

Complex Systems Informatics and Modeling Quarterly (CSIMQ)

eISSN: 2255-9922

Published online by RTU Press, <https://csimq-journals.rtu.lv>

Article 111, Issue 19, June/July 2019, Pages 42–59

<https://doi.org/10.7250/csimq.2019-19.03>

CSIMQ

Complex
Systems
Informatics
and
Modeling
Quarterly

Automated Modeling with Abstraction for Enterprise Architecture (AMA4EA): Business Process Model Automation in an Industry 4.0 Laboratory

Marco Nardello^{1*}, Shengnan Han², Charles Møller¹ and John Gøtze^{3,4}

¹ Aalborg University, Department of Materials and Production, Fibigerstræde 16, DK-9220 Aalborg East, Denmark

² Stockholm University, Department of Computer and Systems Sciences, Borgarfjordsgatan 12, 164 07 Kista, Sweden

³ IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen, Denmark

⁴ QualiWare aps, Ryttermarken 15, 3520 Farum, Denmark

man@mp.aau.dk, shengnan@dsv.su.se, charles@mp.aau.dk, jogo@itu.dk

Abstract. The transformation towards the Industry 4.0 paradigm requires companies to manage large amounts of data. This poses serious challenges with regard to how effectively to handle data and extract value from it. The state-of-the-art research of Enterprise Architecture (EA) provides limited knowledge on addressing this challenge. In this article, the Automated Modeling with Abstraction for Enterprise Architecture (AMA4EA) method is proposed and demonstrated. An abstraction hierarchy is introduced by AMA4EA to support companies to automatically abstract data from enterprise systems to concepts, then to automatically create an EA model. AMA4EA was demonstrated at an Industry 4.0 laboratory. The demonstration showed that AMA4EA could abstract detailed data from the Enterprise Resource Planning (ERP) system and Manufacturing Execution System (MES) to be relevant for a business process model that provided a useful and simplified visualization of production process data. The model communicated the detailed business data in an easily understandable way to stakeholders. AMA4EA is an innovative and novel method that contributes new knowledge to EA research. The demonstration provides sufficient evidence that AMA4EA is useful and applicable in the Industry 4.0 environment.

Keywords: Enterprise Architecture, Automated, Enterprise Modeling, Abstraction Hierarchy, Industry 4.0, AMA4EA.

* Corresponding author

© 2019 Marco Nardello et al. This is an open access article licensed under the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reference: M. Nardello, S. Han, C. Møller, and J. Gøtze, “Automated Modeling with Abstraction for Enterprise Architecture (AMA4EA): Business Process Model Automation in an Industry 4.0 Laboratory,” *Complex Systems Informatics and Modeling Quarterly*, CSIMQ, no. 19, pp. 42–59, 2019. Available: <https://doi.org/10.7250/csimq.2019-19.03>

Additional information. Author ORCID iD: M. Nardello – <https://orcid.org/0000-0003-0387-0047>, S. Han – <https://orcid.org/0000-0001-8477-887X>, C. Møller – <https://orcid.org/0000-0003-0251-3419>, and J. Gøtze – <https://orcid.org/0000-0003-3517-6180>. PII S225599221900111X. Received: 6 December 2018. Accepted: 17 May 2019. Prepublished: 12 March 2019. Available in Issue 19 of CSIMQ: 19 July 2019.

1 Introduction

The paradigm shift towards Industry 4.0 poses challenges for companies. Industry 4.0 introduces interconnected and intelligent systems to realize self-regulating production environments where people, equipment and products communicate with one another [1, p. 122]. For instance, manufacturing companies implement systems like the Microsoft Azure cloud platform to leverage data collected by Internet-of-Things (IoT) and cyber physical systems. In addition, new Enterprise Systems (ES) will be introduced to replace multiple legacy systems (e.g. Manufacturing Execution Systems) in order to collect more business data from production processes. Therefore, the paradigm shift towards Industry 4.0 requires companies to develop the ability to handle business data efficiently. Business data are related to the core business of an enterprise, for instance production processes and performance measurements. Although companies are able to gather large amounts of these business data, they have challenges in understanding them and extracting value from them. One of the reasons is that the data gathered is too detailed in different ES, and too difficult for stakeholders to understand. The current industry practice for coping with this problem is largely dependent on people to analyze and abstract these data manually. This practice is not able to meet the challenges in the business transformation. Recently, Reference Architecture Model Industry 4.0 (RAMI4.0) and Industrial Internet Reference Architecture (IIRA) have been proposed to support companies to automatically manage data, for instance, by implementing a uniform structural metadata for the business data [2]. However, these new initiatives do not provide any automated methods to implement a uniform structural metadata. Thus, a new automated method for Enterprise Architecture is needed to manage and abstract data in order to generate higher business value. Also, new Enterprise Architecture research is required for the handling of the problems and challenges in the Industry 4.0 environment [3].

Enterprise Architecture (EA) is the discipline that provides a holistic view of an enterprise [4, p. 7]. EA is concerned with modeling the business, application and technology aspects of an enterprise [4]. It is common practice for enterprise architects to develop EA models manually [5]. This manual approach makes EA modeling inefficient, time-consuming and error-prone [5]–[8]. Therefore, “manual documentation activities pose one of the biggest challenges to EA management” [9, p. 397]. It is critical to reduce the manual activities in EA modeling [10]. To address these limitations, researchers have started to develop automated EA documentation and modeling methods. However, the state-of-the-art of automated EA documentation and modeling methods is concerned with application and technology aspects and neglects business aspects. Therefore, the existing automated methods do not adequately address the industrial challenges. Moreover, no method has addressed the challenge of abstracting the excessively detailed data from ESs to be relevant for EA models [6], [9]. In addition, explanations of the existing automated methods are insufficient to allow them to be replicated. The algorithms included in the existing methods are described superficially and no article mentions how the EA models were created and how elements in an EA repository were positioned in the EA model.

In this article, we aim to deal with the limitations of automated EA modeling methods and address the industry challenges related to Industry 4.0. Therefore, we propose Automated Modeling with Abstraction for Enterprise Architecture, AMA4EA. AMA4EA is a method that automatically abstracts detailed data from ESs to concepts. These concepts are further aggregated to be relevant information for EA models, for instance, an EA model related to the business aspect. Furthermore, AMA4EA automatically instantiates the relevant information in elements in an EA repository and creates EA models with these elements. We tested our method at the Industry 4.0 laboratory at Aalborg University, Denmark. The laboratory provides industrial ESs that manage an end-to-end production process. AMA4EA abstracted detailed data from the Enterprise Resource Planning (ERP) system and MES to be relevant for an EA model at a business level. AMA4EA automatically created a business process model that provided a

simplified visualization of production process data. The EA model communicated this business data in an easily understandable way to the laboratory manager.

The article is an extension of our previous work [11] and it is structured as follows. Section 2 presents the related work and the research gap. Section 3 describes AMA4EA. Section 4 presents the demonstration of AMA4EA and its evaluation. The last two sections discuss the findings and conclude the article.

2 Related Work

2.1 Enterprise Architecture and Enterprise Architecture Modeling

Enterprise Architecture (EA) is a discipline that supports enterprises in managing the complexity of changes impacting the whole enterprise. Different definitions of EA exist [12]. In this article, EA is defined as a discipline that aims to “effectively implement the overall enterprise strategy by designing the various enterprise facets (governance structures, IT capabilities, remuneration policies, work design, and so on) to maximize coherency between them and minimize contradictions” [12].

EA is supported by an EA repository to manage EA elements (e.g. activities, relationships, business actors, business goals, etc.). The structural metadata of EA elements defines what data is stored and in which fields of the element in an EA repository. A uniform structural metadata of EA elements exists when all the elements of a certain type (e.g. activities) store data in the same fields. An EA repository visually represents EA elements in EA models.

EA modeling is at the core of the EA discipline. EA modeling is typically structured in three levels of abstraction – business, application and technology – that are defined in [4, p. 76] as follows:

1. *“The business layer offers products and services to external customers, which are realized in the organization by business processes (performed by business actors or roles).*
2. *The application layer supports the business layer with application services which are realized by (software) application components.*
3. *The technology layer offers infrastructural services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication devices and system software.”*

EA modeling is organized in five main activities [4]. The first activity is *establishing the purpose, scope and concepts* of an EA model. Each EA model has a purpose, for instance to provide insight into processes or enable business–IT alignment. The purpose restricts the part of reality that will be modeled. The second activity is the *selection of the viewpoints* to create an EA model to fulfill the requirements of stakeholders. This includes selecting concepts and defining the relationship between concepts represented in an EA model.

The third activity is *creating and structuring the model* and it involves five actions. When previous documentation is available it is important to (1) “check the validity of any existing models, and (2) incorporate the information on an appropriate level of abstraction” [4, p. 145]. When the information available is not sufficient, (3) additional information is gathered. Based on this information, (4) EA models are created. This involves (5) structuring an EA model by positioning elements in an EA model to “make a model as self-explanatory as possible” [4, p. 157].

Based on the requirements of the stakeholders, the fourth activity, *visualizing the model*, focuses on presenting an EA model in an appropriate way. The visualization can use specific modeling notations or templates. The fifth activity includes the *use* of the representation of the model to communicate with the stakeholders, and iterative *maintenance of the EA model* to keep the EA model up to date and in line with the stakeholders’ requirements.

2.2 Abstraction in Enterprise Architecture Modeling

EA modeling creates EA models abstracted from reality [4]. Various types of abstraction have been presented in the EA literature. Lankhorst et al. [4] and Pulkkinen [13] defined different *abstraction levels* (Pulkkinen refers to them as dimensions) of an EA model. An EA model can be at a business, application, or technology abstraction level [4]. The choice of the abstraction level of an EA model is based on the stakeholders using the EA model and the purpose of the EA model.

Furthermore, Lankhorst et al. [4] and Arbab et al. [14] distinguished between abstraction from aspects and abstraction from properties. The *abstraction from aspects* relates to the selection of a perspective to be used when modeling an entity, for instance, modeling only the functional aspects of a manufacturing equipment and not its structural aspects. Following the abstraction from aspects, the *abstraction from properties* limits the properties considered when modeling an entity. In the remainder of the article, we will refer to properties also as fields. This abstraction selects the properties that are modeled from all the available properties pertaining to an aspect (e.g. business, application or technology). When modeling using existing data that are included in an EA model, it is required to relate the fields of the data source with the fields of the system receiving the data.

An example of this is performing abstraction from properties by extracting only a subset of the fields available in an ES and relating them one-to-one with fields of an EA element in an EA repository. This type of alignment is also called “mapping” [5], [15].

Arbab et al. [14] presented the *generalization* abstraction. Generalization relates data to concepts. For instance, the generalization of a manufacturing operation with ID 10 in an ERP system consists in relating this data to the drilling concept.

The *hierarchical abstraction* aggregates and combines concepts [7], [16]. It leverages an abstraction hierarchy to organize various concepts on different hierarchy levels and relate them with each other. Although different types of hierarchical abstraction exist, for instance time or organizational abstractions, only two are relevant for this article. From an analytical approach, *structural abstraction* is concerned with “the concept of part–whole physical decomposition” where the hierarchical levels of a system help to classify the physical structure or granularity of a system [16]. It is used to break down the structure of a system into its components in order to facilitate their analysis. It is applied iteratively until the system is decomposed to the level of detail required. Examples of industrial standards of structural abstraction include the EA Enterprise-Domain-System decomposition [13] and the equipment hierarchy in the IEC 62264 / ISA-95 standard [17], which is also included in RAMI4.0 [2]. Another hierarchical abstraction is *functional abstraction*, which relates to a multi-level hierarchy for describing the functional structure of a domain [16]. It “is defined as many-to-many structural goal–means relationships between adjacent levels” [16, p. 579]. The higher level of a particular function “explains the reasons why the function is designed, whereas its lower level illustrates how the function is actually implemented” [16, p. 579]. The five abstraction levels of a functional hierarchy are listed from the top level to the bottom level [16, pp. 577, 579]:

1. *Functional Purpose (FP)* “represents the ultimate functions that a work domain should fulfil” [16].
2. *Abstract Function (AF)* “describes how and in what priority Generalized Function-level functions work together to realize FP-level functions” [16].
3. *Generalized Function (GF)* defines the more concrete functions that “should be implemented to realize the ultimate functions identified at the FP level. They are sometimes called as purpose-related functions” [16].
4. *Physical Function (PF)* “represents how GF-level functions are implemented more concretely or physically. PF describes physically implemented functions (e.g. electrical,

mechanical, and chemical functions) that can be identified from the physical states of a component or physical objects” [16].

5. *Physical form (P)* represents “the visible appearance and form of components and devices designed in a work domain” [16].

2.3 Automated Enterprise Architecture Documentation and Modeling Methods

There are two automated EA methods. One is automated EA documentation that relates to the collection of new data necessary for creating an EA model. The other is automated EA modeling that includes an automated EA documentation method and instantiates the data collected in elements in an EA repository. Automated EA modeling also positions these elements in an EA model. We analyze each method with regard to the five actions related to the creating and structuring of EA models (see Section 2.1). Furthermore, we analyze the methods with regard to the abstraction types (see Section 2.2) and summarize our analysis in Table 1.

Farwick et al. [9] developed a method for automated EA documentation. The method included four techniques [9]: (1) task-based reminders, (2) automated structured data collection, (3) external event triggers, and (4) internal model event triggers. Among these techniques, three of them (1, 3, and 4) notified the users that an EA model needed to be revised. The automated structured data collection technique focused on the collection of structured data. The purpose of this technique was “the reuse of external structured data sources into the EA model in order to reduce or even eliminate the manual data collection effort for specific model elements in the repository” [9, p. 408]. The automated structured data collection technique was divided into three main activities: (1) import data from the data source to the EA repository, (2) assess if manual intervention is required to import data or if it can be fully automated; (3) instantiate elements in an EA repository, either manually by an enterprise architect or automatically. Farwick et al. [9] mentioned neither how the automated activities are executed nor how elements were automatically instantiated in the EA repository and the EA models automatically created.

The method gathered data and created EA models automatically. However, Farwick et al. [9] did not explain how the method created EA models and how elements were positioned in the EA models. As shown in Table 1, the method was useful exclusively for EA models in the application and technology abstraction levels. Furthermore, the method performed the abstraction from properties by eliminating irrelevant details from the data collected. The method also mapped the fields of the data source to the fields of the element in the EA repository. Other types of abstraction were not mentioned.

The automated EA modeling method was a “method for automatic generation of EA models with respect to the complex IT architectures of enterprises” [5, p. 839]. Buschle et al. [18] outlined the method and other researchers further applied and developed it [5], [15], [19]. This automated EA modeling method collected data using predominantly network scanner applications [18]. The method instantiated the data collected in elements of ArchiMate modeling notation in the EA Analysis Tool (EAAT) repository. The automated EA modeling method was divided into two main activities [5]: (1) metamodel alignment and (2) EA model generation. The first activity focused on mapping fields from the data source (e.g. network scanner application) to the fields of the elements of ArchiMate (see abstraction from properties in Section 2.2). The second activity instantiated the data gathered into elements in EAAT. This activity included the automatic creation of an EA model with these elements.

Table 1. Abstraction types in EA automated methods

Abstraction types	Farwick et al. [9]	Buschle et al. [18]	Holm et al. [5]	Välja et al. [15]	Välja et al. [19]
Abstraction levels					
Business level	N/A	N/A	(X)*	N/A	N/A
Application level	X	X	X	N/A	X
Technology level	X	X	X	X	X
Abstraction from aspects	N/A	N/A	X	N/A	N/A
Abstraction from properties	X	X	X	X	X
Generalization	X	N/A	X	N/A	X
Hierarchical					
Structural abstraction	N/A	N/A	N/A	N/A	N/A
Functional abstraction	N/A	N/A	N/A	N/A	N/A

Although the method did not analyze existing documentation, it automated the gathering of new information, the creation of an EA model, and the structuring of an EA model. However, while a logical explanation was provided for the first activity, a full logical explanation in support of the instantiation of elements in EAAT and positioning of elements in an EA model was not given. As presented in Table 1, the method [5], [15], [18], [19] covered the automation of EA models related to the application and technology abstraction levels. Examples and cases related to the abstraction from aspects and properties, and generalization abstraction were also included. However, the method presented elements at the business abstraction level only in [5]. The EA models created focused on application and technology abstraction levels and included one single element from the business abstraction level, “Business actor” in ArchiMate modeling notation [5]. Moreover, the authors realized that their contribution was not reliable since “the accuracy of Business actors could not be evaluated” [5]. Therefore, this method is not adequate for creating an EA model at the business abstraction level. No evidence was found that this method implemented an abstraction hierarchy.

In summary, current research on automated EA documentation and modeling has three limitations. The first limitation is that existing methods insufficiently cover the business abstraction level. Although [5] attempted to include the “Business actor” element pertaining to the business abstraction level, the authors realized that their contribution was not reliable [5]. Existing automated EA documentation and modeling methods do not adequately cover business aspects and therefore cannot support stakeholders in managing business data.

The second limitation is concerned with the fact that data from ESs is too detailed to be useful for creating EA models. Since Hauder et al.’s survey [6] reported this as the main challenge for automated EA documentation, no method has fully addressed this challenge. This is still an open research area, as highlighted by [9]. Also in this case, existing automated EA documentation and modeling methods fall short in supporting stakeholders to understand the detailed business data from ESs.

The third limitation is the superficial explanation of existing automated EA documentation and modeling methods. Logical explanation of the automated methods is largely missing. This limitation inhibits stakeholders from implementing automated EA documentation and modeling methods to apply a uniform structural metadata for business data. The expected increase in the amount of data managed by future ESs will increase the need for uniform structural metadata for business data still further.

Process mining approaches to generate process models are related to the automated EA documentation and modeling presented above. These approaches [20], [21] use input sources (e.g. log data) that are different from the type of data sources used in AMA4EA.

3 Automated Modeling with Abstraction for Enterprise Architecture (AMA4EA)

AMA4EA is a method to automatically abstract detailed data from ESs to concepts. The abstraction is achieved through the use of the AMA4EA environment. The AMA4EA environment is a system that abstracts data, for instance from ESs, following predefined abstraction hierarchies. AMA4EA also instantiates the relevant information in an EA repository and creates EA models automatically.

AMA4EA requires four roles. A stakeholder (S) initiates the modeling process and specifies his requirements. An enterprise architect (A) manages the execution of AMA4EA. The architect collaborates with the stakeholder to define the purpose, scope and concepts of the desired EA model. A data source manager (DSM) is responsible for the ES that provides data for AMA4EA. The DSM makes available the information about the ES, ensures that the system has the interfaces to extract data for AMA4EA and guarantees the data quality. A subject-matter expert (SME) has knowledge on particular subjects or technologies. An SME, in some cases, may collaborate with the architect to define an abstraction hierarchy. If required, an SME manually abstracts data from the ES.

AMA4EA requires data in the ES to be available and reliable, otherwise the data need to be corrected before the application of AMA4EA. Furthermore, data extracted from the ES must be in a format readable by the AMA4EA environment and EA repository. Our method is divided into preparation and execution phases.

In the preparation phase of AMA4EA, first a stakeholder and an architect define the desired EA model. Second, the DSM and architect identify relevant data sources. Third, the architect identifies the abstraction hierarchy. Finally, the architect sets up the AMA4EA environment. The four activities in the preparation phase are described in more detail.

1. *Define the desired EA model.* A stakeholder and an architect collaboratively define the purpose, scope and concepts of the desired EA model. They also decide the abstraction level (e.g. business, application, technology) and abstraction aspect (e.g. functional, structural) of the desired EA model (see Section 2.2 for more details). The architect selects the desired type of EA model (e.g. business process model, product architecture model, strategy model) and the modeling notation (e.g. ArchiMate, BPMN, UML, industry specific).
2. *Identify data sources.* The DSM and architect collaboratively identify which ES stores the data related to the concepts of the desired EA model. They specify the ES and location of the relevant data and the structural metadata of these data. Furthermore, they specify which interfaces will be used for extracting data from the ES (e.g. database query, APIs). In this activity, abstraction is performed by selecting certain fields from all the available fields in the ESs.
3. *Identify abstraction hierarchy.* The architect chooses an abstraction hierarchy in line with the purpose, scope and concepts of the desired EA model. If a suitable abstraction hierarchy is not available in the AMA4EA environment or an EA repository, the architect and SME search for one, for instance from industrial standards, and import this to both systems. If no appropriate abstraction hierarchy is found, they define a new abstraction hierarchy by specifying the concepts to be included, the organization of these concepts on different hierarchy levels and the relationship between these concepts. Subsequently, the architect imports the abstraction hierarchy to the AMA4EA environment and EA repository.
4. *Set up AMA4EA environment.* In this activity two tasks are completed. First, the architect creates in the AMA4EA environment a dedicated data storage area to import data with the same structural metadata as those in the ESs. This ensures that the AMA4EA environment

can import data from ESs automatically. Second, the architect creates the structural metadata of the “main” interface of the AMA4EA environment. This interface is structured in three sections. The first one contains fields of the structural metadata from the dedicated data storage area. The second one includes information for performing generalization and hierarchical abstraction. This information includes the concepts and relationships specified in the abstraction hierarchy. The third one contains the necessary information for performing abstraction from properties. The information includes a list of fields from the elements in an EA repository.

In the execution phase, the AMA4EA extracts data from ESs to the AMA4EA environment. An ES is used by enterprises to run their business. The AMA4EA environment abstracts the data automatically by using an *abstraction algorithm*. The *abstraction algorithm* queries the AMA4EA environment for previous manual abstractions and applies them to the extracted data. If the *abstraction algorithm* cannot abstract some data, the AMA4EA environment supports an SME to manually abstract the remaining data. These new manual abstractions are then saved by the *abstraction algorithm* to the AMA4EA environment, which will automatically abstract the same data in the future. As a result, the AMA4EA environment can abstract automatically different types of data extracted from various ESs. Data with abstraction is imported by an EA repository. At first, the *instantiate and position algorithm* automatically instantiates elements containing these data with abstraction in the EA repository. Then, the same algorithm automatically presents these elements in the desired EA model. The three activities in the execution phase are described in more detail.

1. *Extract data from ES to AMA4EA environment.* The DSM extracts data from an ES. The automatic execution of this activity depends on how frequently a stakeholder requires an updated version of the desired EA model or the number of EA models required. If beneficial, the DSM may extract the data automatically by using enterprise integration software (e.g. middleware, extract-transform-load software, robotic process automation) in order to reduce the manual effort required for extracting and transferring data. The software can also control the data quality to increase the reliability of the data.

The data from the ES will be automatically imported by using the *data import algorithm* in the AMA4EA environment. This algorithm parses data and stores them in a dedicated data storage area in the AMA4EA environment that has the same structural metadata of the ES.

2. *Abstract data in AMA4EA environment.* The *abstraction algorithm* queries the AMA4EA environment for previous manual abstractions and applies to the data the three types of abstraction specified in the last activity of the preparation phase – abstraction from properties, generalization and hierarchical abstraction. The *abstraction algorithm* also assesses if all the data have been abstracted. If some of the data lack abstraction, this algorithm notifies the architect who shares the AMA4EA environment with the SME and requires the SME to manually abstract the remaining data. The SME abstracts data by mapping the fields from the ES to the corresponding fields of the elements in the EA repository. Moreover, the SME relates the data to the concepts in the abstraction hierarchy. The *abstraction algorithm* saves the manual abstractions in the AMA4EA environment.
3. *Create the desired EA model.* The *instantiate and position algorithm* imports data with abstraction from the AMA4EA environment to the EA repository. This algorithm parses data with abstraction and instantiates new elements in the EA repository storing additional information in the elements’ fields. The *instantiate and position algorithm* creates the desired EA model and adds these instantiated elements in the desired EA model. During this activity this algorithm structures the EA model by determining the position of elements in the desired EA model. Different algorithms can be used to determine the position of elements, from simple ones that position elements in a sequential order (e.g. left to right, or top to bottom), to more advanced ones like genetic algorithms [4, p. 309].

4 Demonstration of AMA4EA in an Industry 4.0 Laboratory

The demonstration of AMA4EA focuses on EA models pertaining to the business abstraction level because this abstraction level is under-researched (see Section 2.3). AMA4EA was demonstrated in the Smart Production Laboratory at Aalborg University [22] in 2018 (see Figure 1). The Smart Production Laboratory “is a Learning Factory [22] and it includes a fully automated small production line integrating and demonstrating various Industry 4.0 concepts and technologies” [23, p. 28]. It replicates industrial environments by including state-of-the-art production technologies in terms of both production equipment and software applications [22]. Students, researchers and practitioners collaborate in the development of new technologies and solutions for the manufacturing industry. In this demonstration, we focused on the simplified “mobile phone” shown in Figure 1. This mobile phone is composed of four parts: back-cover, top-cover, circuit board, and fuses. The production process of this mobile phone involves assembly, drilling and inspection activities.

In this demonstration of AMA4EA, we abstracted data from the laboratory’s ERP system and MES to production concepts (e.g. lay component on top, drilling, check properties). Additionally, we implemented the “Production Process Classification” [24] as an abstraction hierarchy to further abstract production concepts. The Production Process Classification used originated from industrial standards. Furthermore, AMA4EA instantiated automatically the concepts in the classification in QualiWare’s EA repository and created a business process model.

Since this was the first demonstration of AMA4EA, we focused on the manual abstraction activities necessary for the automated execution of AMA4EA. The stakeholder was the laboratory manager and the first author acted as architect, DSM and SME. Furthermore, we used the FESTO MES (MES4[†]) and SAP ERP system (version 6.0) as ESs, QualiWare’s EA repository [25] (QLM version 6.6) and a Microsoft Excel file as the AMA4EA environment. Figure 2 presents screenshots of these systems.

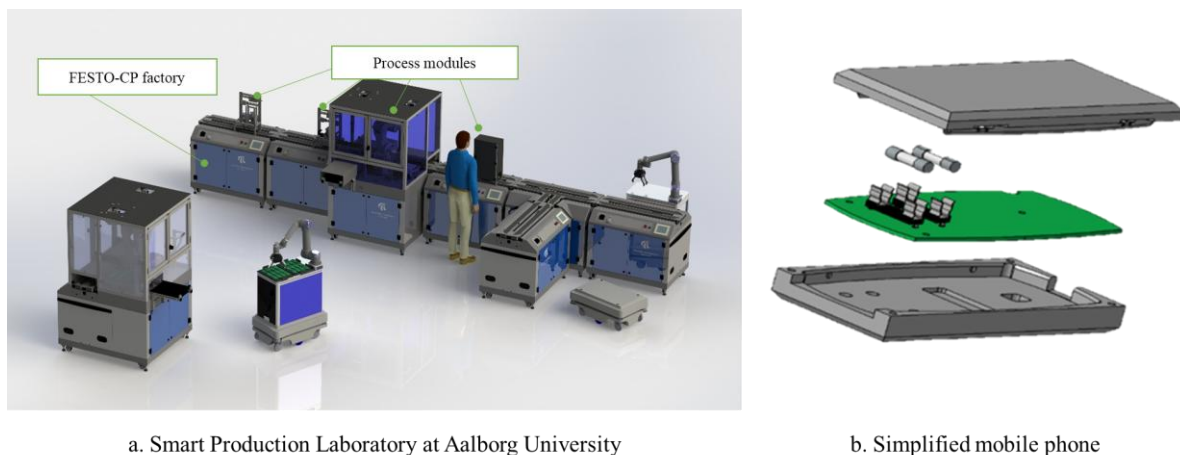


Figure 1. Overview of the Smart Production Lab [23] and simplified “mobile phone”

4.1 Preparation Phase of the Demonstration

The *define the desired EA model* activity started with a meeting between the stakeholder and the first author to discuss the desired EA model. The purpose of the desired EA model was to provide a visual representation of the production process that the stakeholder could use to

[†] <https://www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/mes4.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC41ODUuNTM3NjA> [Last accessed on 15.07.2019.]

introduce the laboratory to master degree students. The scope of the model was limited to the production process of the mobile phone. The desired EA model included activities of the production process and the relationship between these activities to show the sequence flow. The abstraction level of the desired EA model was the business abstraction level. The desired EA model focused on the functional aspect of the production process. The type of desired EA model chosen was a business process model. The modeling notation for the desired EA model was an industry specific modeling notation for production processes, namely Production Process Classification [24]. An example of this modeling notation is presented in Figure 3.

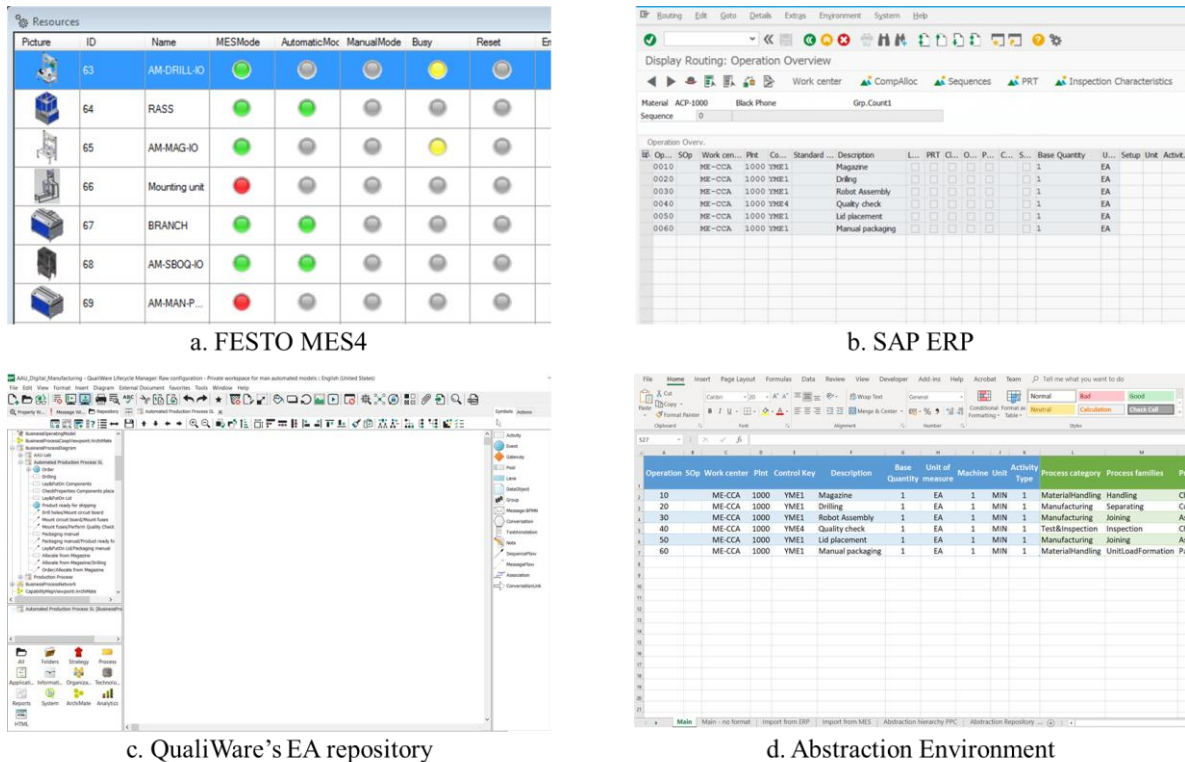


Figure 2. Enterprise Systems and EA repository used in the demonstration

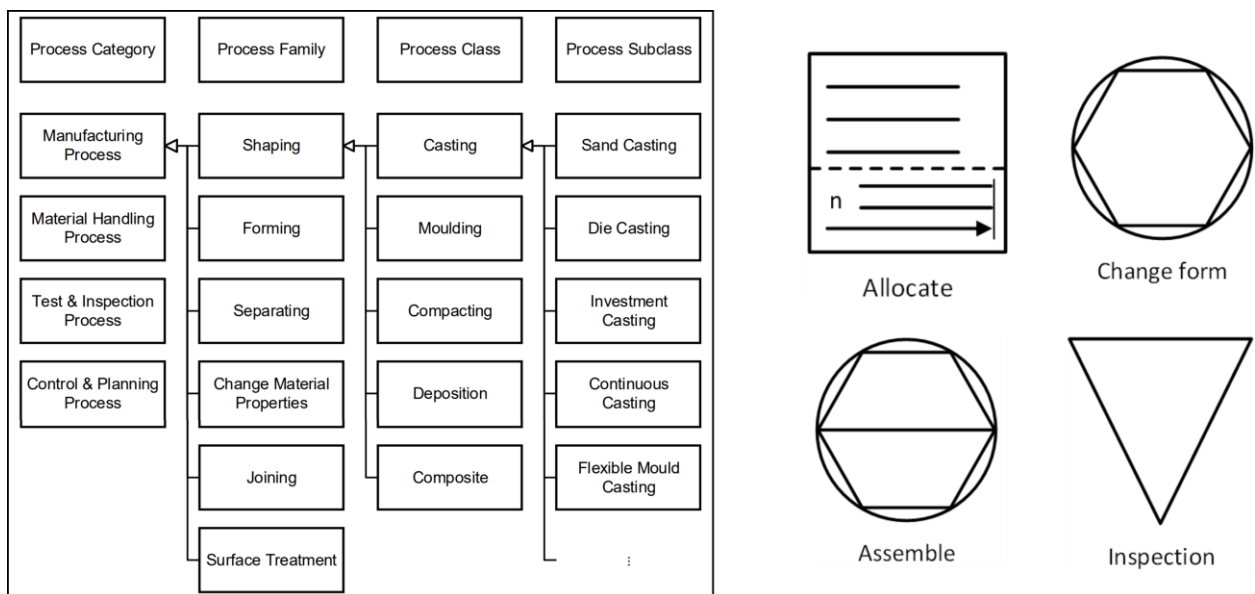


Figure 3. Example of Production Process Classification [24] and modeling notation used in the demonstration

The *identify data sources* activity was performed by selecting the two ESs managing production processes at the laboratory, namely SAP ERP and FESTO MES4. The data in the ESs were analyzed and the tables containing the data related to the production process of the mobile phone were identified. In the SAP ERP system, the “Routing: Operations Overview” table was selected since it contains[‡]: “operations and sub-operation, which describe the process steps in the routing; the work center where they are carried out; and a short description of the process step”. The location of the “Routing: Operations Overview” table in SAP ERP is the following: SAP Menu, Logistics, Production, Master Data, Routings, Routings, Standard Routings, Display. In order to create the desired EA model, abstraction from properties was performed by selecting the 12 fields from the “Routing: Operations Overview” table (see below).

Operation, SOp, Work center, Plnt, Control Key, Standard text key, Description, Base Quantity, Unit of measure for Op., Machine, Unit, Activity Type.

In FESTO MES4, the “tblStepDef” table was selected because it contained the steps of the production process and their sequence. The structural metadata of this table included the fields relevant for the purpose of the desired EA model (see below).

WPNo, StepNo, Description, OpNo, NextStepNo, FirstStep, ErrorStepNo, NewPNo, OpNoType, ResourceID, TransportTime, ErrorStep, SqlWrite.

The Operation field from the “Routing: Operations Overview” table and StepNo field from the “tblStepDef” table corresponded semantically. Therefore, these fields were the keys to relate data between the two tables. The structural metadata of these tables were stored in a text document. This demonstration did not require dedicated interfaces for extracting data from the ESs since data could be extracted by simply copying the data from the ESs to the AMA4EA environment.

The *identify abstraction hierarchy* activity was executed by searching for an abstraction hierarchy. Because the laboratory did not have an available one in support of creating the desired EA model, the abstraction hierarchy of the Production Process Classification [24] was chosen (see example in Figure 3). Table 2 shows how the classification is mapped to the functional abstraction hierarchy [16].

Table 2. Mapping of the Production Process Classification [24] to the functional abstraction hierarchy [16] used in the demonstration

Levels of the functional abstraction hierarchy [16]	Levels of the Production Process Classification [24]	Examples from the demonstration
Functional Purpose (FP)	N/A	Produce a mobile phone
Abstract Function (AF)	Production Category	1. Manufacturing 2. Material handling 3. Test and inspection 4. Control and planning
Generalized Function (GF)	Process Family & Process Class	Manufacturing → Assembly, Fastening
Physical Function (PF)	Process Subclass	Fastening → Bolting, Riveting, Screwing
Physical form (P)	N/A	Screwing → Screwing machine

The Production Process Classification was in line with the purpose, scope and concepts of the desired EA model. In addition, the four levels of the Production Process Classification could be

[‡] https://help.sap.com/saphelp_46c/helpdata/en/03/bb1d06a6e811d189010000e8323492/frameset.htm
[Last accessed on 15.07.2019.]

matched with the five levels of the functional abstraction hierarchy [16]. At the functional purpose (FP) level, the purpose of the laboratory was to produce a mobile phone. To fulfil this purpose, four abstract functions were required: (1) manufacturing, (2) material handling, (3) test and inspection, and (4) control and planning. These abstract functions (AF) were more concretely implemented by generalized functions. These generalized functions (GF) were specified in the Production Process Classification in two levels, process family and process class. For instance, the manufacturing abstract function was implemented using assembly and fastening concepts. Generalized functions were even physically implemented by concepts at the physical function (PF) level. For instance, the fastening general function was physically executed as either bolting, riveting, or screwing. Physical functions were directly related to the physical form (P) level as a specific production activity or equipment existing in reality.

The concepts and relationships of the Production Process Classification were imported into a dedicated sheet in the AMA4EA environment. The text file defining the metadata of the “Activity” element in QualiWare’s EA repository was extended. First, four fields were added, one for each level of the Production Process Classification – process category, process families, process classes, and process subclasses (see Figure 4). Second, in the same text file the concepts and relationships between them were specified. This will enable the automated abstraction of data in the execution phase.

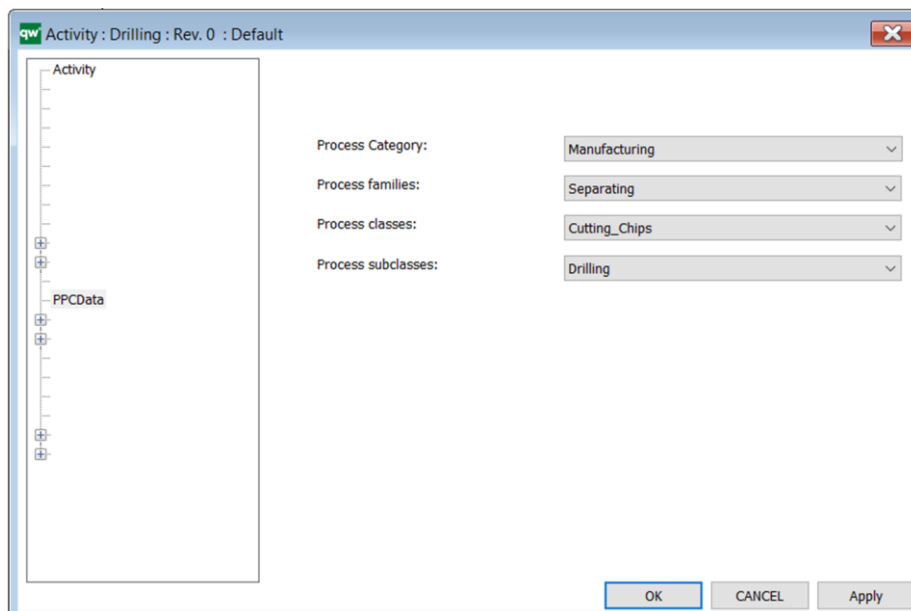


Figure 4. Extension of the meta-model of the Activity element in QualiWare’s EA repository. Image manipulated due to confidentiality.

The *set up AMA4EA environment* activity is also key to enable automated abstraction in the execution phase. A data storage area was created in the AMA4EA environment that included: one sheet for importing data from SAP ERP, and one for importing data from FESTO MES4. In each sheet, the structural metadata of the specific ES was replicated. In the “import from ERP” sheet, the fields selected in the preparation phase were specified. In the “import from MES” sheet, all the fields in FESTO MES4 were specified. The “main” sheet was created in the AMA4EA environment. This “main” sheet is where the relevant data from the data storage area will be abstracted. As specified in activity 4 of the preparation phase, information to enable abstraction from properties, generalization and hierarchical abstraction was copied in the “main” sheet. The structural metadata of the “main” sheet is the following:

- one column for each field in “import from ERP” sheet (see Figure 5b);
- followed on the right by the “Next step” column for the `NextStepNo` field from the “import from MES” sheet;

- followed on the right by one column for each of the levels of the Production Process Classification;
- followed on the right by an extra column to specify the name displayed in the EA model (see Figure 5c). This last column is required to distinguish between abstracted elements that would otherwise not be distinguishable.

The mapping between the fields from the ESs and the fields of the elements in the EA repository was excluded from the AMA4EA environment in this demonstration. Instead, this mapping was included in the *instantiate and position algorithm* that created the desired EA model in the execution phase.

4.2 Execution Phase of the Demonstration

The *extract data from ES to AMA4EA environment* activity was manually executed. For the data in the “Routing: Operations Overview” table of SAP ERP, the data of the fields selected in the *identify data sources* activity of the preparation phase were copied to the “import from ERP” sheet in the AMA4EA environment. Figure 5a shows the data in SAP ERP. For the data in the “tblStepDef” table of FESTO MES4, all the data in the table were copied to the “import from MES” sheet in the AMA4EA environment. Afterwards, data from these sheets was automatically copied to the “main” sheet using Excel formulas. An Excel formula is “an expression which calculates the value of a cell”[§] in an Excel file. The implementation of the *data import algorithm* was not necessary since the data were imported automatically using Excel formulas.

In this demonstration, AMA4EA was executed for the first time in the laboratory. Therefore the AMA4EA environment did not contain previous abstractions and for this reason we did not execute the *abstraction algorithm*. The *abstract data in AMA4EA environment* activity was performed manually. The data in the “main” sheet were analyzed and concepts from the Production Process Classification were assigned based on the data provided (see Figure 5b and Figure 5c). Operations 30 “Robot Assembly” and 50 “Lid Placement” were both abstracted to the “Lay&PutOn” production concept. For this reason their label and symbol should have been the same in the desired EA model. To allow the stakeholder to identify which “Lay&PutOn” concept related to operation 30 and which one to operation 50, the labels were extended in the last column of the “main” AMA4EA environment respectively to “Lay&PutOn Components” and “Lay&PutOn Lid”. Figure 5d provides another visualization of the results of the abstraction activity.

The *instantiate and position algorithm* imported data with abstraction from the AMA4EA environment to QualiWare’s EA repository. In addition, it instantiated “Activity” and “SequenceFlow” elements in QualiWare’s EA repository. The “Activity” elements stored abstraction information in the dedicated fields that were developed in the *identify abstraction hierarchy* activity in the preparation phase. The *instantiate and position algorithm* also determined the position of the elements in the business process model and their sequence flow, as shown in Figure 5e. The pseudo code is shown in Algorithm 1.

Algorithm 1. Pseudocode of the instantiate and position algorithm used in the demonstration

```

Create Business Process Diagram in the EA repository
  Ask user to insert the name of the diagram
Open an Excel file contained in a specific folder
  Access the "main" sheet
  For each row in the sheet
    Create Activity and SequenceFlow elements in the EA repository
    Assign fields from the "main" sheet to the fields in Activity element
    Assign position (from left to right)

```

[§] <https://www.excel-easy.com/introduction/formulas-functions.html> [Last accessed on 15.07.2019.]

Instantiate Activity in the EA model
 Instantiate the SequenceFlow elements between Activities

End

Since the process modeled did not have branches, the position of elements was assigned in a sequential order from left to right. For more complex business process models, specific algorithms determine the position of elements [26], [27].

a. Extract data from SAP

Op...	SOp	Work cen...	Pnt	Co...	Standard ...	Description	L...	PRT	Cl...	O...	P...	C...	S...	Base Quantity	U...	Setup	Unit	Activt...	Machine	Unit	Activt...	La
0010		ME-CCA	1000	YME1		Magazine								1	EA				1		MIN	1
0020		ME-CCA	1000	YME1		Drilling								1	EA				1		MIN	1
0030		ME-CCA	1000	YME1		Robot Assembly								1	EA				1		MIN	1
0040		ME-CCA	1000	YME4		Quality check								1	EA				1		MIN	1
0050		ME-CCA	1000	YME1		Lid placement								1	EA				1		MIN	1
0060		ME-CCA	1000	YME1		Manual packaging								1	EA				1		MIN	1

b. Import data in AMA4EA environment

Operation	SOp	Work center	Pnt	Control Key	Description	Base Quantity	Unit of measure	Machine	Unit	Activity Type	Process category
10		ME-CCA	1000	YME1	Magazine	1	EA	1	MIN	1	
20		ME-CCA	1000	YME1	Drilling	1	EA	1	MIN	1	
30		ME-CCA	1000	YME1	Robot Assembly	1	EA	1	MIN	1	
40		ME-CCA	1000	YME4	Quality check	1	EA	1	MIN	1	
50		ME-CCA	1000	YME1	Lid placement	1	EA	1	MIN	1	
60		ME-CCA	1000	YME1	Manual packaging	1	EA	1	MIN	1	

c. Abstract data in AMA4EA environment

Operation	Process category	Process families	Process classes	Process subclasses	Name in QualiWare (the most detailed value available in the classification)
10	MaterialHandling	Handling	ChangeQuantities	Allocate	Allocate from Magazine
20	Manufacturing	Separating	Cutting_Chips	Drilling	Drilling
30	Manufacturing	Joining	Assembly	Lay&PutOn	Lay&PutOn Components
40	Test&Inspection	Inspection	CheckProperties	Lay&PutOn	CheckProperties Components placement
50	Manufacturing	Joining	Assembly	Lay&PutOn	Lay&PutOn Lid
60	MaterialHandling	UnitLoadFormation	Packaging	Lay&PutOn	Packaging manual

d. Results of the abstraction

Abstraction levels	Operation 10	Operation 20	Operation 30	...
Process category	MaterialHandling	Manufacturing	Manufacturing	
Process families	Handling	Separating	Joining	
Process classes	ChangeQuantities	Cutting_Chips	Assembly	
Process subclasses	Allocate	Drilling	Lay&PutOn	

e. Instantiate desired EA model

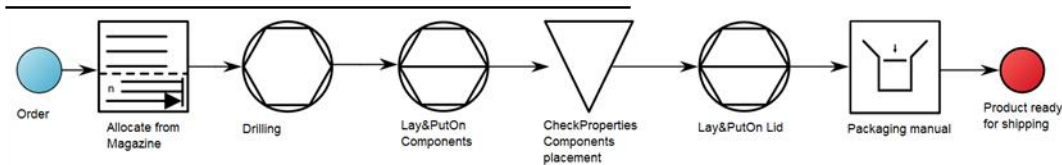


Figure 5. Demonstration of AMA4EA at the Industry 4.0 Laboratory at Aalborg University

4.3 Evaluation of AMA4EA Demonstration

We evaluated AMA4EA with the laboratory manager. We assessed the usability, utility and applicability of AMA4EA. Usability evaluates the degree to which the stakeholder was able to easily apply our method. Utility evaluates the degree to which AMA4EA was useful for automatically creating EA models. Applicability measures the degree to which our method can

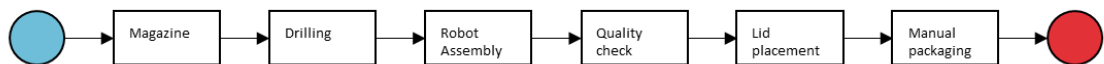
be applied in various industrial environments. In addition, we discussed future developments of AMA4EA.

The first author presented AMA4EA to the laboratory manager. The stakeholder confirmed that the method was usable since he could have executed AMA4EA himself. He also explained that AMA4EA was useful because it abstracted data from SAP ERP and FESTO MES4 to a more readable and understandable format. Furthermore, he said that AMA4EA is able to transform data in these systems from being understandable only by SMEs to be understandable by stakeholders who have not worked with these specific applications before (e.g. high level managers).

The author presented to the stakeholder two versions of the business process model, one that was generated using only the data from ESs and one that used data with abstraction (see Figure 6). The first model represented elements using standard “business activity” modeling notation and the naming available in SAP ERP (see Figure 5a and Figure 6a). The second model represented elements using the industrial modeling notation from the Production Process Classification and the naming resulting from the abstraction activities. The stakeholder expressed that the model automatically generated without abstraction (Figure 6a) was hard to read and of little use since it represented activities in an unclear way and the model included very little information. In contrast, he thought that “going from the raw model [Figure 6a] to the one with the classifications [Figure 6b] is a big step for the industry because they get much more information into the same model”.

Moreover, the stakeholder proactively presented a list of extensions to AMA4EA. He recommended creating a business process model for the different mobile phones since each mobile phone has its activities. Once the business process models for all mobile phones have been developed, he recommended creating a generic business process model that overlaps the activities in common in the different business process models and represents distinctively the activities specific to a type of mobile phone. “This would be valuable if you want to know if these ten products can be produced on the same production line”.

a. Without abstraction



b. With abstraction

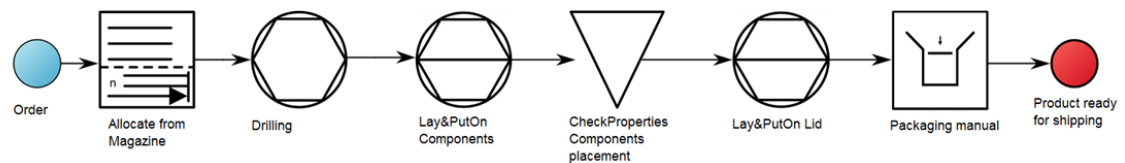


Figure 6. Model comparison between execution of the process with and without abstraction activity

5 Discussion

This article addresses the limitations of existing automated EA documentation and modeling methods. As summarized in Table 1, the three limitations were insufficient coverage of the business data, excessively detailed data from ESs for creating EA models, and the lack of logical explanation of the automated methods. This article proposes AMA4EA and demonstrates it in an Industry 4.0 laboratory. AMA4EA abstracted detailed data from ESs to production concepts. The abstraction was achieved by leveraging the Production Process Classification as a functional abstraction hierarchy. AMA4EA automatically instantiated production concepts in elements in

QualiWare's EA repository and automatically created a business process model presenting these elements.

AMA4EA addressed the limitations of automated EA documentation and modeling with the three contributions. First, in addition to covering abstraction at application and technology levels, AMA4EA can focus on the business abstraction level by creating a business process model automatically. Business process models are common EA models at the business abstraction level [4]. AMA4EA created a useful and simplified representation of business data automatically.

Second, AMA4EA introduced an abstraction hierarchy in the automated EA documentation and modeling methods. AMA4EA is unique in the implementation of an abstraction hierarchy that allowed detailed data to be abstracted from ESs to production concepts at the abstraction level required. This addresses the main challenge for automated EA documentation [6], [9]. It supported the stakeholder to implement a uniform structural metadata and to easily understand detailed business data by providing a simplified visualization.

Third, AMA4EA and its algorithms were logically explained in more detail compared to the existing automated EA documentation and modeling methods. In contrast with the existing methods, AMA4EA's logical explanation was provided. This include the demonstration of how the uniform structural metadata was implemented, how the EA models were created, and how the elements were positioned in the EA model was also outlined.

These contributions have three main implications for industry. First, AMA4EA can support stakeholders by automatically creating a useful simplified representation of business data. As the stakeholder said, AMA4EA's automated creation of EA models at the business abstraction level is a "big step for the industry because they get much more information into the same model". Second, AMA4EA can support stakeholders in understanding, managing and extracting value from detailed business data. The benefits of AMA4EA are expected to be increased when new ESs that can manage large amounts of data are used by companies. Third, the detailed explanation of AMA4EA and its demonstration enable companies to understand AMA4EA and apply it.

Our research has four main limitations. First, although the Industry 4.0 laboratory replicates industrial environments, the demonstration of AMA4EA included in this article used products and production processes that are less complex than most of the manufacturing processes in industry. Greater product and production process complexity will require further developments of the *instantiate and position algorithm*. Second, the *data import* and *abstraction algorithms* in AMA4EA were described in Section 3 but not demonstrated in Section 4. Third, the demonstration focused on one domain, namely production processes. Fourth, our demonstration assumes that data from ES is complete and correct.

Although AMA4EA was applied in the manufacturing industry in this demonstration, it could also be applied in other industries. The abstraction hierarchy (Production Process Classification) was the only industry specific element of AMA4EA. By changing the abstraction hierarchy, AMA4EA can be applied in other industries.

In order to address the limitations encountered in the demonstration of AMA4EA, future research will focus on the use of complex industrial data. We will also demonstrate the execution of the remaining algorithms in AMA4EA. We plan to apply AMA4EA in other domains, namely products and manufacturing equipment. In addition, we will gather more empirical evidence to demonstrate that AMA4EA can be applied also to create EA models on the application and technology levels. Finally, future research will further develop the method to operate in cases where data from the ES is either incomplete or incorrect, for instance, by implementing a closed loop mechanism that allows for the correction of data in the EA model and replicates the corrections in the ES.

6 Conclusion

The paradigm shift towards Industry 4.0 poses challenges for companies handling large amounts of data for creating business value. The state-of-the-art of EA research provides limited knowledge for addressing this problem. AMA4EA is a method to automatically abstract detailed data from ESs to concepts. AMA4EA is an innovative and novel method that contributes new knowledge to automated EA documentation and modeling methods. AMA4EA introduces an abstraction hierarchy in automated EA documentation and modeling methods, and creates EA models at various abstraction levels. AMA4EA also implements a uniform structural metadata of business data. It also generates a simplified and useful visualization of detailed business data which can be easily communicated to and understood by different stakeholders. Hence, AMA4EA addresses the limitations of the automatic EA documentation and modeling methods. The demonstration provides sufficient evidence that the method can be used in an Industry 4.0 environment. The implementation of AMA4EA is expected to create more business value for companies transforming towards Industry 4.0.

References

- [1] G. Kovács and S. Kot, “New Logistics and Production Trends as the Effect of Global Economy Changes,” *Polish Journal of Management Studies*, vol. 14, pp. 115–126, 2016.
- [2] International Electrotechnical Commission, “IEC PAS 63088:2017. Smart Manufacturing – Reference Architecture Model Industry 4.0 (RAMI4.0),” *IEC, Genf*, pp. 1–35, 2017.
- [3] F. Gampfer, A. Jürgens, M. Müller, and R. Buchkremer, “Past, Current and Future Trends in Enterprise Architecture—A View Beyond the Horizon,” *Computers in Industry*, vol. 100, pp. 70–84, 2018. Available: <https://doi.org/10.1016/j.compind.2018.03.006>
- [4] M. Lankhorst et al., *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, 2017. Available: <https://doi.org/10.1007/978-3-642-29651-2>
- [5] H. Holm, M. Buschle, R. Lagerström, and M. Ekstedt, “Automatic Data Collection for Enterprise Architecture Models,” *Software and Systems Modeling*, vol. 13, no. 2, pp. 825–841, 2014. Available: <https://doi.org/10.1007/s10270-012-0252-1>
- [6] M. Hauder, F. Matthes, and S. Roth, “Challenges for Automated Enterprise Architecture Documentation,” in *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*, Springer, 2012, pp. 21–39. Available: https://doi.org/10.1007/978-3-642-34163-2_2
- [7] M. Buschle, S. Grunow, F. Matthes, M. Ekstedt, M. Hauder, and S. Roth, “Automating Enterprise Architecture Documentation using an Enterprise Service Bus,” in *Americas Conference on Information Systems (AMCIS 2012)*, 2012, pp. 1–14.
- [8] M. Hauder, S. Roth, S. Pigat, and F. Matthes, “Tool Support for Conflict Resolution of Models for Automated Enterprise Architecture Documentation,” in *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, vol. 3, pp. 11–15, 2013.
- [9] M. Farwick, C. M. M. Schweda, R. Brey, and I. Hanschke, “A Situational Method for Semi-automated Enterprise Architecture Documentation,” *Software and Systems Modeling*, vol. 15, no. 2, pp. 397–426, 2016. Available: <https://doi.org/10.1007/s10270-014-0407-3>
- [10] R. Perez-Castillo, F. Ruiz-Gonzalez, M. Genero, and M. Piattini, “A Systematic Mapping Study on Enterprise Architecture Mining,” *Enterprise Information Systems*, vol. 13, no. 5, pp. 675–718, 2019. Available: <https://doi.org/10.1080/17517575.2019.1590859>
- [11] M. Nardello, C. Møller, and J. Götze, “Process Model Automation For Industry 4.0: Challenges For Automated Model Generation Based on Laboratory Experiments,” *CEUR Workshop Proceedings*, vol. 2218, pp. 201–216, 2018.
- [12] J. Lapalme, “Three Schools of Thought on Enterprise Architecture,” *IT Professional*, vol. 14, no. 6, pp. 37–43, 2012. Available: <https://doi.org/10.1109/MITP.2011.109>
- [13] M. Pulkkinen, “Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 8, no. C, pp. 1–9, 2006. Available: <https://doi.org/10.1109/HICSS.2006.447>

- [14] F. Arbab, F. de Boer, M. Bonsangue, M. Lankhorst, E. Proper, and L. van der Torre, "Integrating Architectural Models - Symbolic, Semantic and Subjective Models in Enterprise Architecture," *Enterprise Modelling and Information Systems Architectures*, vol. 2, pp. 40–47, 2007.
- [15] M. Välja, R. Lagerström, M. Ekstedt, and M. Korman, "A Requirements Based Approach for Automating Enterprise IT Architecture Modeling Using Multiple Data Sources," in *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*, pp. 79–87, 2015. Available: <https://doi.org/10.1109/EDOCW.2015.33>
- [16] D. H. Ham, "Modelling Work Domain Knowledge with the Combined Use of Abstraction Hierarchy and Living Systems Theory," *Cognition, Technology & Work*, vol. 17, no. 4, pp. 575–591, Nov. 2015. Available: <https://doi.org/10.1007/s10111-015-0338-y>
- [17] International Electrotechnical Commission, "IEC 62264-1 Enterprise-control system integration–Part 1: Models and terminology," *IEC, Geneva*, 2003.
- [18] M. Buschle, H. Holm, T. Sommestad, M. Ekstedt, and K. Shahzad, "A Tool for Automatic Enterprise Architecture Modeling," *CAiSE Forum (Selected Papers)*, pp. 1–15, 2011. Available: https://doi.org/10.1007/978-3-642-29749-6_1
- [19] M. Välja, M. Korman, R. Lagerström, U. Franke, and M. Ekstedt, "Automated Architecture Modeling for Enterprise Technology Management Using Principles from Data Fusion: A Security Analysis Case," in *Proceedings of PICMET '16: Technology Management for Social Innovation Automated*, pp. 14–22, 2016. Available: <https://doi.org/10.1109/PICMET.2016.7806662>
- [20] R. Conforti, M. Dumas, L. García-Bañuelos, and M. La Rosa, "BPMN Miner: Automated Discovery of BPMN Process Models with Hierarchical Structure," *Information Systems*, vol. 56, pp. 284–303, 2016. Available: <https://doi.org/10.1016/j.is.2015.07.004>
- [21] A. Augusto *et al.*, "Automated Discovery of Process Models from Event Logs: Review and Benchmark," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 686–705, 2019. Available: <https://doi.org/10.1109/TKDE.2018.2841877>
- [22] M. Nardello, O. Madsen, and C. Møller, "The Smart Production Laboratory: A Learning Factory for Industry 4.0 Concepts," in *CEUR Workshop Proceedings*, vol. 1898, p. 5, 2017.
- [23] M. Nardello, C. Møller, and J. Götze, "Organizational Learning Supported by Reference Architecture Models: Industry 4.0 Laboratory Study," *Complex Systems Informatics and Modeling Quarterly*, no. 12, pp. 22–38, 2017. Available: <https://doi.org/10.7250/csimq.2017-12.02>
- [24] D. G. H. Sorensen, T. D. Brunoe, and K. Nielsen, "A Classification Scheme for Production System Processes," in *51st CIRP Conference on Manufacturing Systems*, vol. 72, pp. 609–614, 2018. Available: <https://doi.org/10.1016/j.procir.2018.03.021>
- [25] QualiWare, "QualiWare Enterprise Architecture. Center of Excellence," 2018. [Online]. Available: <https://coe.qualiware.com/>.
- [26] T. Gschwind, J. Pinggera, S. Zugal, H. A. Reijers, and B. Weber, "A Linear Time Layout Algorithm for Business Process Models," *Journal of Visual Languages & Computing*, vol. 25, no. 2, pp. 117–132, 2014. Available: <https://doi.org/10.1016/j.jvlc.2013.11.002>
- [27] I. Kitzmann, C. König, D. Lübke, and L. Singer, "A Simple Algorithm for Automatic Layout of BPMN Processes," in *2009 IEEE Conference on Commerce and Enterprise Computing, CEC 2009*, pp. 391–398, 2009. Available: <https://doi.org/10.1109/CEC.2009.28>