined as out-of-context perspective, which provides a very detailed description about the media of end products$^2$. According to [59], row 6 is not a model or architecture any more; it is the real end product.

In this Master thesis, the most relevant perspectives are Row 2 and Row 3, because the most important stakeholders are the users, the engineers, and the architects of MES (See outcomes and contributions in Section 1.4). For users, their concerns are mostly run-time information. For engineers and architects, both run-time and design-time information are interested. Table 4.1 presents our analysis on the necessary information that enables process execution. The columns follow the five abstractions that are important in MES integration, and the rows concern run-time and design-time. Within each cell, we analyze what kinds of information should be modeled. Since the motivation is normally addressed with external tools, it is not included in this table.

<table>
<thead>
<tr>
<th>Run-time</th>
<th>Process</th>
<th>Data</th>
<th>People</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MES process</td>
<td>Order materials</td>
<td>Manager, worker</td>
<td>MES function systems</td>
<td>Timing</td>
</tr>
<tr>
<td>Design-time</td>
<td>System functions</td>
<td>Entities, data model in DB</td>
<td>People who do manual tasks</td>
<td>Distributed system structure</td>
<td>Event processing</td>
</tr>
</tbody>
</table>

Table 4.1: Necessary Information for Modeling and Executing MES Processes

A further examination of this table reveals that the Data and People columns have relatively lower priorities for having separate models in process execution. The Data column addresses the information of the order and materials. This type of information is normally stored in databases, and it requires database engineers

$^1$An example of this technical constraints in MES could be that the highest speed of a conveyor system for transporting goods is 0.4m/s.

$^2$An example given in [59] about the media of end products is that for producing an airplane, the needed materials could be aluminum, titanium, composites, etc.
to create proper data models to access, update, and retrieve. The People column refers to MES workers and managers. Normally, workers have direct interaction with their workstations. They are already authorized in some way, if they are able to physically access the workstations. For managers, they create orders and start the order picking process, but normally they do not directly participate in MES tasks during run-time. When an error occurs, the system might need to authenticate the users to cancel or stop a process. But in the context of this research, dealing with this exception case is not in our focus.

The necessity for modeling the information in Time (when) and Location (where) columns is relatively high, because the process model needs to control the functions of technical systems (PLCs) to perform manufacturing tasks. For example, in a warehouse, the process model informs the conveyor about when to start transporting materials, at which workstation the materials should arrive, when to return the remaining materials to racks. Knowing the importance of these information, Table 4.2 analyzes the modeling techniques from our SLR, to understand what information is already addressed in existing researches, and what are still missing. This table targets to BPMN and its extensions, because BPMN related techniques have good ranks in our evaluation.

<table>
<thead>
<tr>
<th>Run-time</th>
<th>Process</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design-time</td>
<td>BPMN activities [21, 40, 15]</td>
<td></td>
<td>Event processing network (EPN) model [28, 45, 46]</td>
</tr>
</tbody>
</table>

Table 4.2: Missing Information When Using Existing Models from SLR

From Table 4.2, it is obvious that there has been considerable amount of research on using and extending BPMN to model these three information abstractions. By using BPMN and its extensions, we could address four types of information in Table 4.2. But, there are two exceptions.

For the cell in Design-time row and Time column, we need an overall model of the event processing network (EPN). One selected study in our SLR ([28]) mentions a specification named “event metamodel and profile” for modeling EPN in this perspective. Event Metamodel and Profile (EMP) is proposed by the Object Management Group (OMG), and it is still in RFP\(^1\) status. Besides EMP, we could find some other articles in modeling EPN. For example, Sharon and Etzion propose a conceptual model of EPN for expressing the event-based interactions...
and event processing specifications among components [45, 46].

For the cell in Design-time row and Location column, a model to describe the network about how the technical systems (PLCs) are distributed is still missing. This model should be able to present how PLCs are physically connected in manufacturing plants. However, such model requires domain-specific knowledge, because the technical systems for producing different products could be very specific to a certain domain; and even for producing the same type of products, manufacturers may require different system configurations.

To sum up, the answer to RQ2.1, what is the necessary information required for actual process model execution, has been provided by Table 4.1. A further analysis presents that not all the information in Table 4.1 require a separated model for process execution. Table 4.2 exposes the important models that should be established.

We reflect the finding to vertical integration in Figure 4.2. In the MES layer, BPMN and its extensions support the information abstractions for both design-time and run-time. For the communication between the MES and the control layer, we could also find the models of EPN to express the interaction. In the control layer, the technical system model is used to model functions of PLCs. With the technical system model, we are able to model what a manufacturing plant can do (e.g., moving materials to a workstation). But in a real-world manufacturing plant, multiple manufacturing systems perform the same function (e.g., 3 stacker cranes move materials from 6 racks to the conveyor). Thus, one additional model to describe the structure of how the technical systems are physically distributed in the plant becomes important.

\[\text{Figure 4.2: Necessary Information to Model Vertical Integration}\]

\(^{1}\text{RFP stands for request for proposal.}\)
4.1.2 The Additional View

After having understood that the missing information is the structure of distributed technical systems in the manufacturing plant, in this section we propose an additional view to compensate this absent information. We name it as “abstract plant view”. To create this abstract plant view of technical systems, we firstly need to understand what elements should be included in this view.

Object-oriented modeling techniques, particularly UML class diagrams, are widely applied for mapping types of entities to classes. For describing and analyzing purposes, the UML class diagrams seem a natural option to use. With a class diagram, it is easy to establish a one-to-one link from the real-world entity to a class with the attributes and operations. In Figure 2.2 (Section 2.1.1), we already created a domain model to represent the elements within a warehouse. Is this domain model good enough to help us to establish the abstract plant view? Our answer to this question is “no”.

Isoda differentiates two kinds of object-oriented modeling: genuine real-world modeling and pseudo real-world modeling [26]. According to Isoda, genuine modeling is used for representing and simulating the real-world entities. Our domain model shown in Figure 2.2 is a genuine real-world model. However, for implementation purposes, we need the pseudo real-world modeling, by which we model the information understood by the software application [26]. Figure 4.3 shows the pseudo real-world modeling for the order picking process execution.

![Figure 4.3: Meta Model of Abstract Plant View](image)

Comparing Figure 4.3 with Figure 2.2, Material is removed and two entities are added. For a software system, the materials are identified by: the type id, the required amount, the stored location, etc. When the process starts, the data related to the Material entity are already included in the Order.
The first new included entity is *Pick*. Normally, one picking *Order* requires several different types of materials. This would lead to several picking tasks for executing the process. Each of the picking tasks viewed by software system is a *Pick*. The second new entity is *Buffer*. When conveyor brings TU to *WorkStation* and the worker is still performing the previous picking task, *Buffer* buffers the TU and waits till workers could finish the previous *Pick*. *Buffer* usually has a fixed size (e.g., size=5), which means that one workstation could buffer maximum 5 TUs. Exceeding this size, *WorkStation* would reject the transportation. In genuine real-world modeling, we did not consider *Buffer* as an individual entity, because it was considered as a part of *WorkStation*.

Figure 4.3 marks the types of entities that should be considered at design-time in the blue rectangle, which becomes the meta model of the abstract plant view of warehouse management systems. These elements in the blue rectangle are long-living objects in a warehousing plant, no matter whether the order picking process exists or not. The entities outside the rectangle are run-time entities. The software system creates them to start an order picking. During process execution, the software system may need to update their status.

Finally, we come up with the meta model of an abstract plant model on the left side of Figure 4.4. It contains four top elements: Rack, Stacker Crane, Conveyor, and Workstation; and two sub-elements: Storage Bin contained by Rack, and Buffer belonging to Workstation. Based on this understanding, we sketch the instance of an abstract plant view on the right side of Figure 4.4.

![Figure 4.4: Sample Model of Abstract Plant View](image)

**Summary and Contribution**

To sum up, in this section, we categorized warehouse entities into design-time and run-time groups. Based on the entities in the design-time group, we established...
the meta model for warehousing plant, then further instantiated this meta model to represent our warehouse example as the abstract plant view.

The major contribution of this section is not the meta model of the abstract plant view, neither the instance of the warehousing plant. Instead, the method to look for the meta model of the abstract plant view is what we would like to emphasize:

1. Establish the pseudo real-world model for MES.
2. Choose long-living objects in the pseudo real-world model as the meta model of manufacturing plants.
3. Instantiate the meta model of step 2 to create the abstract plant view for an individual plant.

From our analysis, it is very important to differentiate long-living objects, and run-time objects in MES for process execution. We consider both the abstract plant view, and this method to establish the abstract plant view, as the answer to RQ2.2.

4.1.3 Bridging the Process View and the Additional View

During process execution, the process instance should be able to communicate with PLCs in the plant, and control them to manufacture products. Based on the knowledge we have learned till now, modeling MES processes both for design-time and run-time can be solved by BPMN and its extensions. We also have modeled the technical systems (PLCs) in the real-world manufacturing plant as the abstract plant view. In this section, we are going to address how we could bridge these two models. By doing so, we actually realize the vertical integration between the MES layer and the Control layer. The first step for bridging these two layers is to understand how they communicate with each other. Figure 4.5 illustrates the communication in industry context. When the MES process requires a PLC to perform a task, it sends a command to the PLC, indicating what should be done. After the PLC finishes the task, it sends an event back to the MES process, so that the process can continue to do the next task. Based on our understanding in industry, typical protocols to enable this event exchange include: TCP/IP, UDP, etc.

Focusing on the order picking process, Figure 4.6 demonstrates how ‘Transport TU’ task could control the PLCs (the stacker cranes and the conveyor) to bring materials\(^1\) to the workstations.

\(^1\)Materials are transported within a TU. TU stands for Transportation Unit. As we already introduced in Section 2, a TU is normally a tray, that contains certain amount of materials. The
Figure 4.5: Communication Between MES and PLCs

1. ‘Transport TU’ asks Stacker Crane 1 to bring a ‘Transport TU’ from a storage bin in Rack 1, to the Conveyor. Similar events might be sent to Stacker Crane 2 and Stacker Crane 3 for bringing different materials.

2. After the stacker cranes finish the bring task, they send an event to ‘Transport TU’ to indicate that the materials are brought.

3. Then, ‘Transport TU’ sends a command to the Conveyor to bring the materials to a target workstation. For example, the materials from Rack 1 should be transported to Workstation 1.

4. After the Conveyor finishes the bringing task, it sends an event to the running picking process to notify that materials are brought to the target workstation.

In order to receive the command events from the MES process PLCs in manufacturing plants should provide some ports. In turn, MES process also should provide some ports to receive the events from PLCs. By doing so, we are able to establish a virtual mapping in between the process model and the manufacturing plant to route the exchanged events. In the abstract plant view, each function of every technical system should have a port to receive the command. The typical tray is not only used for storage. During the transportation in a warehouse, this tray becomes a TU, and it moves the materials inside as one group.
functions of technical systems include heating, moving, cutting materials. In the MES process, each task that requires PLC should have a port to receive the event that notifies the PLC has done the work. Depending on what kinds of protocols or technology in real industry settings, this port can be a message queue, a socket, or an interface for remote process calls. With regard to the EPN for modeling the Time abstraction (see Figure 4.2), we see that this mapping exactly becomes the model of the EPN that connects the MES processes and the manufacturing plant, and routes the events between them.

Figure 4.7 sketches this mapping view between the MES process view and the abstract plant view. In this mapping, Storage Locations do not involve into any event processing, because they are physical objects within a warehouse. Instead, they have the virtual linkage to MES databases. Except Storage Locations, all the other three groups of ports enable the event processing between a running MES process and PLCs in manufacturing plants. As a result, this mapping provides one additional support for representing the Time abstraction. By having this mapping, we effectively integrate the three important abstractions in process execution: the MES process view (How), the mapping view (When), and the abstract plant view (Where), which becomes a three-view-based approach to realize process execution in MES. Table 4.3 provides more detailed description of the port elements in this mapping view.

![Three-View-Based Integration Approach](image)

**Figure 4.7:** The Three-View-Based Integration Approach

In this section, we have proposed one mapping view between the MES process view and the abstract plant view. This mapping view becomes our answer to RQ2.3. We come up with this three-view-based integration approach to satisfy
Chapter 4. A Three-View-Based Integration Approach

<table>
<thead>
<tr>
<th>Storage Locations</th>
<th>Virtually map the position information of racks and storage bins to databases. The data model in databases depends on this information for material storage.</th>
</tr>
</thead>
<tbody>
<tr>
<td>StackerCrane ports</td>
<td>Bridge the Request TU task and Return TU task to two functions: bringing and returning TU (between racks and the conveyor).</td>
</tr>
<tr>
<td>Conveyor ports</td>
<td>Bridge the Request TU task and Return TU task to two functions: bringing and returning TU (between the storage area and workstations).</td>
</tr>
<tr>
<td>Workstation ports</td>
<td>Map to workstation functions: reserving a place in the buffer for the arriving TU, or releasing buffer. Here, each workstation possibly has one port for both reserving and canceling, because these two functions are logical services. They don’t require PLCs work.</td>
</tr>
<tr>
<td>Some other ports</td>
<td>In order to receive events from PLCs, process model provides some ports for each type of technical systems. But the mapping is not strictly one port per function, because the process engine or routing mechanism could be intelligent enough to forward or broadcast the information. For logging purpose, for example, important events should be sent to a centralized logging service.</td>
</tr>
</tbody>
</table>

*Table 4.3: Elements of the Mapping View*

the process modeling and execution requirements in MES vertical integration. The event routing within the mapping view depends on how the manufacturing process and the plant structure are defined during design-time. This gives the possibility to automatically generate the message routing parts of this mapping view, in case the process and the plant have been designed. We are going to address the generation of the mapping in our prototype.

4.2 Prototyping and Tool Support

This section describes the implemented prototype for the three-view-based approach. The order picking process will be used again to drive this section. This prototype is considered as one of our research methods, “proof-of-concept”, to demonstrate the idea of the proposed integration approach is possible to support MES process execution from academia perspective. Since it is not an important evaluation for choosing suitable tools to develop this approach in real industry context, we have chosen tools and technologies that are freely available and easy to extend for implementation. Therefore, we mostly prefer open source software, especially under the Eclipse framework.
4.2. Prototyping and Tool Support

4.2.1 MES Process View

In the previous sections, we already applied a BPMN modeling tool named Gliffy Business Process Software. However, Gliffy is a web-based modeling tool, which is good for representation, but hardly supports the process execution. To build up the prototype, we need a BPMN tool that not only provides modeling functions, but also contains a relatively good business process management (BPM) engine, because we need to implement and execute some MES services.

After having compared some BPM engines under the Eclipse platform, we decided to use Activiti BPM\(^1\) to implement the MES process model of prototype. The first reason to choose Activiti BPM is that it provides good user experience for BPMN modeling. The second reason is that Activiti BPM provides additional support to some other event processing and management middleware that we need for the prototyping of the mapping view.

Figure 4.8 presents the implementation of the order picking process implemented in Activiti. As can be seen from the figure, Activiti differentiates service types with different symbols. In our process, we utilize the service task (e.g., Determine Location), receive task (e.g., Receive TU) and user task (e.g., Register Picking Box). Additional tasks can be added as extension to Activiti’s built-in task types.

![Figure 4.8: Order Picking Process Implemented in Activiti BPM](image)

Activiti BPM automatically generates and synchronizes an XML configuration based on the process model shown in Figure 4.8. It is named “AKLOrderPicking.bpmn20.xml”, which can be found in Appendix B.1.

\(^1\)Activiti BPM: [http://activiti.org/](http://activiti.org/).
At the bottom of Figure 4.8, the Properties view shows that the Determine Location task in the process is implemented by a Java class, located at “com.siemens.ct.warehouse.service.LocationService”. We mocked up this LocationService.java to return the materials’ position information. The other service tasks in this process are mocked up with Java classes as well.

4.2.2 Abstract Plant View

The implementation of the abstract plant view has two steps. Firstly, we created the pseudo real-world model of warehouse with Eclipse Modeling Framework\(^1\) (EMF). Secondly, we choose the long-living entities from the pseudo real-world model as the meta model, in order to build up a graphical editor of the warehousing plant view based on the EMF meta model.

EMF is a modeling framework for software developers to create structured data models. Based on the data model, EMF provides some additional plugins to support model-to-model, model-to-code, and mode-to-text generation. In EMF, meta models are described using so-called Ecore models.

Implementation of the Warehousing Meta Model

Figure 4.9 presents the screen shot of the pseudo real-world warehouse model implemented in EMF. The meta model includes both design-time and run-time entities.

\[\text{Figure 4.9: The Pseudo Real-World Model of A Warehouse in EMF}\]

On the left side of Figure 4.9, there is a traditional tree view of all warehouse entities. The root of this tree is the warehouse.ecore model. On the right side of

\(^1\)Eclipse Modeling Framework: http://www.eclipse.org/modeling/emf/.
this figure, the class diagram of all entities is presented. As we already explained
in Section 4.1.2, the long-living entities in this pseudo real-world model are: Rack
(with Storage Bin), Conveyor, Stacker Crane, and Workstation (with Buffer). So
the next step is to use these selected entities to build up the abstract plant view.

Implementation of a Graphical Editor for the Abstract Plant View

For creating the graphical editor of abstract plant view, we applied Eclipse Graph-
ical Modeling Framework¹ (GMF), which is an Eclipse plug-in based on EMF.
GMF supports generating graphical editors based on structured Ecore models.
   GMF has a dashboard to help on generating the graphical editor [1]. It has
been applied as follows:

1. Create and select Ecore model as the domain model of GMF. In our case,
   we used our warehouse.ecore model.
2. Create the figures to be displayed on the diagram. What we did in this step
   was to change the default icons created by GMF for different elements.
3. Create the nodes to be shown on the canvas and the palette of the graphical
   editor. As shown in Figure 4.10, we chose Rack, Workstation, Conveyor,
   and Stacker Crane as the elements to define the modeling elements.

4. Map the figures and nodes for the editor: This mapping step is performed
   automatically by GMF.
5. Ensure that each node and its figure are matched: Finally, we check man-
   ually whether the figures and nodes are matched correctly.

After having finished all the generation steps of the GMF dashboard, the editor of Warehousing Abstract View can be generated. Actually, it is an Eclipse plug-in that can be installed to Eclipse development environment. Figure 4.11 shows the screen shot.

![Implementation of the Abstract Plant View](image)

Figure 4.11: Implementation of the Abstract Plant View

In Figure 4.11, with the canvas in the middle and the palette on the right side, we created the warehouse instance for “CustomerA”. It has 6 racks, 3 stacker cranes that connect every two of the racks, 1 conveyor system, and 3 workstations. The Properties view at the bottom allows us to textually input or update all information. Buffer and StorageBin can be added or deleted with the tree view of this CustomerA.warehouse, which looks similar to the tree view of warehouse.ecore. This CustomerA.warehouse is used as input to generate the mapping view, which we explain in the next section.

### 4.2.3 Generating the Mapping View

At the end of Section 4.1.3, we conclude that it should be possible to generate the mapping, between the MES process and the manufacturing plant, when both are defined. We do not consider on generating database linkage part in the mapping view, since the design of data model in database is not our focus in this research. This section describes how we generate the ports for event processing in the
mapping by using Eclipse Xtend\(^1\).

As analyzed in Section 4.1.3, this mapping view is actually the representation of the Time (event) abstraction in MES integration. Sharon and Etzion propose to use messaging mechanism to implement the event processing network [45]. According to [45], it is common to envelope an event into a message to gain better routing and processing support.

The message routing engine we chose for implementing our prototype is Apache Camel\(^2\). One important reason is that Activiti BPM supports already the integration with Apache Camel, which means that the processes modeled in Activiti can send and receive messages from Camel’s routing engine. One alternative is to use Mule Enterprise Service Bus (ESB), because Activiti BPM also supports the integration with Mule ESB, which is called Mule Activiti\(^3\).

**Assumptions**

In current industry context, the communication between MES management systems and PLCs is using protocols, such as TCP or UDP. In this prototype, we envelop the events to messages, and consequently the communication mechanism needs to be adjusted, as shown in Table 4.4.

<table>
<thead>
<tr>
<th>Event Exchange in Industry</th>
<th>Messaging in Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Connection</td>
<td>Java Message Service (JMS) (listening on tcp://localhost:61616)</td>
</tr>
<tr>
<td>Ports</td>
<td>JMS Message queues</td>
</tr>
<tr>
<td>Port Number</td>
<td>JMS Queue name</td>
</tr>
<tr>
<td>TCP Header</td>
<td>JMS Message Header</td>
</tr>
<tr>
<td>TCP Data</td>
<td>JMS Message body (e.g. XML)</td>
</tr>
</tbody>
</table>

*Table 4.4:* From Event to Messaging

We have had a discussion about this assumption with domain experts, and got the feedback that such enveloping of events is reasonable and feasible to reflect the communication in MES.

**Message Routing Generator with Xtend**

Xtend defines a syntax for model-to-text transformation from EMF models to, for example, Java classes, Extensible Markup Language (XML) files, etc. In our chosen routing engine, the message routes are defined in an XML-based configuration file. Therefore, we use Xtend to generate this XML configuration file that

---

\(^1\)Xtend: [http://www.eclipse.org/xtend/](http://www.eclipse.org/xtend/).


describes the message routes between message queues. Figure 4.12 presents the implemented generator.

![Figure 4.12: The Implemented Generator for Mapping View](image)

Xtend allows us to access the elements in the CustomerA.warehouse model. For each rack within the warehouse, the generator retrieves the “id” of Racks, and the “id” of the stacker crane that works for the rack. Then, it generates the corresponding message routes. The output of the generator is an XML file that describes message routes among the process and technical systems. More detailed code of this generator is attached in Appendix B.3.

**Generated Message Routes and Enterprise Integration Patterns**

The Xtend generator generates a mapping that describes message routes. In our prototype, this is an XML configuration file to be used by Apache Camel. In case another routing engine is chosen, the implementation of the generator and the mechanism for routing could be very different. In Appendix B.4, process2plcmapping-context.xml, the complete generated XML for message routing is attached. Figure 4.13 shows one part of the XML configuration that routes the “bring” message to stacker cranes. When a message arrives to the stack crane’s bring port, it is routed to one of the three stacker cranes, based on where the requested materials are stored. For example, if materials are stored on rack 1 or 2, the message goes to stacker crane 1’s port.

However, there could be a better way to represent this message routing than the plain XML-based descriptions. Hohpe and Woolf discover enterprise integration patterns (EIPs) aiming to help architects and developers to describe and design robust software integration solutions [23]. Up to now, 65 EIPs have been identified. These EIPs are categorized into different groups: message routing, message transformation, messaging endpoints, system management, etc. For each
4.2. Prototyping and Tool Support

Figure 4.13: Implementation of EIPs in Apache Camel

pattern, they also define an icon. This gives us the chance to make our mapping view graphically. The configuration XML in Figure 4.13 basically corresponds to two EIPs: a wire tap and a content-based router.

According to Hohpe and Woolf, the wire tap pattern belongs to system management group. A wire tap does not change the message itself. It adds simply a new recipient to the message, and still sends the message to its original destination [23]. In our case, we wire tap the message to a logger. The content-based router pattern belongs to the message routing group. It checks the content of messages, and routes the messages to different endpoints based on defined rules. It works similar to the “switch” statement in some programming languages. In our implementation, messages that request materials from rack 1 and 2, are routed to stacker crane 1; from rack 3 and 4, are routed to stacker crane 2; from rack 5 and 6, are routed to stacker crane 3.

Figure 4.14 presents a small part of the graphical mapping. The whole mapping XML file (see Appendix B.4) has 12 routes. Figure 4.14 contains two of them. It describes the route for the “Bring TU” ports of stacker cranes. Event messages pass through the wire tap and the content-based router, then they are routed to one of the ports of stacker cranes. We implemented a mock-up PLC, which returns the “brough” event to a JMS “brought” port. After the “brought” message is received, the process continues to do the next task. In Figure 4.14, the icons suggested in [23] are applied to graphically represent message routes among endpoints.

In our mapping view, there are two major benefits to apply EIPs for message routing. Firstly, EIPs provide solutions for message routing problems. Besides the
wire tap and the content-based router, other patterns can be useful for implementation of complex event processing, such as aggregator, splitter, publish-subscribe channel, etc. By connecting and combining these patterns, it is easy to establish the message routing configuration. Secondly, to ease the design and development of our Xtend message routing generator, it is good to have such kinds of patterns in the XML configuration template.

4.2.4 Application: Process Execution

After having implemented the MES process view, the abstract plant view, and the mapping view, we can utilize them to execute the order picking process. In this section, we also need to address the run-time information that we decide not to address with separated models. Listing 4.1 presents the code to initialize the order that we introduced in Section 2.1.2. As we analyzed in Section 4.1.2, Order and Pick are the run-time entities. We initialize the order with 3 picks: 80 packages of tee, 50 packages of coffee and 200 packages of milks. (The “Determine Location”, as the first task in the order picking process, would split the 200 packages of milks into two picks. Each of them has 100 as the amount. In real warehouse management system, there should be a user interface provided to warehouse manager to input this information).

```java
private Map<String, Object> prepareVariables() {
    Map<String, Object> variableMap = new HashMap<String, Object>();
    Order order = new Order();
    Pick tee = new Pick("Tee", 80);
    order.getPicks().add(tee);
    Pick coffee = new Pick("Coffee", 50);
    order.getPicks().add(coffee);
    Pick milk = new Pick("Milk", 200);
    order.getPicks().add(milk);
    variableMap.put("order", order);
    variableMap.put("picks", order.getPicks());
    return variableMap;
}
```

Listing 4.1: Run-time entities for Process Execution
4.3. Discussion

As a result, the execution of this order picking requires 4 picks running in parallel. To support this, we enhanced the activiti-camel.jar. Before, the activiti-camel.jar, the integration library between Activiti BPM and Apache Camel, could only support one process instance running in Camel. Details of this code upgrade are attached in Appendix B.2.

List 4.2 presents the code for executing the process. In line 1, the “process2plc-mapping-context.xml” is loaded into the execution context. This is our automatically generated mapping configuration file. In line 3, the “AKLOrderPicking.bpmn20.xml” is deployed. This is the order picking process that we created with Activiti BPM. From line 8 to 10, we start the AKLOrderPicking process with a business key, and the variableMap that contains the information of orders and picks.

In our implementation, service tasks are implemented in Java. However, in order picking process, there are also two manual tasks done by workers: register picking box, and pick. In line 14, we see that manual tasks in the Activiti BPM’s engine can be queried by the assignee’s name, “miao”. In line 15, the tasks assigned to worker “miao”, are marked as completed, so that the process can be continued. In line 18 and 19, we assert that the whole process has been executed, and has stopped properly. The complete code of this test case for executing the order picking process can be found in Appendix B.5.

```
@ContextConfiguration("classpath:routes/process2plc-mapping-context.xml")
public class OrderPickingProcessTest extends SpringActivitiTestCase {

    @Deployment(resources = { "processes/AKLOrderPicking.bpmn20.xml" })
    public void testRunProcess() throws Exception {
        applicationContext.getBean(CamelContext.class);

        // Start a process instance
        Map<String, Object> variableMap = prepareVariables();
        ProcessInstance processInstance = runtimeService
            .startProcessInstanceByKey("AKLOrderPicking",
                "business_key_001", variableMap);

        Thread.sleep(3000);
        tasks = taskService.createTaskQuery().taskAssignee("miao").list();
        completeManualWork(tasks); // worker's manual task

        Thread.sleep(3000);
        assertNotNull(processInstance.getId());
        assertProcessEnded(processInstance.getId());
    }
}
```

Listing 4.2: Process Execution

4.3 Discussion

In Section 4.1, we have proposed the three-view-based approach for MES process modeling and execution, and in Section 4.2 we have described how we implement the prototype as the proof-of-concept. Applying the three-view-based approach
for vertical integration actually provides one candidate architecture to implement the controlling part of a MES work flow engine. In this section we discuss the benefits and drawbacks of using the three-view-based approach in MES.

### 4.3.1 Benefits

The first benefit of using this approach is the separation of concerns. With the MES process view, business analysts and users are able to understand the system behaviors in an easier manner. With the abstract plant view, system engineers (MES vendors) can focus on the structure and deployment of the end system. Having both of these two views reduces the difficulty of re-configuring the system. Therefore, it brings the possibility to reduce the overall cost of implementation.

The second benefit we can expect is the support for effective management. This is one of our evaluation criteria we used in the evaluation of our SLR. In this three-view-based approach, the mapping view not only represents the communication between the other two views, but also enables asynchronous messaging communication within the system. This provides the possibility to improve productivity and to maximize the utilization of resources, compared to existing MES solutions. Having the mapping view, the systems would ease the way to manage and monitor the status about the running processes. In case we wire tap error messages to a monitoring service, system managers or administrators would be able to analyze them, and quickly response to the error. To achieve this, the existing warehouse solution requires more effort.

### 4.3.2 Drawbacks

We also see the potential drawbacks of this approach. The first difficulty is using BPMN as modeling notations. We found that even if BPMN is a standard, different BPMN tools implement and illustrate it in different ways. During our implementation, we encountered some problems because of the ambiguous meaning of BPMN notations. We take the process in warehouse management system as the example to drive this research. But warehouse management systems cover only some of the MES function areas. It is still unclear whether BPMN is applicable to use in the other production or manufacturing processes. One option could be to adapt BPMN or to develop a tailored approach for MES process modeling. This, however, is out of the scope of this thesis.

The second drawback we see from our approach is the generation of the mapping view. In our prototype, the mapping view is fully generated based on the instance of the abstract plant model. This is because our abstract plant model has
a relatively simple structure. We only have racks, workstations, stacker cranes, and conveyors. However, in a real manufacturing plant it is highly possible that the abstract plant model becomes more complicated. This would lead to a much more difficult mapping for message routes. Consequently, to develop a generator would be more difficult. As the generator can reduce the implementation effort each time a warehouse is to be delivered to a customer, a generator with high implementation effort has the potential to pay off over time.

4.4 Summary

In this chapter, we suggest a three-view-based integration approach to overcome the limitation of using SpeziMES for process execution. We propose an abstract plant view to model the structure of long-living objects in manufacturing plants. Moreover, one additional mapping is created in order to bridge the process view and the abstract plant view.

We implement a prototype, as a proof-of-concept of our approach. The tools and techniques we used in prototyping are summarized in Table 4.5. With these three views, it is possible to achieve vertical integration between the MES layer and the Control layer.

<table>
<thead>
<tr>
<th>Views</th>
<th>Modeled Information</th>
<th>(Implemented) Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>MES Process view</td>
<td>The Process and activities</td>
<td>We used Activiti BPM.</td>
</tr>
<tr>
<td>Abstract Plant View</td>
<td>Structure of the plant</td>
<td>We developed a graphical editor with Eclipse EMF and GMF plug-ins.</td>
</tr>
<tr>
<td>Mapping View</td>
<td>The ports for sending and receiving event messages</td>
<td>We developed a generator for the XML configuration.</td>
</tr>
</tbody>
</table>

*Table 4.5: Views and Prototyping*
Chapter 5 Validation

According to our research process, this chapter is the industrial validation step. In order to get the answer to RQ3, we conducted semi-structured interviews in Siemens, by following the suggestions in [25]. This chapter describes the planning phase, the conducting phase, and the results of the interviews.

5.1 Planning and Conducting the Interview

The steps of planning and conducting the interviews are as follows:

1. A presentation was prepared with 17 pages of slides, regarding the idea of the proposed three-view-based approach.

2. A questionnaire was prepared for the interviews. The questions in this questionnaire are separated into two parts (The detailed questions can be seen in Section 5.1.1):

   - In the first part, we formulated questions covering the four criteria of the “execute” aspect (cf. Table 3.6) that we used to evaluate the existing manufacturing modeling techniques in Chapter 3. The reason to restrict the interview questions to this aspect is that, according to our evaluation, SpeziMES already well fulfills the criteria of the “describe” and “analyze” aspects. With the questions, we therefore intended to understand whether the proposed three-view-based approach, as an extension of SpeziMES, could improve in the “execute” aspect.
   - In the second part, we formulated questions to compare our solution with the existing warehouse solution. The questions are related to the understandability of BPMN, the reconfigurability, and the generalization of the approach to other sub-domains within the MES domain.
   - Only the last question regarding the generalization is an open-ended question. For all the other structured questions, we used ordinal measures ranging from 1 to 5. They are explained in Table 5.1:
Chapter 5. Validation

### Table 5.1: Ordinal Measures

<table>
<thead>
<tr>
<th>Measures</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very positive</td>
</tr>
<tr>
<td>2</td>
<td>Positive</td>
</tr>
<tr>
<td>3</td>
<td>Neutral</td>
</tr>
<tr>
<td>4</td>
<td>Negative</td>
</tr>
<tr>
<td>5</td>
<td>Very negative</td>
</tr>
</tbody>
</table>

3. Interviews were scheduled. Due to the time constraints of doing this Master thesis, only three industrial interviews were conducted. All three interviewees are software architects in Siemens. Table 5.2 presents general information about the interviews and interviewees. Two of them are domain experts in warehouse management systems.

<table>
<thead>
<tr>
<th>Interviews</th>
<th>Details</th>
</tr>
</thead>
</table>
| 1          | Interviewee: Software Architect  
|            | Year of Experience in Industry: 10  
|            | Time of Interview: May, 2012  
|            | Location: Erlangen, Germany  |
| 2          | Interviewee: Software Architect  
|            | Year of Experience in Industry: 8  
|            | Time of Interview: May, 2012  
|            | Location: München, Germany  |
| 3          | Interviewee: Software Architect  
|            | Year of Experience in Industry: 5  
|            | Time of Interview: May, 2012  
|            | Location: Erlangen, Germany  |

### Table 5.2: General Information about the Interviews

4. Both the slides of presentation and the questionnaire were sent to the interviewees before the interview, so that they have time to read and understand the context of the proposed approach, as well as the questionnaire.

5. Starting the interview, the presentation was given for around 20 minutes.

6. The questions in the first part of questionnaire were asked and answered by the interviewees.

7. The prototype was shown to the interviewees, including the BPMN process modeling tool in Eclipse, the graphical editor of the abstract plant view, and the message routing generator.

8. The questions in the second part of questionnaire were asked.

9. Before this Master thesis was finalized, a version of this thesis including the summary of the interviews was sent to all three of our interviewees. Two
of them replied and confirmed that the conclusion of the interviews convey their opinions.

5.1.1 The Questionnaire

1. *Questions regarding the criteria from SLR (for RQ3.1)*
   In Chapter 3, some evaluation criteria are proposed to evaluate the existing modeling techniques from the SLR. The result of evaluation implies that there is a lack of regard for the process execution aspect. We developed the following questions based on the four criteria in the process execution aspect, in order to get practical feedback about how good our approach satisfies these criteria.

   (a) Is it practical to implement the ideas of our approach in real industry context?
   (b) How good does our approach support system management and monitoring?
   (c) How is the coherence of the three views?
   (d) Do you think the mapping view supports good interoperability between the services in the MES layer and the PLCs in the control layer?

2. *Questions regarding the existing solutions (for RQ3.2)*
   The following questions were developed especially for evaluating how the three-view-based approach can support the understandability and reconfigurability (or flexibility) of warehouse systems, since these quality attributes are the limitations we have identified in Chapter 2.

   (a) What is your opinion about BPMN?
      i. Do you think BPMN notations are easy to understand by customers?
      ii. Are BPMN notations capable of modeling MES processes?
   (b) How good is the reconfigurability (flexibility) of the process and real-world entities in the warehouse?
      i. Scenario 1: Is it easy to add a manual task to the process model? (e.g., add a quality control task)
      ii. Scenario 2: Is it easy to add a service task to the process model? (e.g., add a weight checking task)
      iii. Scenario 3: Is it easy to add a PLC control task to the process model? (e.g., add a checking weight device)
      iv. Scenario 4: Is it easy to add new racks to the warehouse?
v. Scenario 5: Is it easy to add a new workstation to the warehouse?

(c) How to generalize the approach\(^1\)?

i. Is it possible to apply the idea of this approach without using our tool?

ii. How possible is it to generalize our integration approach (e.g., from warehousing to production)?

iii. Is the generation of the mapping scalable?

### 5.2 Reporting on the Interview

Table 5.3 presents the result of our specific questions using the ordinal measures shown in Table 5.1. The interviews have been conducted individually with three interviewees to avoid influences among them, so that we could get more direct feedback. Since the location of interviewee 2 is in München, the interview was conducted via telephone conference. The limited knowledge of the phone interviewee of BPMN and SpeziMES and the time constraints of the phone interview might have been the problem for our interviewee to answer some of the questions, such as question 2.a.ii. Additionally, from question 2.b.i to 2.b.v, interviewee 2 also differentiates two situations that could lead to two different answers to these questions. The meaning of results from the semi-structure interviews are interpreted and analyzed in this section in the following, as the answer to our RQ3.

<table>
<thead>
<tr>
<th>Question No.</th>
<th>Interviewee 1</th>
<th>Interviewee 2</th>
<th>Interviewee 3</th>
<th>Average value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.a.</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1.3</td>
</tr>
<tr>
<td>1.b.</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1.3</td>
</tr>
<tr>
<td>1.c.</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2.6</td>
</tr>
<tr>
<td>1.d.</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2.3</td>
</tr>
<tr>
<td>2.a.i.</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1.7</td>
</tr>
<tr>
<td>2.a.ii.</td>
<td>3</td>
<td>-</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>2.b.i.</td>
<td>2</td>
<td>2 or 5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>2.b.ii.</td>
<td>2</td>
<td>2 or 5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>2.b.iii.</td>
<td>2</td>
<td>2 or 5</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>2.b.iv.</td>
<td>1</td>
<td>2 or 5</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>2.b.v.</td>
<td>1</td>
<td>2 or 5</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 5.3: The Results of Specific Questions*  

\(^1\)This is the only open-ended question in this questionnaire, all the other questions are measured with the ordinal measures 1 to 5, as explained in Table 5.1.
5.2. Reporting on the Interview

The answer to: **RQ 3.1 How does the improvement compare to the existing modeling techniques?**

In Table 3.6 (in Chapter 3), we have already evaluated the existing modeling techniques from SLR with three groups of criteria. Focusing on the four process execution criteria, we give the answer to RQ3.1 as follows:

1. *Support for faithful implementation:* The feedback is positive in this aspect. In comparison with the existing modeling techniques, the combination of three views provides the chance to execute the processes, and provides the possibility to apply the approach in real industry context.

2. *Support for effective management:* The general feedback is positive regarding this aspect, in contrast to previous approaches, the mapping view gives the advantage to diagnose the system status, thus improving system management. Although MES need effective management, the existing modeling techniques found in the SLR have not put enough attention on this topic.

3. *Coherence of different views:* The feedback regarding this aspect is not so positive. The idea of generating the mapping based on the MES process and the plant abstraction is to bridge these two “parts”. In comparison to existing modeling techniques, the idea is considered as an advantage. However, one interviewee raised an issue that the relationships between technical systems (PLCs) and the ports are still unclear. As these relationships expressed by the mapping view, a clearer representation of its contents (e.g., via a graphical notation) could be of value as future work. The second issue mentioned by another interviewee is that the visualized connectivity among the three views is still missing. In our prototype, to create the three views, different tools are utilized. What we could imagine is to develop one plugin, for example in Eclipse, to display three views within one single modeling tool. As this will require quite significant development efforts, it is left to future work.

4. *Interoperability & Integrability:* The opinions about this aspect differ among our interviewees. Interviewee 2 argued that the interoperability of the MES processes and the underlying PLCs it controls is not improved by having a generated mapping in between, as it does not provide special benefits. The reason is that without the generated mapping, the services and PLCs could still communicate with each other, it only is necessary to manually implement this mapping. However, when thinking about the integrability of MES, having such a mapping “layer” gives the chance to support different communication protocols by having dedicated generators for them. By doing so, the integrability of MES could be improved, as argued by the
other two interviewees.

The answer to: **RQ 3.2 How does the approach of improvement solve the limitation of the existing warehouse solution?**

1. *About BPMN:* Regarding the understandability of BPMN, all of our interviewees gave positive answers. They consider BPMN is easy to understand by the users without technical background. However, BPMN, as a relatively standardized modeling language, requires high efforts to extend and customize for specific MES domains to really use it in industry.

2. *About reconfigurability and flexibility:* The general feedback to this aspect is relatively positive. In case the editors of the MES process and the abstract plant are comprehensive enough and the generator of the mapping is powerful enough, the flexibility of the systems can be improved. For interviewee 2, this leads to a “2”, which is a positive answer. In the context of this Master thesis, we utilized the existing BPMN editor, developed a graphical editor for the abstract plant view, and created a generator of the mapping. The effort of developing these editors and the generator in real industry context would be rather high. Interviewee 2 argued that, without having such industrial-strength comprehensive editors, the effort for adapting the process or the technical system might in turn also require changes to the editors. If this would be the case his answer to this question would be a “5”, and therefore negative.

3. *About generalization of the approach:*
   - The interviewees commented that it should be possible with reasonable effort to replace the tools we used in the prototype to other open source or commercial tools. For example, Workflow Foundation could be used, instead of Activit BPM, or IBM WebSphere MQ instead of ActiveMQ.
   - In the opinion of the interviewees, it should be possible to generalize the idea of integrating the process and the real-world model to other domains. BPMN, however, may not always be the best option for modeling very complex processes. In the production domain, for example, it should be possible to reuse the approach with a domain-specific process modeling language tailored to this specific domain.
   - According to the answers from our interviewees, the approach could be scalable. In a very large warehouse, the number of conveyors can reach more than hundreds. In this case, the effort and cost for developing editors and the generator can pay off, since they would help reducing the complexity of delivering new warehouse systems to customers.
Chapter 6

Threats to Validity

Validity threats need to be considered during a research project to ensure the credibility of findings [57]. In this section we discuss the validity threats of the results produced in the context of this thesis.

6.1 Validity Threats to the Literature Review

Conclusion validity concerns how reasonable it is to make the conclusion based on the obtained information. If other researchers apply the same research protocol again, it is possible that some other articles could be selected. The reason is that, by reading and interpreting titles and abstract of the potential studies, the selection and evaluation of papers turn to be subjective. To alleviate this problem, the selected studies and the evaluation are submitted to the supervisors from both academia and industry for re-examination and inspection. The supervisor from University of Kaiserslautern carefully checked the evaluation criteria and results, showed in Table 3.6. The industrial supervisor had also read and agreed with the results of our systematic literature review. Thus, we decided to take the conclusion we made from Chapter 3 as the foundation for our solution and validation.

There is also a threat to the coverage of literature. The search strings are defined to harvest all the existing studies in MES process modeling and execution. However, there is still a possibility that some articles are missing in our final collection. The reason could that by applying the inclusion/exclusion criteria, some articles are mistakenly neglected. There might also be other articles in the other scientific databases that are not used in our research. To eliminate this threat, we conducted snowball searching to include other relevant articles which were not originally from our literature searching process.
Chapter 6. Threats to Validity

6.2 Validity Threats to Prototyping

In the context of this Master thesis, the warehouse management system, as one of MES, is chosen to analyze the problem of modeling and executing MES processes. Both the proposed approach and a prototype are established on warehouse management systems. In our prototype, a typical process in a warehouse management system is modeled, and shown to be executable by following the proposed approach to establish the abstract plant view and mapping view.

However, this leads to a potential threat to external validity, because the processes in warehouse management systems may not be enough to represent all the critical challenges in MES processes. To alleviate this threat, we discussed with domain experts within Siemens Cooperate Technology, and chose the order picking process in warehouse as the exemplary process, which at least shows most of the complexities in warehouse management systems. We also analyzed in Chapter 2 that the critical integration challenges in warehouse management systems are valid for the other MES domain. Therefore, there is a high potential to generalize this idea into other MES domain. Furthermore, we added open-ended questions in the questionnaire to explicitly ask our interviewees about the the possibility to generalize the proposed approach. The answers we gathered show relatively positive opinions with regard to generalization.

6.3 Validity Threats to the Semi-Structured Interviews

The construct validity and external validity constitute the most significant threats to the credibility of the results of semi-structure interviews.

The purpose of conducting the semi-structured interviews is to validate: RQ3. How does the improved approach overcome the limitations of the existing solution? To ensure construct validity, it is important to construct questions to effectively measure the improvement in the questionnaire. During the planning phase of the interviews, the questions were sent to the supervisors both from academia and industry for review. Based on the feedback from the industrial supervisor, the questions were adjusted, to make them easier understandable for industrial engineers. We decided to separate the questionnaire into two parts. In the first part, we took the four criteria in the “execute” aspect of Chapter 3 to build up the four questions. Each of the questions is aiming at validating one criterion. In the second part of the questionnaire, we constructed the questions to understand how good it is to use our approach to improve the understandability and
reconfigurability of the existing warehouse solution. Therefore, we have questions in this part that are intended to validate these non-functional attributes that we identified as limitations in Section 2.2.

There is also an external validity threat to the result of the semi-structured interviews, regarding how possible it is to generalize the three-view-based approach into the other manufacturing domain [48], or even to other MES vendors outside Siemens. Due to time and resource constraints, we could not establish interviews outside Siemens. All three of our interviewees are from Siemens Cooperate Technology in Germany. Further validation outside Siemens would still be necessary to verify the possibility of generalization of the proposed approach.
Chapter 7

Conclusion and Future Work

This Master thesis has presented a three-view-based approach, aiming at bridging MES and control layers in non-enterprise integration. A warehouse management system is used as the example of MES, to formulate the problems, to present the approach, and to validate the approach.

The systematic literature review (SLR) in Chapter 3 is conducted in five scientific databases to understand the state of the art of current process modeling techniques. The selected primary articles are categorized into four groups: traditional models, object-oriented models, process decomposition, and multiple views & hybrid techniques. We identified three aspects—the “describe”, the “analyze”, and the “execute” aspect—that need to be considered for supporting process execution. Each aspect is further refined into a number of criteria, which are then used for evaluating the selected studies. The result of the evaluation shows that the SpeziMES approach fulfills the evaluation criteria best among all selected studies. Additionally, the limitation of all previous approaches is pointed out, which is a lack of support for executing processes on the PLCs.

To overcome this limitation, Chapter 4 proposes a three-view-based approach. Firstly, we propose to have an MES process view, which, in line with SpeziMES, uses BPMN as modeling notation. Secondly, an abstract plant view is suggested to represent the real-world and long-living objects in manufacturing plants. These long-living objects include technical systems (PLCs) and the technical systems that are not changed during run-time. As the concrete types of these objects are different depending on the concrete sub-domain of an MES, we propose a method to create a suitable meta model for the abstract plant view, which derives the relevant object types from the domain model of the concrete sub-domain. Thirdly, a mapping view is generated based on the MES process view and the abstract plant view. It represents event processing at run-time. It describes how events are routed and exchanged in between a running MES process and the technical system within a manufacturing plant. Finally, as a proof of concept, we developed a prototype, which comprises a graphical editor for the abstract plant view, a
generator of message routes for the mapping view, and a typical MES process to be executed in the context of a warehouse management system.

As validation of the proposed three-view-based approach, semi-structured interviews with three industrial experts were conducted to collect feedback on the approach from industry. The responses were positive with regard to applying the idea of the approach in industry context, as well as for the generalizability of the approach to other MES sub-domains.

**Future Work**

Whether BPMN is sufficient to comprehensively represent various manufacturing or production processes still needs to be analyzed. The result of this research indicates that BPMN is suitable for modeling warehousing processes. However, warehouse management systems do not cover all MES function areas. As future work, it is important to examine the application of BPMN to these other function areas as well. Tailoring or customizing the BPMN language and the corresponding editors, or even the design of a dedicated domain-specific language, might be required depending on the needs of the respective MES sub-domain where it is applied.

In order to generalize the result, it is necessary to further apply the proposed three-view-based approach to some other MES sub-domains, not only limited to warehouse systems. To achieve this, the representation power of BPMN might still need to be re-evaluated. The meta model and the graphical editor of the abstract plant view need to be created according to different sub-domains of MES. And the message routing generator would also need to be re-developed.
Appendix A

Search Terms of Systematic Literature Review

<table>
<thead>
<tr>
<th>Database Name</th>
<th>Search Terms</th>
<th>Execution Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Xplore</td>
<td>(&quot;Process model&quot; OR &quot;modeling language&quot; OR &quot;Process view&quot; OR &quot;workflow model&quot;) AND manufacturing process</td>
<td>Dec. 2011</td>
</tr>
<tr>
<td>ACM Digital Library</td>
<td>(&quot;manufacturing process&quot;) and (&quot;Process model&quot; or &quot;modeling language&quot; or &quot;Process view&quot; or &quot;workflow model&quot;)</td>
<td>Dec. 2011</td>
</tr>
<tr>
<td></td>
<td>(Abstract manufacturing) and (Abstract &quot;process enabling&quot;, Abstract &quot;process execution&quot;)</td>
<td>Mar. 2012</td>
</tr>
<tr>
<td>Springer Link</td>
<td>(&quot;Process model&quot; or &quot;Process view&quot; or &quot;workflow model&quot;) and &quot;manufacturing process&quot; with filters: English, Journal Articles, Computer Science</td>
<td>Dec. 2011</td>
</tr>
<tr>
<td></td>
<td>(&quot;Process enabling&quot; or &quot;process execution&quot;) and manufacturer*</td>
<td>Mar. 2012</td>
</tr>
<tr>
<td></td>
<td>(((Manufacturing OR Production) WN All fields) AND (process execution) WN All fields)) AND (process enabling) WN All fields))</td>
<td>Mar. 2012</td>
</tr>
<tr>
<td>Science Direct</td>
<td>&quot;Process model&quot; OR &quot;modeling language&quot; OR &quot;Process view&quot; OR &quot;workflow model&quot; AND manufacturing</td>
<td>Dec. 2011</td>
</tr>
<tr>
<td></td>
<td>TITLE-ABSTR-KEY(&quot;Process enabling&quot; OR &quot;process execution&quot;) ) and TITLE-ABSTR-KEY(Manufacturer)</td>
<td>Mar. 2012</td>
</tr>
</tbody>
</table>

*Figure A.1:* Combination of Search Terms in Each Database
Appendix B

Implementation Details

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:activiti="http://activiti.org/bpmn"
expressionLanguage="http://www.w3.org/1999/XPath"
targetNamespace="http://www.activiti.org/test">

<!−−Due to some limitations, we do not display the complete BPMN file here−−>
</definitions>
</definitions>
<bpmn: BPMNDiagram id="BPMNDiagram_AKLOrderPicking">
  <bpmn:BPMNPlane bpmnElement="AKLOrderPicking"
    id="BPMNPlane_AKLOrderPicking">
    <bpmn:BPMNShape bpmnElement="startevent1"
      id="BPMNShape_startevent1"
      <omgdc:Bounds height="35" width="35" x="10" y="182"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="determineLocation"
      id="BPMNShape_determineLocation"
      <omgdc:Bounds height="55" width="105" x="60" y="172"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="pickingSubProcess"
      id="BPMNShape_pickingSubProcess"
      <omgdc:Bounds height="311" width="683" x="178" y="110"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="startevent2"
      id="BPMNShape_startevent2"
      <omgdc:Bounds height="35" width="35" x="188" y="150"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="requestTU"
      id="BPMNShape_requestTU"
      <omgdc:Bounds height="55" width="105" x="238" y="140"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="transportTU"
      id="BPMNShape_transportTU"
      <omgdc:Bounds height="55" width="105" x="290" y="222"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="receiveTU"
      id="BPMNShape_receiveTU"
      <omgdc:Bounds height="55" width="105" x="348" y="140"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="exclusivegateway1"
      id="BPMNShape_exclusivegateway1"
      <omgdc:Bounds height="40" width="468" y="147"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="inclusivegateway1"
      id="BPMNShape_inclusivegateway1"
      <omgdc:Bounds height="40" width="524" y="147"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="registerBox"
      id="BPMNShape_registerBox"
      <omgdc:Bounds height="55" width="468" y="222"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="endevent3"
      id="BPMNShape_endevent3"
      <omgdc:Bounds height="35" width="35" x="800" y="251"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="pick"
      id="BPMNShape_pick"
      <omgdc:Bounds height="55" width="105" x="572" y="140"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="returnTU"
      id="BPMNShape_returnTU"
      <omgdc:Bounds height="55" width="105" x="750" y="140"/>
    </bpmn:BPMNShape>
    <bpmn:BPMNShape bpmnElement="qa"
      id="BPMNShape_qa">
      <omgdc:Bounds height="55" width="105"/>
    </bpmn:BPMNShape>
  </bpmn:BPMNPlane>
</bpmn:BPMNDiagram>
</definitions>
```
Listing B.1: AKLOrderPicking.bpmn20.xml
public class ActivitiProducer extends DefaultProducer {
    private RuntimeService runtimeService;
    public static final String PROCESS_KEY_PROPERTY = "PROCESS_KEY_PROPERTY";
    public static final String PROCESS_ID_PROPERTY = "PROCESS_ID_PROPERTY";
    public static final String TASK_EXECUTION_ID = "TASK_EXECUTION_ID";
    private String processKey = null;
    private String activity = null;

    public ActivitiProducer(ActivitiEndpoint endpoint, RuntimeService runtimeService) {
        super(endpoint);
        this.runtimeService = runtimeService;
        String[] path = endpoint.getEndpointKey().split(":");
        processKey = path[1].replace("/", "");
        if (path.length > 2) {
            activity = path[2];
        }
    }

    public void process(Exchange exchange) throws Exception {
        if (shouldStartProcess()) {
            ProcessInstance pi = startProcess(exchange);
            exchange.getOut().setBody(pi.getId());
        } else {
            signal(exchange);
        }
    }

    private boolean shouldStartProcess() {
        return activity == null;
    }

    private void signal(Exchange exchange) {
        String processInstanceId = findProcessInstanceId(exchange);
        List<Execution> executions = runtimeService.createExecutionQuery()
            .processDefinitionKey(processKey)
            .processInstanceId(processInstanceId).activityId(activity).list();
        if (executions == null) {
            throw new RuntimeException("Couldn’t find activity “ + activity + ” for processId “ + processInstanceId);
        } else {
            for (Execution execution : executions) {
                if (this.findTaskExecutionId(exchange).equals(execution.getId())) {
                    runtimeService.setVariables(execution.getExecutionId(), ExchangeUtils
                        .prepareVariables(exchange, getActivitiEndpoint()));
                    runtimeService.signal(execution.getExecutionId());
                    return;
                }
            }
        }
    }

    private String findProcessInstanceId(Exchange exchange) {
        String processInstanceId = exchange.getProperty(PROCESS_ID_PROPERTY, String.class);
        if (processInstanceId != null) {
            return processInstanceId;
        }
        String processInstanceKey = exchange.getProperty(PROCESS_KEY_PROPERTY, String.class);
        ProcessInstance processInstance = runtimeService
            .createProcessInstanceQuery().processInstanceBusinessKey(processInstanceKey).singleResult();
        if (processInstance == null) {
            throw new RuntimeException("Could not find activity with key “ + processInstanceKey);
        }
        return processInstance.getId();
    }

    private String findTaskExecutionId(Exchange exchange) {
        String taskExecutionId = exchange.getProperty(TASK_EXECUTION_ID, String.class);
        if (taskExecutionId != null) {
            return taskExecutionId;
        }
        return "";
    }
}
private ProcessInstance startProcess(Exchange exchange) {
    String key = exchange.getProperty(PROCESS_KEY_PROPERTY, String.class);
    if (key == null) {
        return runtimeService.startProcessInstanceByKey(processKey,
            ExchangeUtils.prepareVariables(exchange, getActivitiEndpoint()));
    } else {
        return runtimeService.startProcessInstanceByKey(processKey, key,
            ExchangeUtils.prepareVariables(exchange, getActivitiEndpoint()));
    }
}

Listing B.2: ActivitiProducer.java
package generator
import org.eclipse.emf.ecore.resource.Resource
import org.eclipse.xtext.generator.IGenerator
import org.eclipse.xtext.generator.IFileSystemAccess
import org.eclipse.emf.ecore.EObject
import org.eclipse.xtext.naming.IQualifiedNameProvider
import warehouse.Warehouse
import warehouse.Rack
import warehouse.StackerCrane
import warehouse.WorkStation
import warehouse.Conveyor
import com.google.inject.Inject
class MyGenerator implements IGenerator {
  @Inject extension IQualifiedNameProvider nameProvider
  override void doGenerate(Resource resource, IFileSystemAccess fsa) {
    for (EObject obj : resource.contents)
      { fsa.generateFile("process2hardware-mapping-context.xml", obj.createMapping) }
  }

def createMapping(Warehouse warehouse)
    """<xml version="1.0" encoding="UTF-8"/>
    <beans>
    <bean id="jms" class="org.apache.camel.component.jms.JmsComponent">
        <property name="connectionFactory">
            <bean class="org.apache.activemq.ActiveMQConnectionFactory">
                <property name="brokerURL" value="tcp://localhost:61616"/>
                </bean>
        </property>
    </bean>
    <bean id="stacker_crane_plc" class="com.siemens.ct.warehouse.mock.StackerCrane"/>
    <bean id="conveyor_plc" class="com.siemens.ct.warehouse.mock.Conveyor"/>
    <bean id="controller" class="com.siemens.ct.warehouse.mock.PlcController"/>
    <camelContext id="AKLOrderPicking" xmlns="http://camel.apache.org/schema/spring">
        <route>
            <from uri="jms://PLC_controller-stackCrane-bring"/>
            <wireTap uri="direct:messageLogger"/>
            FOR rack: warehouse.racks
            rack.iterateRack
            ENDFOR
            <otherwise>
                <stop/>
            </otherwise>
            </route>
            FOR stackerCrane: warehouse.stackerCranes
            stackerCrane.iterateStackerCrane
            ENDFOR
            FOR conveyor: warehouse.conveyors
            conveyor.iterateConveyor
            ENDFOR
            <route>
                <from uri="jms://PLC_controller-conveyor-return"/>
                <bean ref="conveyor_plc" method="returnTU"/>
                <bean ref="controller" method="returnTU"/>
                <wireTap uri="direct:messageLogger"/>
                </route>
            <route>
                <from uri="activiti:AKLOrderPicking:transportTU"/>
                <stop/>
            </route>
            </route>
            <from uri="direct:messageLogger"/>
            <choice>
                FOR workStation: warehouse.workStations
                workStation.iterateWorkstation
                ENDFOR
                <otherwise><stop/></otherwise>
            </choice>
            </route>
            </camelContext>
    </beans>
```

### Appendix B. Implementation Details

```java

```
<?xml version="1.0" encoding="UTF-8"?>
<beans>
  <bean id="jms" class="org.apache.camel.component.jms.JmsComponent">
    <property name="connectionFactory">
      <bean class="org.apache.activemq.ActiveMQConnectionFactory">
        <property name="brokerURL" value="tcp://localhost:61616"/>
      </bean>
    </property>
  </bean>
  <bean id="controller" class="com.siemens.ct.warehouse.mock.PlcController"/>
  <bean id="conveyor plc" class="com.siemens.ct.warehouse.mock.Conveyor"/>
  <bean id="stacker plc" class="com.siemens.ct.warehouse.mock.StackerCrane"/>
  <camelContext id="AKLOrderPicking" xmlns="http://camel.apache.org/schema/spring">
    <route>
      <from uri="jms://PLC_controller-stackCrane-bring"/>
      <wireTap uri="direct:messageLogger"/>
      <choice>
        <when>
          <xpath>
            /bring/from[starts-with(text(), 'rack-01')]
          </xpath>
          <to uri="jms://stackerCrane-01-bring"/>
        </when>
        <when>
          <xpath>/bring/from[starts-with(text(), 'rack-02')]
          <to uri="jms://stackerCrane-01-bring"/>
        </when>
        <when>
          <xpath>/bring/from[starts-with(text(), 'rack-03')]
          <to uri="jms://stackerCrane-02-bring"/>
        </when>
        <when>
          <xpath>/bring/from[starts-with(text(), 'rack-04')]
          <to uri="jms://stackerCrane-02-bring"/>
        </when>
        <when>
          <xpath>/bring/from[starts-with(text(), 'rack-05')]
          <to uri="jms://stackerCrane-03-bring"/>
        </when>
        <when>
          <xpath>/bring/from[starts-with(text(), 'rack-06')]
          <to uri="jms://stackerCrane-03-bring"/>
        </when>
        <otherwise>
          <stop/>
        </otherwise>
      </choice>
    </route>
    <route>
      <from uri="jms://PLC_controller-stackCrane-brought"/>
      <bean ref="stacker Crane plc" method="bring"/>
      <to uri="jms://PLC_controller-stackCrane-brought"/>
    </route>
    <route>
      <from uri="jms://stackerCrane-01-return"/>
      <bean ref="stacker Crane plc" method="returnTU"/>
    </route>
    <route>
      <from uri="jms://PLC_controller-stackCrane-brought"/>
      <bean ref="stacker Crane plc" method="bring"/>
      <to uri="jms://PLC_controller-stackCrane-brought"/>
    </route>
    <route>
      <from uri="jms://stackerCrane-02-return"/>
      <bean ref="stacker Crane plc" method="returnTU"/>
    </route>
    <route>
      <from uri="jms://PLC_controller-stackCrane-brought"/>
      <bean ref="stacker Crane plc" method="bring"/>
      <to uri="jms://PLC_controller-stackCrane-brought"/>
    </route>
    <route>
      <from uri="jms://stackerCrane-03-return"/>
      <bean ref="stacker Crane plc" method="returnTU"/>
    </route>
  </camelContext>
</beans>
Listing B.4: process2plc-mapping-context.xml
package processes;

import java.util.HashMap;

@ContextConfiguration("classpath:routes/process2hardware-mapping-context.xml")
public class OrderPickingProcessTest extends SpringActivitiTestCase {

    @Deployment(resources = { "processes/AKLOrderPicking.bpmn20.xml" })
    public void testRunProcess() throws Exception {
        applicationContext.getBean(CamelContext.class);
        // Start a process instance
        Map<String, Object> variableMap = prepareVariables();
        ProcessInstance processInstance = runtimeService
                .startProcessInstanceByKey("AKLOrderPicking",
                "business_key_001", variableMap);

        Thread.sleep(3000);
        List<Task> tasks = taskService.createTaskQuery()
                .taskName("Register Picking Box").list();
        assertEquals(4, tasks.size());
        System.out.println("=*=*=*=*=*=*=*=*= Manually Work - Register picking Box =*=*=*=*=*=*=*=*==");
        completeManualWork(tasks);
        runtimeService.setVariableLocal(processInstance.getId(),
                "pickingBoxExists", true);

        Thread.sleep(3000);
        tasks = taskService.createTaskQuery().taskName("Pick").list();
        assertEquals(4, tasks.size());
        System.out.println("=*=*=*=*=*=*=*=*= Manually Work - Do picking =*=*=*=*=*=*=*=*==");
        completeManualWork(tasks); // worker’s manual task

        Thread.sleep(3000);
        tasks = taskService.createTaskQuery().taskAssignee("miao").list();
        assertEquals(1, tasks.size());
        assertEquals("Merge Picks", tasks.get(0).getName());
        System.out.println("=*=*=*=*=*=*=*=*= Manually Work - Merge picks =*=*=*=*=*=*=*=*==");
        completeManualWork(tasks); // worker’s manual task

        Thread.sleep(3000);
        assertNotNull(processInstance.getId());
        assertProcessEnded(processInstance.getId());

        private void completeManualWork(List<Task> tasks) {
            for (Task task : tasks) {
                System.out.println("Task name : " + task.getName()
                        + "--Assignee : " + task.getAssignee());
                taskService.complete(task.getId());
            }
        }

        private Map<String, Object> prepareVariables() {
            Map<String, Object> variableMap = new HashMap<String, Object>();
            Order order = new Order();
            Pick tee = new Pick("Tee", 80);
            order.getPicks().add(tee);
            Pick coffee = new Pick("Coffee", 50);
            order.getPicks().add(coffee);
            Pick milk = new Pick("Milk", 200);
            order.getPicks().add(milk);
            variableMap.put("order", order);
            variableMap.put("pickingBoxExists", false);
            return variableMap;
        }
    }
}

Listing B.5: OrderPickingProcessTest.java


