UHASSELT
KNOWLEDGE IN ACTION

Faculty of Business Economics

Master of Management

Masterthesis

Discover process dependencies of manufacturing operations in the CPG industry

Frederik Verwimp

Thesis presented in fulfillment of the requirements for the degree of Master of Management, specialization Business Process Management

SUPERVISOR :
Prof. dr. Mieke JANS

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2017
2018

Faculty of Business Economics

Master of Management

Masterthesis

Discover process dependencies of manufacturing operations in the CPG industry

Frederik Verwimp

Thesis presented in fulfillment of the requirements for the degree of Master of Management, specialization Business Process Management

SUPERVISOR :
Prof. dr. Mieke JANS

SUMMARY

The purpose of the research in this master dissertation is to construct a method to discover the dependendies between business processes in the Consumer Packaged Goods (CPG) industry. This purpose has been driven by questions which came up during my personal career, but also due to the relation between process dependencies and management issues in the CPG industry.

The management issues which were mentioned above are elaborated in the Introduction part of this master dissertation. First, the requirements in order to be competitive in the CPG industry are discussed. These requirements are: Efficiency, Flexibility, Quality and Time to Market, which are the main drivers for management issues in the CPG industry. Before going into more detail about these requirements, the typical CPG plant and its manufacturing operations are explained. Based on the explained manufacturing operations, the relation between process dependencies and the requirements for being competitive is demonstrated. This demonstration provides the basis for the problem statement and its related research questions. A problem statement which explains how the construction of the dependency discovery method will be approached using the design science research methodology.

The main foundations of the dependency discovery method are discussed during the Literature review. These foundations are information models and modeling methods related to the manufacturing operations of the CPG industry. The most important foundations are the ISA 95 models coming out of the International standard for automation. The ISA 95 models provide all business processes and manufacturing activities which are needed in the CPG industry. These ISA 95 operating models will be put into the value chain in order to provide the full process and information flow within a CPG plant. Another important foundation is value stream mapping. Value stream mapping offers a framework to identify business and plant objects related to the manufacturing operations of a CPG plant. Finally, modeling methods as BPMN and object lifecycle analysis are discussed in order to be used later in constructing the dependency discovery method.

The actual discovery method will be constructed in the Dependency discovery method part using the design science research methodology. In this part the following steps of the research methodolgy are executed: identifying the problem, define objectives and design of the solution. The remaining steps of the research methodology are executed in a later part of this master dissertation. Initially, all modeling and analysis building blocks of the discovery method are individually discussed during the design step. These building blocks assist in identifying the

processes and objects that need to be analyzed and provide support to perform the time dependency analysis. As a conclusion, all these building blocks are put together in order to form the full dependency discovery method.

The dependency discovery method which has been constructed in the Dependency discovery method part is demonstrated and evaluated in the Demonstration part. Demonstration and evaluation are the remaining elements of the design science research methodology. Three processes will be identified and analyzed using the dependency discovery method in order to obtain the dependencies of these processes. The identified processes are: Outbound inventory tracking, Outbound inventory data collection and Outbound inventory execution. The analysis of these three processes results in the end-to-end dependency matrices of these processes. These matrices form the process dependency model when they are all combined. The objective for developing the dependency discovery method was that the method should have a small footprint. During the evaluation it is shown that the constructed method meets that objective and it can be concluded that also the goal of this master dissertation has been achieved.

TABLE OF CONTENTS

SUMMARY **1**

TABLE OF CONTENTS **3**

LIST OF FIGURES **7**

LIST OF TABLES **9**

LIST OF ABBREVIATIONS **11**

PREFACE **13**

INTRODUCTION **15**

1. COMPETITIVENESS IN THE CONSUMER PACKAGE GOODS INDUSTRY 15
2. THE PROCESS DEPENDENCY PROBLEM 17
   - *2.1. An operational CPG plant overview 17*
   - *2.2. Identification of business processes and manufacturing activities 18*
   - *2.3. Dependencies in manufacturing operations 19*
   - *2.4. Manufacturing operations implementation 21*
   - *2.5. The problem statement 21*
   - *2.6. The research methodology 22*

LITERATURE REVIEW **23**

1. THE ISA 95 STANDARD 23
   - *1.1. The enterprise-control model 23*
   - *1.2. Manufacturing activity models 24*
   - *1.3. ISA 95 and Porter's value chain 27*
2. VALUE STREAM MAPPING 29
   - *2.1. Overview 29*
   - *2.2. Value stream mapping process 29*
   - *2.3. Product families & the critical path 30*
   - *2.4. Collecting Manufacturing business objects 31*
3. CPG PLANT CHARACTERISTICS 32
4. MODELING PROCESS DEPENDENCIES 33

THE DEPENDENCY DISCOVERY METHOD **35**

1. PROBLEM IDENTIFICATION AND MOTIVATION 35
2. OBJECTIVES OF THE DEPENDENCY DISCOVERY METHOD 35
3. DESIGN AND DEVELOPMENT OF THE METHOD 35
   - *3.1. Building block 1: Process identification 36*
   - *3.2. Building block 2: Identify business objects 38*

3.3. Building block 3: Identify plant objects *39*
3.4. Building block 4: Determine object Lifecycles *39*
3.5. Building block 5: Process to object state mapping *39*
3.6. Building block 6: Object transition analysis *40*
3.7. Building block 7: Time dependency analysis *41*
3.7.1. Building block 7-1: Process time dependency analysis 42
3.7.2. Building block 7-2: Object time dependency analysis 43
3.8. Building block 8: End-to-end time dependency analysis *43*
3.9. Building block 9: Global plant objects *46*
4. THE DEPENDENCY DISCOVERY METHOD STEPS 46

DEMONSTRATION **47**

1. THE GENERIC CPG PLANT 47
2. DISCOVERING PROCESS DEPENDENCIES 48
2.1. Outbound inventory tracking *48*
2.1.1. Identify plant objects 49
2.1.2. Identify business objects 49
2.1.3. Perform process to object state mapping 49
2.1.4. Determine object lifecycles 49
2.1.5. Perform object transition analysis 49
2.1.6. Perform process time dependency analysis 50
2.1.7. Perform object time dependency analysis 50
2.1.8. Perform end-to-end time dependency analysis 50
2.2. Outbound inventory data collection *51*
2.2.1. Identify plant objects 51
2.2.2. Identify business objects 51
2.2.3. Perform process to object state mapping 51
2.2.4. Determine object lifecycles 52
2.2.5. Perform object transition analysis 52
2.2.6. Perform process time dependency analysis 52
2.2.7. Perform object time dependency analysis 53
2.2.8. Perform end-to-end time dependency analysis 53
2.3. Outbound inventory Execution *54*
2.3.1. Identify plant objects 54
2.3.2. Identify business objects 54
2.3.3. Perform process to object state mapping 55
2.3.4. Determine object lifecycles 55
2.3.5. Perform object transition analysis 56
2.3.6. Perform process time dependency analysis 57
2.3.7. Perform object time dependency analysis 58
2.3.8. Perform end-to-end time dependency analysis 59

3. Evaluation 61

CONCLUSION AND DISCUSSION 63

REFERENCES 65

APPENDIX 67

1. Outbound shipment order 67
2. Material move 67
3. Material move data 67
4. Source tank 67
5. Product lot 68
6. Load equipment 68
7. Load carrier 68
8. Field operator 68

LIST OF FIGURES

Figure 1 - Global CPG players and their portfolio (Convergence alimentaire, n.d.) 16
Figure 2 - Value growth rates of the global CPG Market 17
Figure 3 - A CPG plant overview 18
Figure 4 - SAP order to cash process (SAP University Alliances, 2015) 20
Figure 5 - The ISA 95 enterprise-control model 23
Figure 6 - The ISA 95 enterprise-control model for manufacturing management activities ... 24
Figure 7 - Generic activity model of operations management 25
Figure 8 - Implemented production activity model 26
Figure 9 - Implemented maintenance activity model 26
Figure 10 - Implemented inventory activity model 26
Figure 11 - Implemented quality activity model 26
Figure 12 - Production resource management elaboration 27
Figure 13 - Positioning of ISA 95 models in the value chain 28
Figure 14 - Value stream mapping example 29
Figure 15 - Pareto Quantity, Pareto Revenue analysis and the product family matrix 30
Figure 16 - MPM (left) and machine similarity matrix (right) 31
Figure 17 - Example of a client contract business object lifecycle 34
Figure 18 - The value chain information flow model 37
Figure 19 - The detailed request-response cycle 38
Figure 20 - Dependency timing cases 42
Figure 21 - Generic CPG plant model 47
Figure 22 - Outbound shipment order object lifecycle 67
Figure 23 - Material move object lifecycle 67
Figure 24 - Material move data object lifecycle 67
Figure 25 - Source tank object lifecycle 67
Figure 26 - Product lot object lifecycle 68
Figure 27 - Load equipment object lifecycle 68
Figure 28 - Load carrier object lifecycle 68
Figure 29 - Field operator object lifecycle 68

LIST OF TABLES

Table 1 - Object catalog 38
Table 2 - Process object state matrix 40
Table 3 - Object transition matrix Process Y, Object B 40
Table 4 - Object transition matrix Process Y, Object C 40
Table 5 - Process dependency matrix process X, Y and Z 42
Table 6 - Object transition matrix, Object C 43
Table 7 - Object dependency matrix, Object C 43
Table 8 - Process dependency matrix Process Y 44
Table 9 - Object dependency matrix Object B 44
Table 10 - Object dependency matrix Object C 44
Table 11 - End-to-End dependency matrix 45
Table 12 - Process dependency result, Process Y 45
Table 13 - Process dependency result, multiple processes 46
Table 14 - Outbound inventory tracking process object state mapping matrix 49
Table 15 - Outbound inventory tracking shipment order object transition analysis 49
Table 16 - Outbound inventory tracking Movement object transition analysis 50
Table 17 - Outbound inventory tracking Movement data object transition analysis 50
Table 18 - Process dependency matrix Outbound inventory tracking 50
Table 19 - Outbound inventory tracking shipment order object dependency matrix 50
Table 20 - Outbound inventory tracking Movement object dependency matrix 50
Table 21 - Outbound inventory tracking Movement data object dependency matrix 50
Table 22 - End-to-end dependency matrix outbound inventory tracking 51
Table 23 - Outbound inventory data collection process object state matrix 52
Table 24 - Outbound inventory data collection shipment order transition analysis matrix 52
Table 25 - Outbound inventory data collection Movement transition analysis matrix 52
Table 26 - Outbound inventory data collection Movement data transition analysis matrix 52
Table 27 - Process dependency matrix Outbound inventory data collection 52
Table 28 - Outbound inventory data collection shipment order object dependency matrix 53
Table 29 - Outbound inventory data collection Movement object dependency matrix 53
Table 30 - Outbound inventory data collection Movement data object dependency matrix ... 53
Table 31 - End-to-end dependency matrix outbound inventory data collection 53
Table 32 - Outbound inventory execution process object state matrix (business objects) 55

Table 33 - Outbound inventory execution process object state matrix (plant objects) 55
Table 34 - Outbound inventory execution shipment order transition analysis matrix 56
Table 35 - Outbound inventory execution material move transition analysis matrix 56
Table 36 - Outbound inventory execution source tank transition analysis matrix 56
Table 37 - Outbound inventory execution Product lot transition analysis matrix.................... 57
Table 38 - Outbound inventory execution Equipment transition analysis matrix.................... 57
Table 39 - Outbound inventory execution Carrier transition analysis matrix 57
Table 40 - Outbound inventory execution Operator transition analysis matrix 57
Table 41 - Process dependency matrix Outbound inventory execution (business objects) 57
Table 42 - Process dependency matrix Outbound inventory execution (plant objects)........... 57
Table 43 - Outbound inventory execution shipment order object dependency matrix 58
Table 44 - Outbound inventory execution material move object dependency matrix 58
Table 45 - Outbound inventory execution Source tank object dependency matrix 58
Table 46 - Outbound inventory execution Product lot object dependency matrix................... 58
Table 47 - Outbound inventory execution Equipment object dependency matrix................... 59
Table 48 - Outbound inventory execution Carrier object dependency matrix 59
Table 49 - Outbound inventory execution Operator object dependency matrix 59
Table 50 - End-to-end dependency matrix outbound inventory execution 59
Table 51 - Final End-to-end dependency matrix outbound inventory execution..................... 60

LIST OF ABBREVIATIONS

CPG	Consumer Packaged Goods
ISA	International Society of Automation
VSM	Value stream mapping
MPPC	Multi- Product Process Chart
MPM	Machine-Part Matrix
RM	Raw Material
SF	Semi-Finished goods
FG	Finished goods
MTS	Make To Stock
MTO	Make To Order
BPMN	Business process modeling notation

PREFACE

The main idea for this master dissertation arose from a question I have asked myself during a project I executed. The question at hand was: How can I prove to higher management which misalignments between processes are the actual root cause of the company's lower performance? During the project, the lower performance of the company was addressed by management due to easy quantifiable problems like failing equipment for example. This was done however there were strong indications within the company that the actual problem could be their wrongly implemented processes. However, as long as this could not be quantified, higher management was not willing to consider those indications as a root cause. This brought me to the idea to get more insights into quantifying process interactions.

Quantifying process interactions is a very large domain which entails many facets. This due to the fact that quantifying process interactions requires several disciplines like managing business processes, managing information flow, managing KPI's and many more. Not only the number of disciplines involved determines the size of the domain, but more especially that it can be applied in any industry. Each industry has its specific processes with its specific requirements. Therefor the first questions which came to mind were: Which discipline will be the focus of my research? How generic in terms of the applicable industries does the research need to be performed? In order to be able to quantify process interactions, these interactions need to be identified and documented. As I did not find a method which demonstrated how to discover process interactions, it seemed to me an obvious first step to develop a method to discover them. In order to have a clear focus on developing the discovery method, I chose the CPG industry as a focus industry. This because I have been professionally active within that industry for many years, allowing me to use my process knowledge of the industry.

Finally, I would like to thank Professors Dr. Mieke Jans and Dr. Koen Vanhoof for their support, inspiration and guidance during my research. Also I would like to thank everyone who supported me in any way during this challenging period in my life.

INTRODUCTION

In this part it is shown how competitiveness of factories in the CPG industry is related to process dependencies. To be able to do this, the CPG industry, its challenges and the typical CPG plant operations are initially discussed.

1. Competitiveness in the Consumer package goods industry

How do I implement my manufacturing operations to achieve optimal performance of my consumer packaged goods manufacturing plant in order to remain competitive and sustainable? When you ask this question, even to experienced managers in the industry, you will get a wide range of answers. The answers will range from a list of operations they have implemented, manufacturing operations process overviews, applied best practices or even used software solutions. What these answers have in common is that they focus on what to implement and not mention how to implement or approach the implementation of these manufacturing operations. The question above and the lack of how in the answer are the base for this master dissertation.

Consumer packaged goods (CPG) are a type of goods consumed every day by the average consumer. Products in this category are those that need to be replaced frequently, compared to those that are usable for extended periods of time. While the CPG industry is a solid and growing market from a demographic point of view, it is highly competitive due to high market saturation and low switching costs for consumers.

90% of the products we buy in the grocery store can be defined as consumer package goods, and most of global manufacturers categorize their products into well-known brands as the eye catchers for the consumer. Convergence Alimentaire, a blog focused on the food industry, put together this interesting infographic below, showing how a small number of companies have a significant hold on the CPG industry.

Figure 1 - Global CPG players and their portfolio (Convergence alimentaire, n.d.)

There are several reasons why managers in the CPG manufacturing industry are more and more asking our initial question. For example, we all know the image of local (European) manufacturing plants which shut the doors in order to be moved to low cost countries. Also the fact that the CPG market is more rapidly growing in Asian Pacific, Middle East and Latin America (Euromonitor, 2017), is an obvious reason for CPG companies to invest more in facilities in those regions. This serves as additional reason why our local (European) plants need to be more efficient in order to be competitive against their internal and external low cost competitors. Where plants part of the same enterprise and producing the same product are called internal competitors, and the actual competing brands external competitors. Although the external competition is obvious, internal competition is also a very important factor due to the fact that the profitability and efficiency are important factors for global manufacturers to determine in which plants to invest for the future.

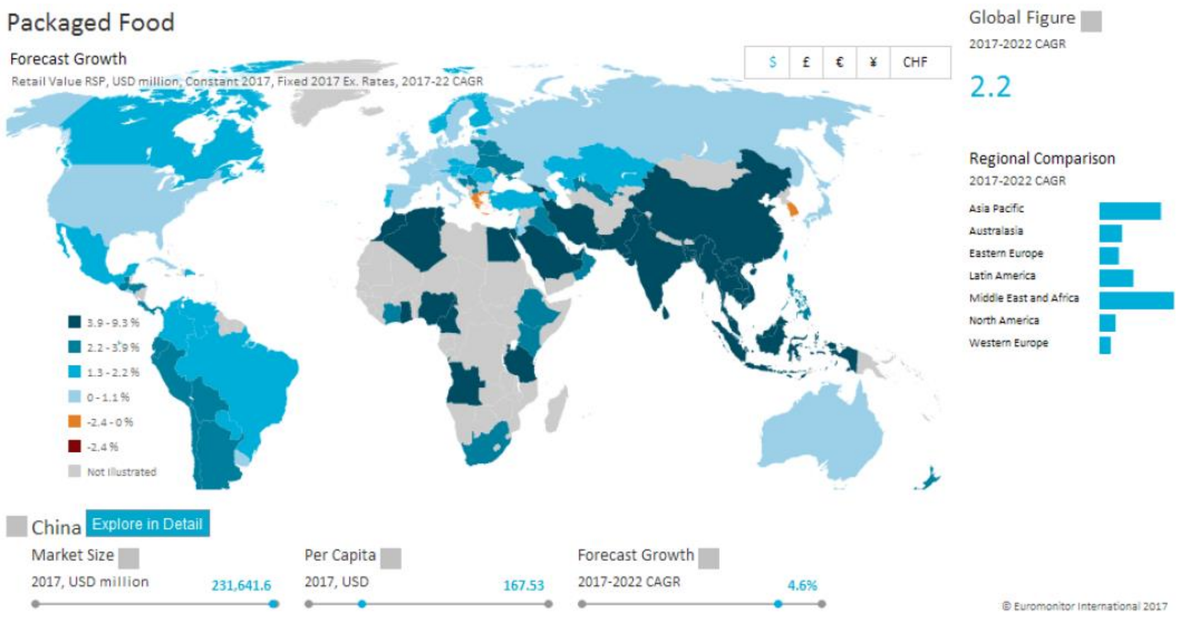

Figure 2 - Value growth rates of the global CPG Market

Competitiveness of a manufacturing plant is summarized into following key requirements by the industry: Efficiency, Flexibility, Time to market and Quality (Siemens, 2018).

Efficiency is related to the usage of plant resources like equipment, raw materials, personnel, … and how efficient they are used in a day to day manufacturing operations. Flexibility is the ability to adjust manufacturing processes to be able to quickly react on fast changing demands of the market. Time to market is the ability to introduce new products into the market and manufacture them. And finally, quality refers to the ability to provide consistent quality in order to avoid waste and product recalls. Each of these requirements are dependent on several factors, but what they all have in common is that they are impacted by how manufacturing operations are implemented.

2. The process dependency problem

2.1. An operational CPG plant overview

Before we look into how manufacturing operations need to be implemented, a high level view of what a typical CPG plant and its manufacturing operations is, is provided.

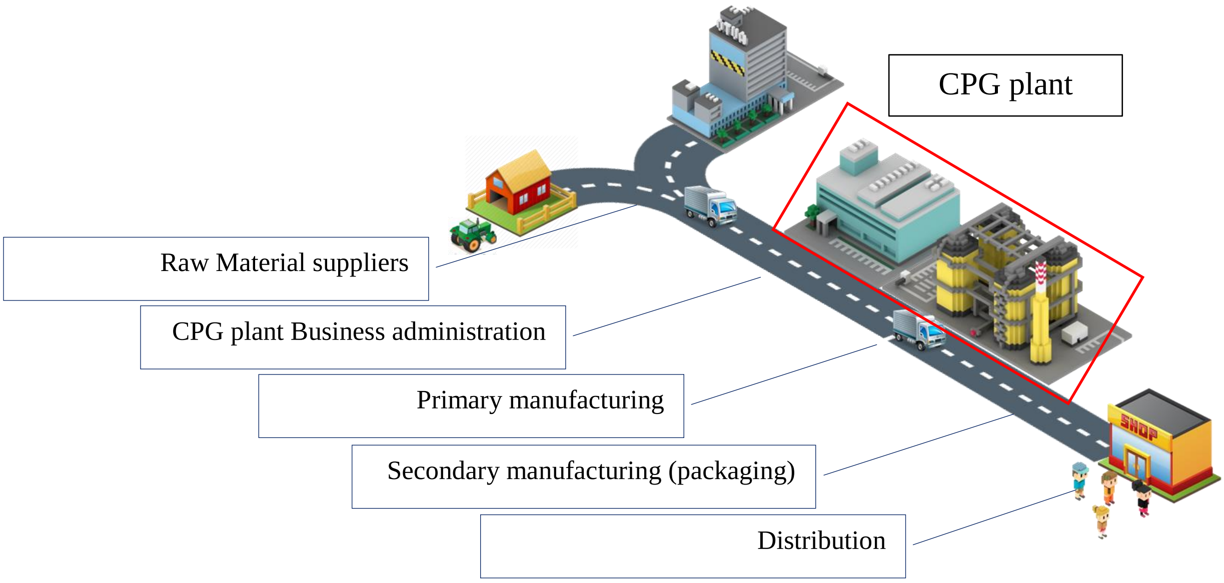

Figure 3 - A CPG plant overview

To show how a typical CPG plant operates, the total picture from suppliers till end customers is shown in the picture above. From top to bottom you see: Raw material suppliers, delivering their packed or bulk raw materials to the CPG plant. These are typically procured by the CPG plant business administration where also customer orders are handled which are turned into manufacturing orders. But also other management services as planning and scheduling, maintenance management, engineering, marketing, customer service and many more are part of the business administration. The CPG business administration drives the actual manufacturing of finished and semi-finished goods by providing manufacturing orders. The primary and secondary packaging manufacturing which execute the manufacturing orders are the core of a CPG plant. They perform the actual transformation from raw materials to finished goods and provide the most of the value for the plant. Finally, when products are manufactured and packaged, they are distributed to retailers and end customers.

The transformation of raw materials to finished goods is performed by manufacturing processes in primary and secondary manufacturing. Allthough these manufacturing processes consist out of activities, flows and gateways like business processes, they are often called manufacturing activities instead of manufacturing processes. This is done to distinguish manufacturing processes from business processes more easily. The collection of business processes and manufacturing activities is often called manufacturing operations by the industry.

2.2. Identification of business processes and manufacturing activities

Identifying which business processes and manufacturing activities are typically used in a CPG plant helps to obtain a more detailed view on how a typical CPG plant operates. One of the main sources of manufacturing operations information is the international standard of automation which has published the ISA 95 standard for manufacturing operations management (International Society of Automation, 2007). This standard provides models of all processes which are generally needed in a manufacturing environment.

The ISA 95 standard provides all strategic processes involved in manufacturing and their relations. The standard shows the relations between business processes as marketing & sales, accounting, order processing,… and manufacturing activities performed in primary and secondary production. It also shows the relations between distinct manufacturing activities. The models it contains are not intended to represent an actual implementation of a manufacturing information system. However, it provides a consistent framework of what to implement in a manufacturing environment (International Society of Automation, 2007).

2.3. Dependencies in manufacturing operations

The relations between the distinct business processes and manufacturing activities imply dependencies. These dependencies can be shown by using the order to cash process as an example. Figure 4 shows the known SAP order to cash process (SAP University Alliances, 2015), which is present in a manufacturing context.

The SAP order to cash process as presented in Figure 4 shows how a sales order of a finished product is entered by the business administration after the pre-sales activities are performed. In the next step, the business administration is going to check if the needed finished product is available in stock. If the finished product is available, it is picked and packed in order to be shipped towards the customer. When shipped towards the customer, a goods issued is posted, which means that the finished product has left the plant towards the customer. After the product has been shipped, the related invoice is being sent to the customer and the payment is expected to be received.

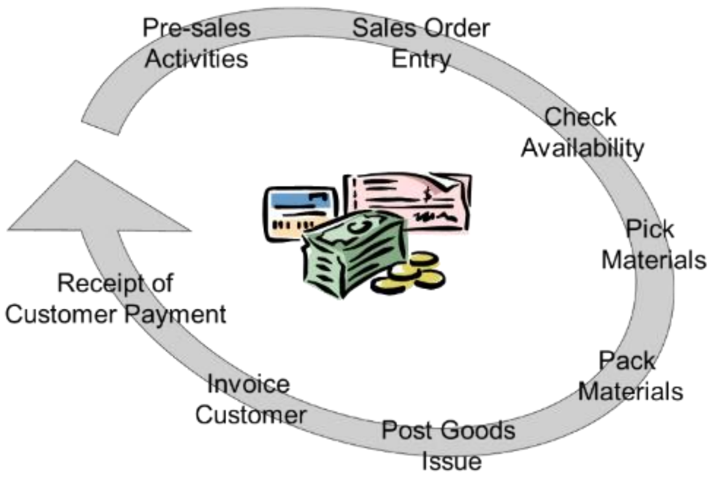

Figure 4 - SAP order to cash process (SAP University Alliances, 2015)

When in the check availability step it seems that there is not sufficient finished product available, it means that the missing finished products needs to be manufactured. This has to trigger the business process to schedule these missing products in the production manufacturing activity. The production manufacturing activity itself includes following prerequisites:

» Availability of manufacturing equipment
» Availability of raw materials
» Storage of (semi-) finished goods possible
» …

The production manufacturing activity also contains following tasks:

» Material dosing
» Sample taking
» Sample measurement
» …

If one of these prerequisites is not fulfilled, for example the availability of raw materials, the material dosing task cannot be performed and another business administration process is triggered. In this example, the procurement of the missing raw materials is launched. The procurement on its turn triggers the material reception manufacturing activity which involves following tasks:

» Weighing operations
» Unloading activities
» Sample taking
» …

The example above clearly shows the relations and dependencies between manufacturing activities and business processes.

2.4. Manufacturing operations implementation

It is clear that if in the previous example the missing raw materials were procured earlier, the production manufacturing activity would not have been delayed. Also if the production schedule was better aligned to customer demands, the products could be directly picked in the order to cash process. These are 2 simple examples to show that the dependencies of manufacturing operations are a cause of inefficiencies in a manufacturing plant and are affecting efficiency.

In order to be able to decrease these inefficiencies, the dependencies between manufacturing operations need to be identified. Knowing these dependencies, the interactions between all manufacturing operations could be aligned. This by adjusting the implementation and start triggers of the impacted business processes and manufacturing activities. Ideally, to align the interactions between processes and activities, one would preferably have guidance of a model which describes all process and activity dependencies and their characteristics. By having such a model, CPG plants would not only be able to increase their efficiency. They also would be able to adjust their process interactions if needed due to changes in the market, providing more flexibility. By being able to adjust the alignment of processes, also a shorter time to market and increased quality could be achieved

2.5. The problem statement

The purpose of this master dissertation is to construct and demonstrate a method to approach the discovery of process and activity dependencies. The deliverable of the method is a model showing a set of dependencies between all manufacturing operations described by their characteristics. The dependencies in the model can then be used to align the interactions between manufacturing operations. The model will be called the process dependency model. In order to reach the objective the following research questions need to answered:

- Which processes of a CPG plant need to be analysed?
- What is the ideal sequence in which processes need to be analyzed?
- How is a dependency between two or more processes detected?
- How is a dependency between two or more processes described?

The last research question can be further detailed with following subquestions:

- Which business and plant objects are involved?
- How to model the dependencies of a process on an object and its state?
- Which global business objects, related to the CPG plant but not to a process, need to considered? E.g. total lead time of a product.
- How to model global dependencies?

2.6. The research methodology

To come to the dependency discovery method, the design science research methodology (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2008) will be used, which contains following steps: problem identification and motivation, define objectives, design and development, demonstration and evaluation. Applied to this master dissertation the research method is split up into two main parts. The dependency discovery method part will handle the steps from problem identification till development. It will show and elaborate all building blocks and steps which are needed in the dependency discovery method. The second part is the Demonstration part, which will demonstrate and evaluate our dependency discovery method.

LITERATURE REVIEW

The purpose of this part is to document process models and methods which are currently used in the CPG industry to collect process information.

1. The ISA 95 standard

1.1. The enterprise-control model

As already mentioned in the introduction, the ISA 95 standard provides all strategic processes and their relations that are involved in manufacturing. The ISA 95 is an industry standard which is developed by the international society of automation. This society is a non-profit organization which brings together people and organizations related to industrial automation and manufacturing. Their goal is to develop standards for all disciplines related to industrial automation and manufacturing. One of those standards is the ISA 95 standard. This standard provides standardized objects, terminology and several models of the processes involved with manufacturing, each of them with another level of detail. The main model is the enterprise-control model which presents:

- The process and activity domains of an enterprise involved in manufacturing.
- The information flows between those domains

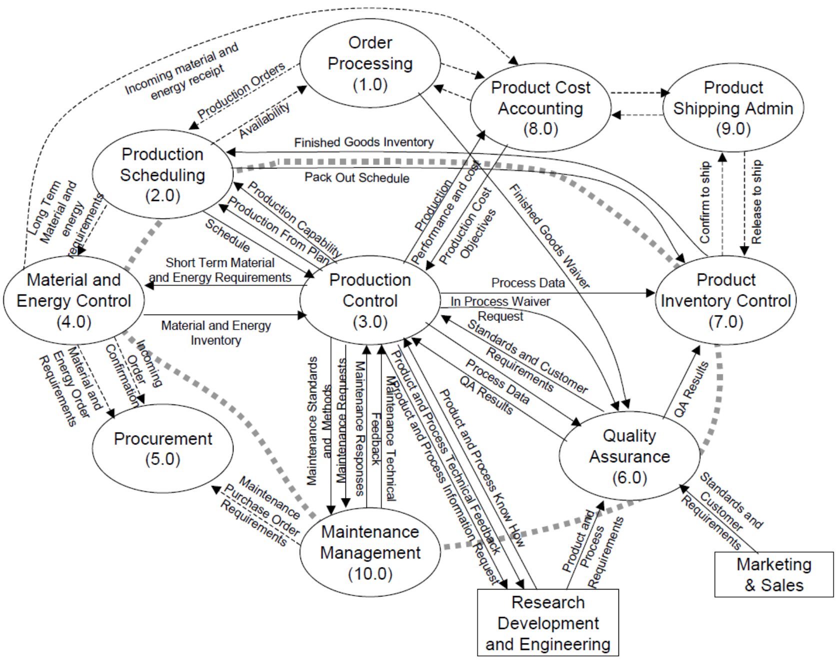

Figure 5 - The ISA 95 enterprise-control model

The grey dotted line illustrates the enterprise-control interface which is the boundary of the enterprise its business processes and manufacturing activities. The production control side of the interface includes the manufacturing activities and the other side the business processes.

The labeled arrows indicate information flows of importance to manufacturing operations. The heavy dotted line also intersects domains that have processes or activities that may fall into the manufacturing domain, or into the enterprise domain depending on the implementation. The main purpose of the enterprise-control model is to provide an inventory of business processes, manufacturing activities and data flows. The full list of processes, activities and data flows can be found in the ISA 95 standard part 1 chapter 6.

1.2. Manufacturing activity models

According to ISA 95, Manufacturing activities are divided into 4 domains: production operations management, maintenance operations management, quality operations management and inventory operations management (International Society of Automation, 2007). The grey areas in the enterprise-control model shown below represent the 4 manufacturing management domains. One layer deeper than the enterprise-control model, the ISA 95 standard provides a generic activity model for manufacturing activities. The generic activity model provides a generic set of activities and their interactions. It also shows how the manufacturing activities are connected to the business processes. This generic activity model will later be used to create a specific implementation of the activity models for each of the 4 manufacturing domains.

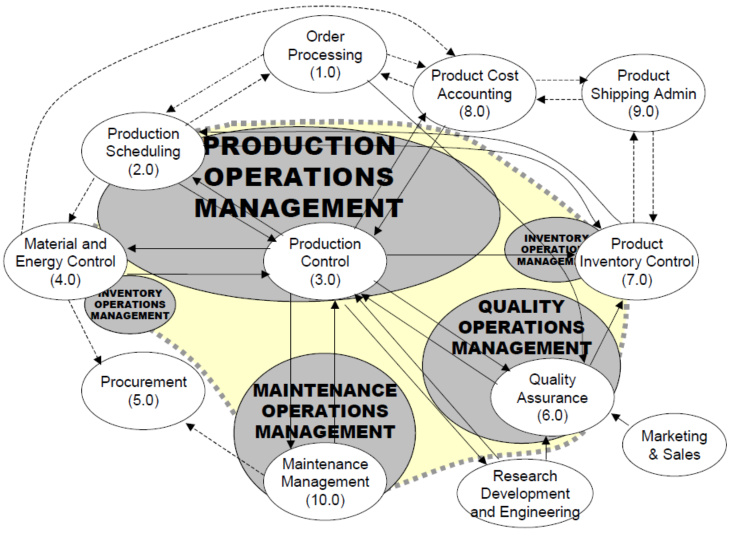

Figure 6 - The ISA 95 enterprise-control model for manufacturing management activities

The generic activity model shown in figure 7 defines a general request-response cycle that starts with the operations request, converts it into a detailed schedule, dispatches work according to the detailed schedule, executes the work, collects data, and converts the collected data back into responses (International Society of Automation, 2007). This request-response cycle is supported with:

» Analysis of the work that was done to allow improvements or corrections
» Management of the resources used in execution of the work to be done.
» Management of the definitions of the work to be done.

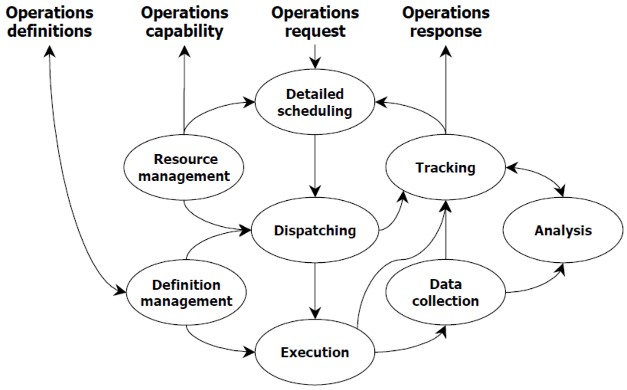

Figure 7 - Generic activity model of operations management

Each of the 4 manufacturing domains has a specific implementation of the generic activity model where each circle is replaced with the specific implementation of the actual activity for that domain. For example, resource management in the generic activity model is replaced by production resource management for the production operation management domain. The implementation of the generic activity model for each domain can be found in figure 8, 9, 10 and 11 in the following order: production operations, maintenance operations, inventory operations and quality operations. These models, combined with the enterprise-control model, give an overview of all manufacturing operation interactions.

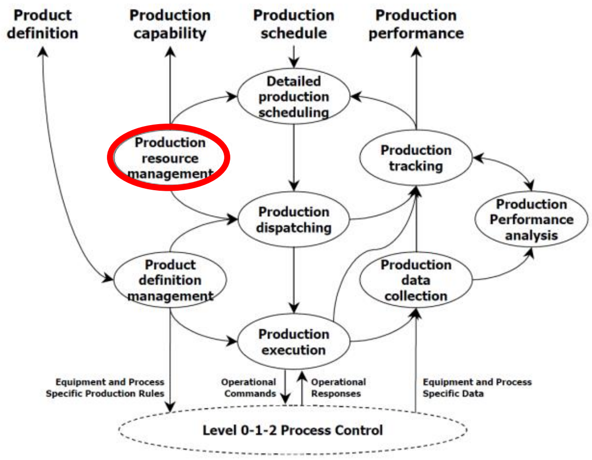

Figure 8 - Implemented production activity model

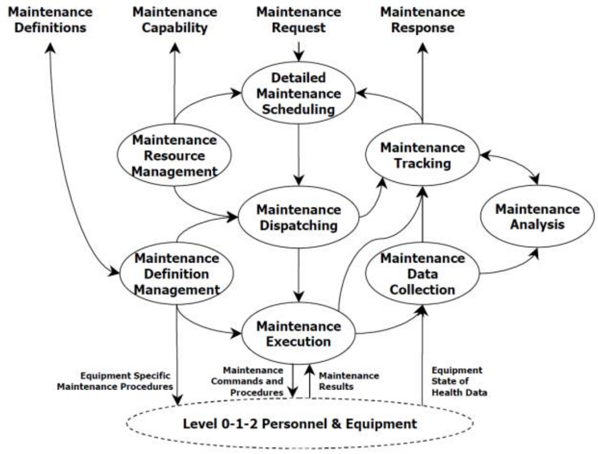

Figure 9 - Implemented maintenance activity model

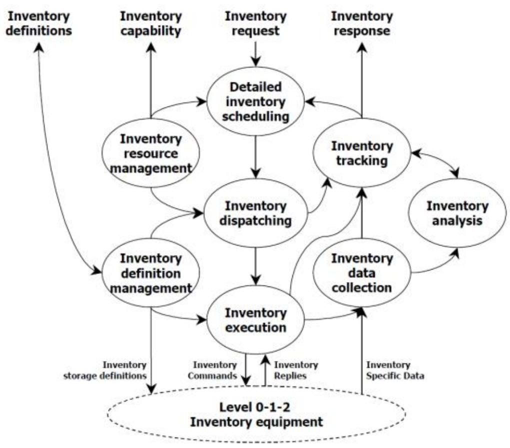

Figure 10 - Implemented inventory activity model

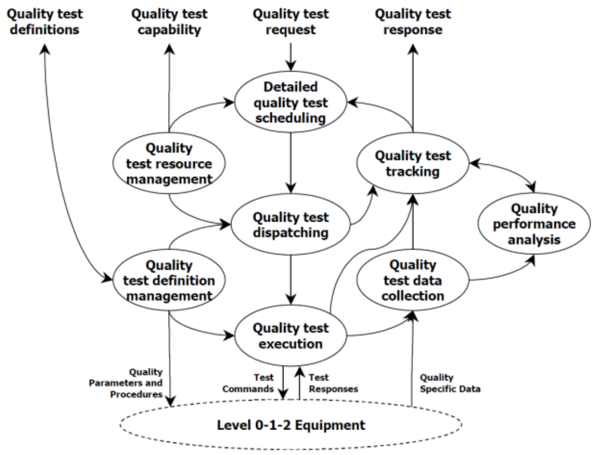

Figure 11 - Implemented quality activity model

Within the ISA 95 standard, each activity (circle) of the implemented activity model of each domain is individually elaborated. The elaboration shows the tasks in the activity and which information is shared with related activities in the implemented model. Because all of these elaborations cannot be enclosed, only a single example is shown. The example activity shown in figure 12 is the production resource management activity of the production operations management domain. This can be found in its respective implemented activity model in figure 8 and is indicated by the bold red circle. This activity consists out of following tasks:

- Providing personnel, equipment and material resource definitions
- Collecting personnel, equipment and material availability
- Providing information on available resources to production scheduling and production dispatching
- Collecting information on the current state of process control
- Collecting future needs from the detailed production schedule
- Report available production capabilities

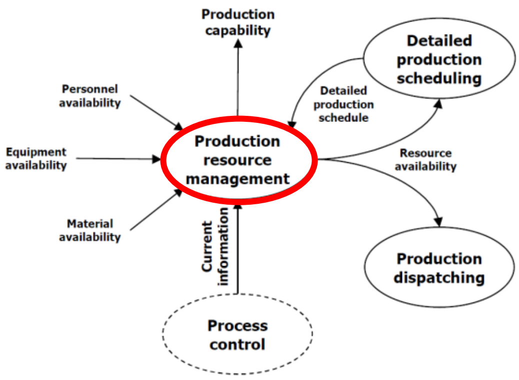

Figure 12 - Production resource management elaboration

It is also clearly shown which high level information serves as input and as output to other activities by the arrows going in and out of the activity. The full elaboration of all processes and information flows is shown in the ISA 95 standard part 3 chapter 5, 6, 7 and 8.

1.3. ISA 95 and Porter's value chain

Brigitte Borja de Mozota (Mozota, 1998) provided an important reason why to visualize the presented ISA 95 models into the value chain and how they interact. *Michael Porter's value chain offers a useful framework for synthesizing research on design management and structuring the knowledge where Information plays a preeminent role in any firm's value chain.*

In figure 13, a visual representation is made of where the individual implemented activity models are positioned into the value chain. Also what their hierarchy is and how information flows through the value chain and the ISA 95 models. Above the value chain you can see the ISA 95 enterprise-control model. This model encompasses all processes which are used in the total value chain. The inventory operations activity model is positioned in the inbound and outbound chevron. The production activity model in the operations chevron. Finally, the quality and maintenance activity models are related to all other activity models in those 3 chevrons. In the marketing and sales chevron, a partial enterprise-control model is placed to indicate that these business process are positioned there.

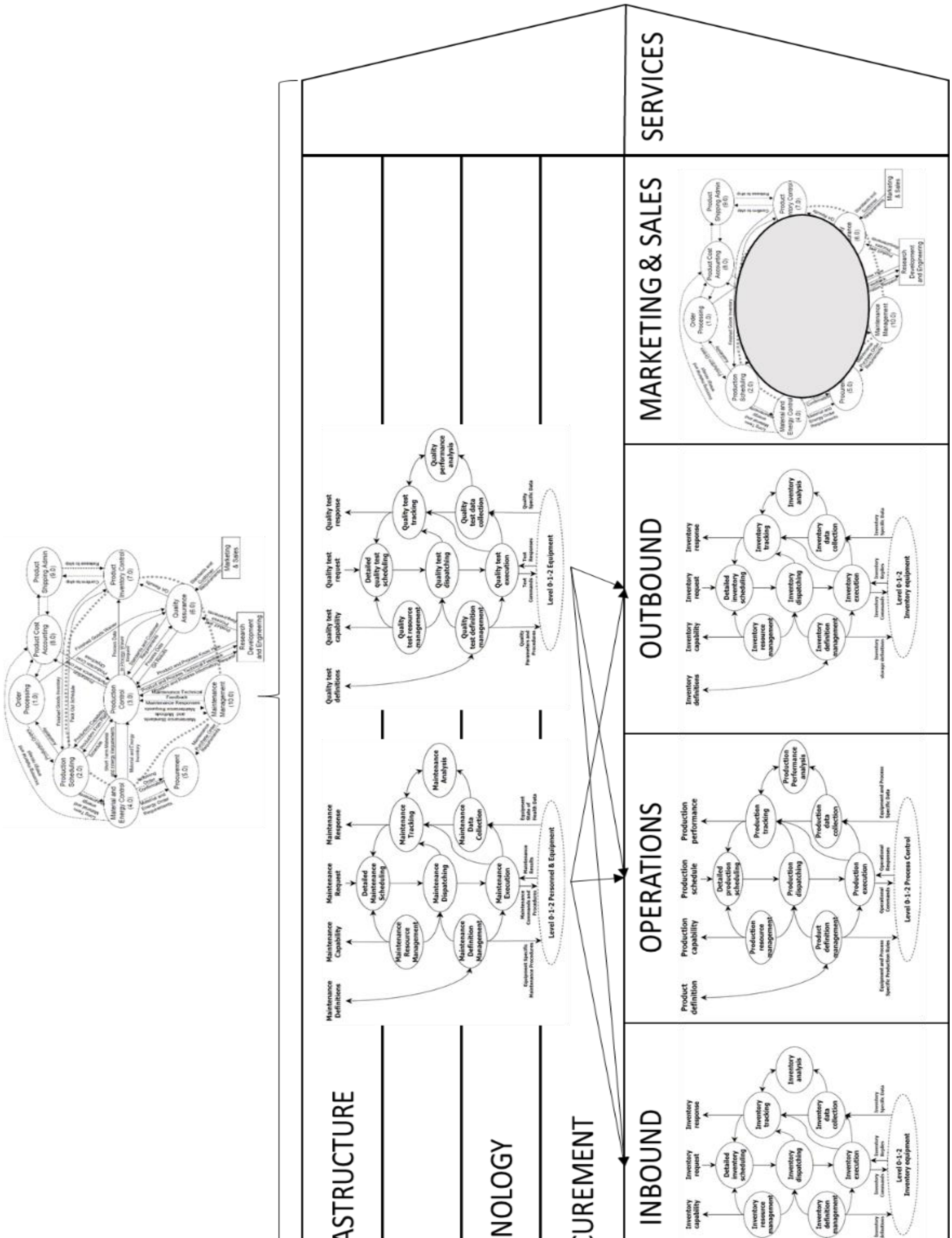

Figure 13 - Positioning of ISA 95 models in the value chain

2. Value stream mapping

2.1. Overview

A value stream is a collection of all actions (value added as well as non-value-added) that are required to bring a product (or a group of products that use the same resources) through the main production flows, starting from raw material and ending with the customer. Value stream mapping (VSM) is a tool that helps you understand the properties of the flow of material and information as a product makes its way through the value stream. It is performed by following a product's production path from supplier to customer and draw a visual representation of every process in the material and information flow. An example of a value stream map is shown below. The ultimate goal of VSM is to identify all types of waste in the value stream and to take actions to try to eliminate these (Rother & Shook, 1999).

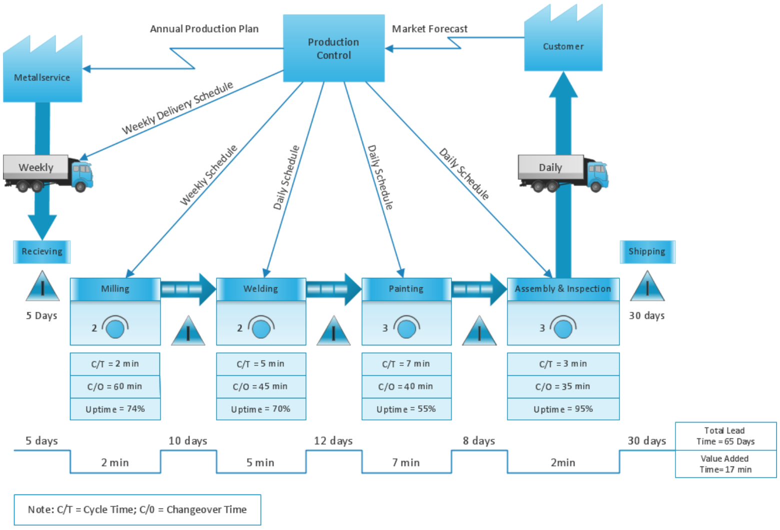

Figure 14 - Value stream mapping example

In order to know which business and plant objects are involved in the business processes and manufacturing activities, this can offer a framework. Value stream mapping can also help to identify global business objects which are not directly related to a business process or manufacturing activity, but are related to the characteristics of the CPG plant.

2.2. Value stream mapping process

VSM starts by selecting a single product family, which is also one of the drawbacks of VSM. It only allows to map one product family at a time. The next step is to draw a current state map that is essentially a snapshot capturing how things are currently being done, within all production steps. In particular, VSM can only be effectively used for production systems characterized by linear product routings. If the production process is complex, VSM application fails to map value streams characterized by multiple flows that merge. This typically happens for products described by a complex Bill of Material (BOM) as in the CPG industry. An alternative and innovative framework for a structured application of VSM to products that require nonlinear value streams, based on existing VSM, is created by Braglia, Carmignani, & Zammori (2006). The new framework adds methods to identify product families, machine sharing and the critical production path in the VSM process.

2.3. Product families & the critical path

As stated in the overview about value stream mapping, the main interest of VSM is to identify business and plant objects related to processes and to the global characteristics of a CPG plant. Also, it has been stated that using VSM, a current state map can only be constructed for a single product family. To get a complete view of all CPG plant processes, characteristics and related business objects, all product families need to be identified. In order to identify all product families, several possibilities are provided by Braglia et al (2006). A Pareto Quantity (PQ) analysis and a Pareto Revenue (PR) analysis of products can be combined in order to result in a product family matrix as shown in figure 15. The product family matrix is obtained by dividing the PR and PQ analysis results in three (or even more) groups A, B and C (high to low revenue or quantity). Then each product is set in the cell which matches its group for quantity and revenue. In this way 9 product families are obtained.

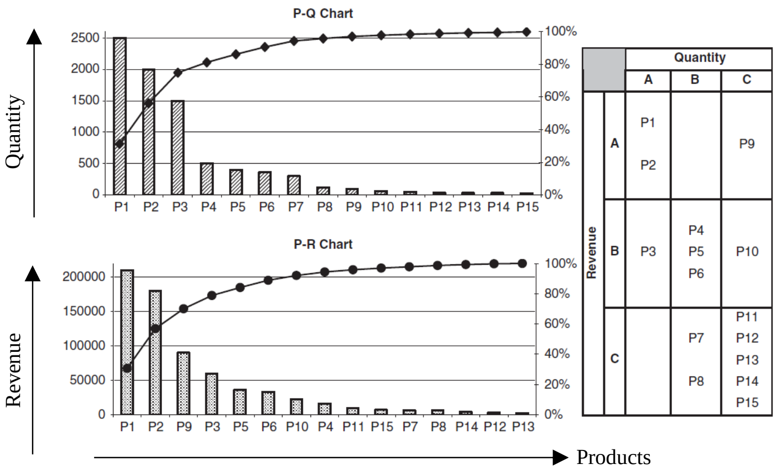

Figure 15 - Pareto Quantity, Pareto Revenue analysis and the product family matrix

A product family can be defined as: "a group of products that pass through similar processing steps and over common machines in the downstream processes". This makes it possible to group products into families through the analysis of the Multi-Product Process Chart (MPPC) or, alternatively, the Machine-Part Matrix (MPM). It is important to note that the use of a MPM is preferable if dealing with a great number of products. The MPM matrix on the left in figure 16 shows the machines in each column used to produce each product in each row. The MPM matrix can be rearranged to identify the used product families based on similarly used machines.

The rearranged MPM or machine similarity matrix on the right of figure 16 clearly shows similar used machines within the product families. These similar used machines are grouped in what are called production cells. It also shows ‘external’ shared equipment between families e.g. Machine M15 which is also used for product P3 and P10. Machine M15 is used in family 1 which consists out of product 2, 5 and 6. Allthough product 3 and 10 are not part of family 1, they also require machine 15. Machine M15 is Therefor a shared equipment between families.

Product: P

Machine: M

	M #1	M #2	M #3	M #4	M #5	M #6	M #7	M #8	M #9	M #10	M #11	M #12	M #13	M #14	M #15
P#1					1	1	1			1		1	1		
P#2	1		1				1		1					1	1
P#3		1		1							1	1			
P#4					1	1		1		1		1			
P#5	1		1				1	1	1					1	1
P#6	1			1			1		1					1	1
P#7					1	1		1		1		1	1		
P#8		1		1							1		1		
P#9					1	1	1			1			1		
P#10		1		1							1		1		1

		Family #1			Family #2				Family #3		
		P#5	P#2	P#6	P#4	P#7	P#1	P#9	P#8	P#10	P#3
Cell #1	M#1	1	1	1							
	M#14	1	1	1							
	M#9	1	1	1							
	M#7	1	1	1			1	1			
	M#15	1	1	1						1	1
	M#3	1	1								
Cell #2	M#5				1	1	1	1			
	M#6				1	1	1	1			
	M#10				1	1	1	1			
	M#12				1	1	1		1		
	M#8	1			1	1					
Cell #3	M#13				1			1		1	1
	M#4			1					1	1	1
	M#2								1	1	1
	M#11								1	1	1

Figure 16 - MPM (left) and machine similarity matrix (right)

Based on the machine similarity matrix, the critical path of each family can be determined. The critical path can be defined as the processing sequence which is responsible for the total production time that determines the minimum time frame needed to schedule production in advance (Braglia, Carmignani, & Zammori, 2006). The critical path can be simply obtained by selecting the production sequence characterized by the maximum vertical length in the machine similarity matrix. For example, product P4 has the longest vertical length for Family 2. This is circled in figure 16 on the right side. Therefor the equipment of product P4 provide the crytical path for family 2. The current state map of the critical path can then be constructed for each product family.

2.4. Collecting Manufacturing business objects

Having identified the product families, VSM can be applied to construct the current state map for each family. In this way, the typical involved business and plant objects can be collected. Rother and Shook (Rother & Shook, 1999) recommend that business object collection, using VSM, should begin at shipping, working backwards in the production process towards raw materials or suppliers and collecting snapshot data on inventory levels at each stage of the value stream.

To build the current state map, VSM has the feature of recording information associated with the material and information flows of the family under analysis. Following business and plant objects are typically collected when building the current state map:

- Information flow:
  - Customer order types
  - Supplier order types
  - Order frequency
  - Forecast frequency
  - Production plan time frame
- Machines:
  - Change over times
  - Up time
  - Cycle time
  - Number of operators
  - Number of shifts
- Production flow:
  - Average customer demand
  - Shipping frequency
  - Average production batch size
  - Inventory levels
  - Push or pull flows
  - Lead time

The business and plant objects shown above can even be used to obtain further derived objects, based on analysis of above objects. For example the identification of the maximum capacity based on the available time and change over time (Langstrand, 2016).

3. CPG plant characteristics

Even though value stream mapping can serve as a useful framework for capturing objects of a manufacturing process, it might not reveal all needed business and plant objects. The constraints which are used to perform planning and scheduling can also be used to identify objects of a CPG plant (Kallrath, 2002). This because planning and scheduling constraints are described by objects of a CPG plant and its products. Therefor the constraints can contain objects which were not revealed during VSM. A comparison of planning and scheduling constraints and VSM objects can be performed in order to reveal unidentified objects. These planning and scheduling constraints can be summarized as (Kallrath, 2002):

- multi-purpose equipment (multi-product, multi-mode)
- sequence-dependent set-up times and cleaning times
- combined divergent, convergent and cyclic material flows
- non-preemptive processes (no-interruption), buffer times
- multi-stage, batch & campaign production using shared intermediates
- multi-component flow and nonlinear blending
- finite intermediate storage, dedicated and variable tanks for different pack types

4. Modeling Process dependencies

To be able to model the dependencies between processes, a way to document them is required. The investigation of how to document process dependencies is approached by looking at existing modeling and notation languages and see what can be useful for our purpose. One of the most well-known modeling notations is Business Process Modeling Notation (BPMN). BPMN is a business process modeling language which is maintained by the Object Management Group. It provides a graphical notation standard to document business processes and is especially known for its strong ability to model control-flows and resource assignments. However, it has limited abilities to handle dependencies of those flows. Allthough BPMN allows to link data objects to activities, it is not possible to show the states and dependencies of those data objects. In order to handle these states and dependencies, BPMN has been extended with data object annotations (Meyer, Pufahl, Fahland, & Weske, 2013). These data object annotations are based on the principles of entity relationship modeling used in databases. The primary and foreign keys of entities can be used in data objects to indicate dependencies between multiple data objects, where at least one data object is linked to an activity. The annotations also provide the ability to annotate the required state of the data objects like for example created, released, updated, etcetera.

Allthough data object annotations show the dependency of an activity on a specific data object and its related objects, it does not fulfil our needs. It does not provide the ability to model the dependency of a process on other processes. However, the dependency of a process on other processes can be indirectly expressed using the states and dependencies of its linked data objects. The dependency of a process on other processes can be shown by indicating which processes are needed, to get the data objects that are linked to the process into their required state. These dependencies are in our particular interest as they are the dependencies we are looking for.

To express dependencies using linked data objects, one should know what is needed to get that linked data object in a specific state. Therefor, it is needed to have a view on the possible states and transitions of that data object. This view can be achieved in a business object lifecycle model. This model shows a business object in all its possible states and all possible transitions between those states. In literature it is expressed as follows: A business model lifecycle model charts all the events and activities that may cause a particular object occurrence to change in some way from its birth to its death (Sanz & Nandi, 2012). Figure 17 shows an example of a business object lifecycle of a client contract.

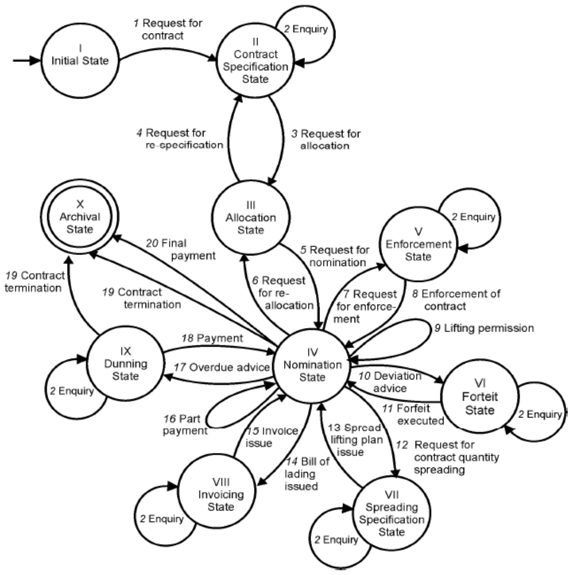

Figure 17 - Example of a client contract business object lifecycle

Allthough a business object lifecycle gives a view on the possible states and transitions of an object, it does not show the dependencies. It shows which transitions exist, but not what caused the transitions. When a single process is responsible for a specific transition, the relation between process and transition is quite clear and the process can be put next to the transition. However, if the transition is caused by multiple processes, it is not so self-evident. Even though it can be shown which processes are related to the transition by listing them next to the transition, it does not show the dependencies between those processes in the lifecycle. In order to show those dependencies, a way to model this has to be found.

THE DEPENDENCY DISCOVERY METHOD

As stated in the introduction, the used research methodology is split up in two parts. The first part focuses on the development of the dependency discovery method and the second part focuses on the demonstration and evaluation of the constructed method. This part focusses on the development of the method using the design science research methodology.

1. Problem identification and motivation

The literature review has shown there is already a lot of information available on manufacturing operations. However, the focus is especially on the implementation of the individual processes and activities rather than on the discovery of process dependencies. Nevertheless, the impact of process dependencies on the efficiency of manufacturing operations has been clearly shown in the introduction.

In order to reduce the impact of process dependencies on efficiency, first of all the actual dependencies need to be identified. Due to the lack of information found regarding modeling and improving process dependencies in the literature review, it can be assumed that improving manufacturing operations efficiency by improving process dependencies is not yet adopted in the industry. By providing a method to discover process dependencies, the usage of process dependencies to improve a plant's efficiency might be adopted in the CPG industry.

2. Objectives of the dependency discovery method

The main objective is to design a dependency discovery method with a footprint which is as small as possible. The small footprint refers to the reuse of existing information as much as possible. By reusing existing information, the effort to obtain all process dependencies is reduced. The lower the effort, the higher the possible willingness to adopt the method by the industry. A derivative of this main objective is that the dependency discovery method also needs to focus on combining existing information and extending existing modeling and analysis methods.

3. Design and development of the method

As already mentioned, the design of the dependency discovery method will focus on using and combining existing information and existing modeling and analysis methods. Each of the following chapters will explain a part which is needed to construct the dependency model. Each of those parts is a building block of the dependency discovery method. The building blocks relate to the research questions mentioned in the problem statement in the introduction. Finally, all these building blocks will be used in order to construct the dependency discovery method.

3.1. Building block 1: Process identification

In order to study process dependencies, one has to know which processes to investigate and in which order. The literature review has shown that there is already an inventory of all manufacturing operations described in the ISA 95 standard, where manufacturing operations are detailed using activity models. The literature review has also shown how these activity models are positioned into the value chain as shown in figure 13. To show the manufacturing activities and processes which need to be analyzed and in which order, we will look at the information flow through the value chain. How information flows through the value chain and its activity models, is further elaborated based on figure 13.

In the value chain information flow model in figure 18, the value chain and the maintenance activity model are left out for simplification reasons. Currently all resources are assumed to be in service. The chevrons are replaced by their respectively implemented activity models which are executed consecutively. You can clearly see that the inbound, operations and outbound activity models are connected to the business processes above and the quality activity model below. The information flow is indicated by the arrows in the model.

Each step and related activity model of the value chain is executed in a similar way and has a similar information flow. Each activity model starts from a demand coming from the business, then it executes the activity model using the request-response cycle as explained in the ISA 95 standard. During the execution step of each cycle, the quality activity model is called to perform quality operations using the quality activity model. When the quality activity model is executed, it will provide a quality result response to the calling activity model its execution step. The calling activity model will then continue from that point. At the end of the cycle a response is provided to the business by the activity model.

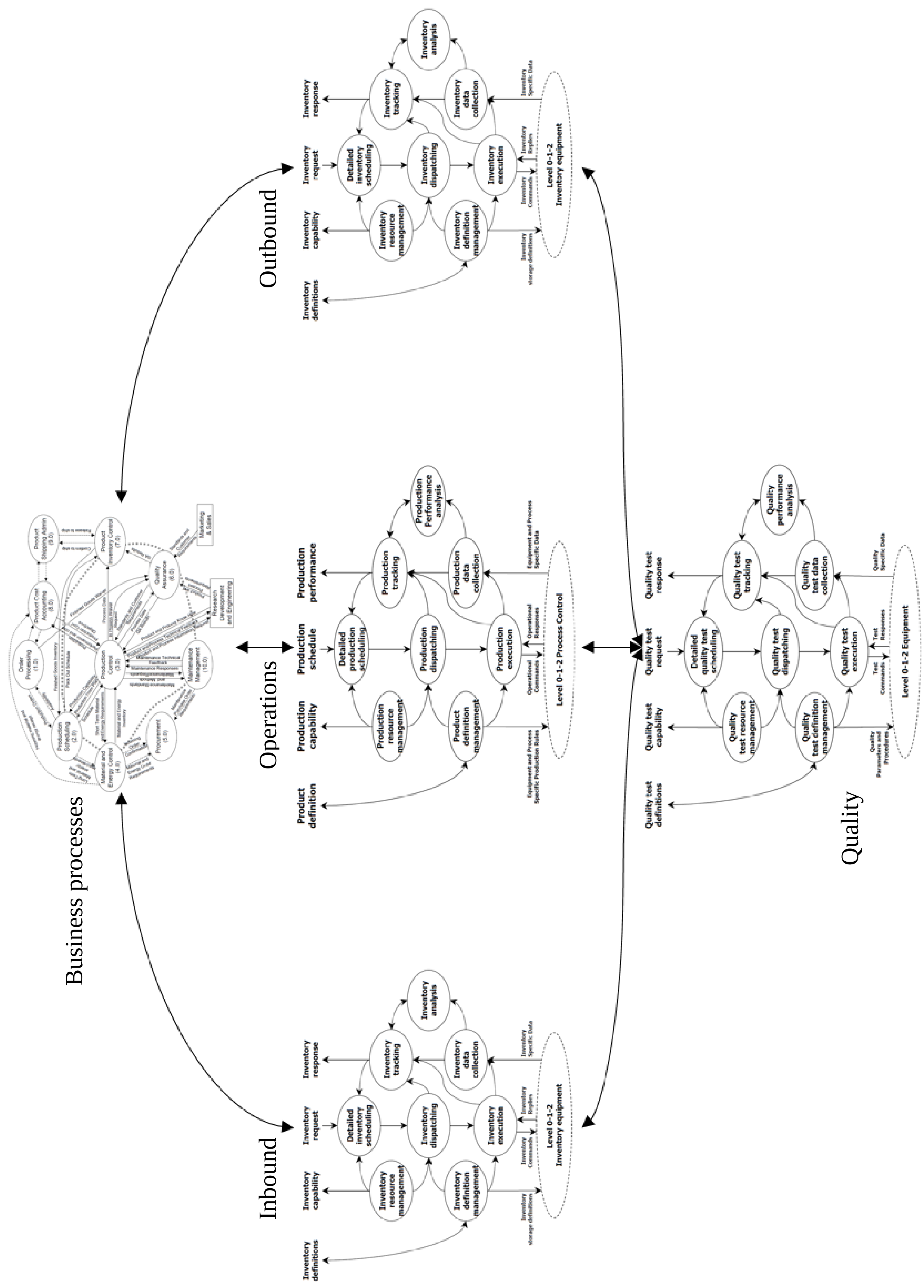

Figure 18 - The value chain information flow model

The detailed request-response cycle of the activity models is shown below. Where the big arrow represents the request-response cycle and the bold marked circles the processes which need to be analyzed being part of that cycle.

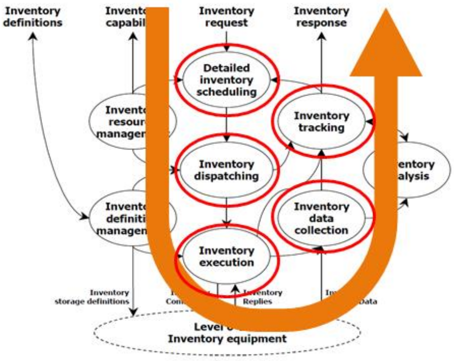

Figure 19 - The detailed request-response cycle

Rother and Shook (Rother & Shook, 1999) recommend that business and plant object collection using VSM should begin at outbound operations and then go backwards in the production process towards the inbound operations. As VSM will be used for object collection, this recommendation will be followed and the same approach will be used. This means that the analysis of process dependencies will start with the outbound activity model and that each activity model will be analyzed going backwards in the request response cycle, starting from the response to the business towards the request.

3.2. Building block 2: Identify business objects

Related to the processes which are going to be analyzed, are the business objects which are part of those business processes and manufacturing activities. These business objects have to be identified in order to be able to analyze their transitions. To identify those business objects, value state mapping can be used. The business objects can be easily identified using the current state map which is created using VSM. All identified objects have to be mapped in an object catalog. This catalog lists the processes and their related objects. An example of such a catalog is shown in table 1. This listing of business objects needs to be done for each process under analysis.

Process	Object
Outbound inventory execution	Outbound shipment order
Outbound inventory execution	Material Move
Outbound inventory execution	Material Move data

Table 1 - Object catalog

3.3. Building block 3: Identify plant objects

Not only business objects related to the business processes and manufacturing activities have to be taken into account. Also the plant objects related to manufacturing activities have to be taken into account. These plant objects need to be in the correct state in order to be able to execute these manufacturing activities. For example, the equipment needed to load the finished product during the outbound shipment execution manufacturing activity. The equipment needs to be available in order to load the finished product to fulfil the demand of the outbound shipment order. Like the business objects, the plant objects for each process or manufacturing activity under analysis need to be identified. To identify those plant objects, value state mapping can be used. The plant objects can be easily identified using the current state map which is created using VSM. The identified plant objects related to the manufacturing activities under analysis also need to be placed in a catalog as done with the business objects.

An important implication for the dependency discovery method is introduced by the usage of value stream mapping. Because value stream mapping is only capable of handling only one product family at a time, this will also be the case for the process discovery dependency method. This because the value stream map can be different for each product family. Meaning that the process dependencies may be different as well. Therefor, having multiple product families implies that the dependency discovery needs to be performed for all product families and will imply that a dependency model will exist for each product family.

3.4. Building block 4: Determine object Lifecycles

Having identified all business and plant objects, the lifecycles of these objects can be determined. This can be done using following steps:

1. Start from the initial object creation state
2. Document each transition, starting from the initial state
3. Document each branch of states until a merge of states or an end state
4. Verify all possible end states

3.5. Building block 5: Process to object state mapping

The first step in detecting process dependencies is to identify the prerequisites of each process. One of the prerequisites is the state of plant and business objects required to be able to execute a business process or manufacturing activity. These object states can be documented in following matrix where in vertical direction the processes are listed and in horizontal direction the objects. In the cells of the matrix the object states required to start these processes are documented. The matrix will be called the process object state matrix and this analysis activity, process to object state mapping. The fictive example below will be used to illustrate the object transition and time dependency analysis where process Y will be analyzed.

Object	Object A	Object B	Object C
Process			
Process X	Created		
Process Y		Released	Ready
Process Z		Loaded	Executed

Table 2 - Process object state matrix

3.6. Building block 6: Object transition analysis

Each of the object states identified during the Process to object state mapping building block, is the result of a transition which has taken place. A transition reflects the changeover from one object state to another object state. To identify what caused the object transitions, an object transition analysis needs to be performed. Below, in table 3 and 4, you can see the analysis for the objects related to process Y. In the horizontal direction you see the resulting object state and in vertical direction the initial state.

Object B To state	Initial	Created	Ready	Released	Loaded
Object B From State					
Initial					
Created			Process A Process D		
Ready				Process B	
Released					Process C
Loaded					

Table 3 - Object transition matrix Process Y, Object B

Object C To state	Initial	Created	Ready	Released	Loaded
Object C From State					
Initial					
Created			Process A Process D		
Ready				Process E	
Released					Process F
Loaded					

Table 4 - Object transition matrix Process Y, Object C

The Object transition matrix in table 3 shows the processes which are needed to move Object B from one state to the other. For example, to move Object B from the Created state (in vertical direction) towards the ready state (in horizontal direction), Process A and D need to be executed. This analysis has to be done for each object related to the process under analysis. In this case the objects related to Process Y. Table 4 shows the object transitions related to object C, which is also related to process Y.

3.7. Building block 7: Time dependency analysis

Two types of possible time dependencies can be identified out of the process to object state mapping and the object transition analysis. A first type is the time dependency on changing object states, where a process is dependent on multiple objects and a required state of each of those objects. The process is dependent on the timing when the required objects obtain their required state. This is shown in table 2 where process Y is dependent on Object B and C. A second type of time dependency can be found within an object itself. Where the transition of a specific object to a specific state is dependent not on one but multiple processes. This creates a timing dependency between those processes. This is shown in table 3 where object B is going from “Created” to “Ready” due to Process A and D.

Both time dependency types mentioned above have in common that they are sequence dependent. In the first case, dependent on the sequence of the required object transitions. In the second case, dependent on the sequence of processes that cause an object transition. Business processes have a similar kind of sequence dependency, as they are dependent on the sequence of their activities. The sequence of activities is determined by sequence flows, sub processes and gateways, which allow to model sequential, parallel, embedded or independent activities. The same sequence mechanisms can be used to model the sequence of object transitions and the sequence of processes that cause an object transition. In general, the sequence mechanisms of business process activities are lifted to the process and object level. This means that objects can transition sequentially, in parallel or independent. It also means that processes that cause an object transition, can be executed sequentially, in parallel, embedded or independent. The list below shows further details about the possible sequence types between object transitions or processes. These sequence types are assumed based on the business process activity sequencing mechanisms:

- Sequential: one process/object needs to be started/transitioned after the other
- Parallel: both processes/objects can to be started/transitioned in parallel
- Embedded: One process is called as a sub process within another process
- Independent: No dependency between processes or objects

In order to document the sequence of processes and object transitions, the following definition will be applied: The earlier the process needs to be started, the higher the rank that will be linked to that process. In figure 20 it is shown how this applies to all sequence types identified above and explained for processes. The sequential case is obvious where the first started process will get rank 1 (between brackets), the latter 2. In the parallel case it cannot be predicted which process will be started first and they both get the same rank, 1 in this case. In the embedded case the process which is started first, the embedding process, gets the highest rank. Finally, the independent case is treated quite the same as the parallel case. However, because these processes can act independently, they are not causing dependencies and they are of lesser interest. Therefor the lowest rank of all processes is given to them. In the figure below you can see the ranks between brackets. These rules are called the timing dependency rules which are also applicable for object transitions. In the next two chapters it will be demonstrated how these dependency rules can be applied in order to perform dependency analysis for processes and object transitions.

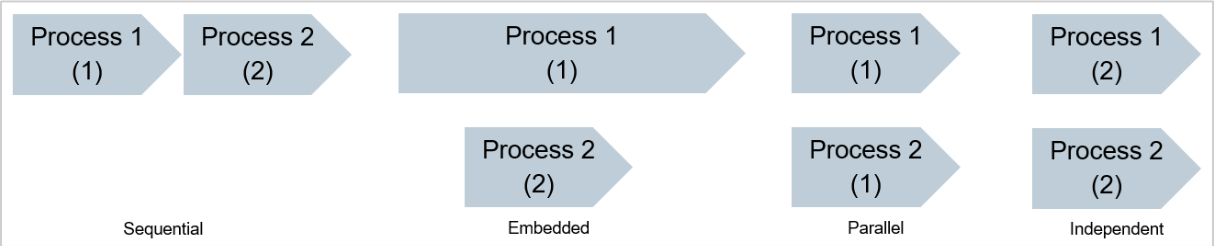

Figure 20 - Dependency timing cases

3.7.1. Building block 7-1: Process time dependency analysis

As mentioned already, the first timing dependency is the dependency of the process under analysis on its required objects getting into their required state. The timing and dependencies of the objects related to the process under analysis are documented by their rank using the timing dependency rules.

Starting from the Process object state matrix, the Process dependency matrix can be created by replacing the names of the object states by their rank. This rank is determined using the timing dependency rules. The Process dependency matrix is shown below in table 5.

Object dependencies (rank)	**Object A**	**Object B**	**Object C**
Processes			
Process X	1		
Process Y		1	2
Process Z		1	1

Table 5 - Process dependency matrix process X, Y and Z

In the Process dependency matrix in table 5 it is shown that Process Y is dependent on the sequential change of Object B and Object C. Where Object B needs to change first before Object C can obtain its required state.

3.7.2. Building block 7-2: Object time dependency analysis

The second timing dependency is the dependency of object states on multiple processes. This can be analyzed in the same way as the Process time dependency analysis using the timing dependency rules. However, in this case starting from the Object transition matrix.

In the Object transition matrix, the processes responsible for the object transition can be ranked using the timing dependency rules. This by adding the rank next to the process in brackets. This is shown for Object C in table 6. Later on this matrix can be refactored in order to show only the ranks of the processes. This by putting the related processes in the horizontal direction, the resulting object states in vertical direction and the ranks of the related processes in the cells.

Object C To state	Initial	Created	Ready	Released	Loaded
Object C From State					
Initial					
Created			Process A (1) Process D (2)		
Ready				Process E (1)	
Released					Process F (1)
Loaded					

Table 6 - Object transition matrix, Object C

Refactoring the object transition matrix gives the object dependency matrix shown in table 7.

Processes (rank)	Process A	Process B	Process C	Process D	Process E	Process F
Object C To State						
Created						
Ready	1			2		
Released					1	
Loaded						1

Table 7 - Object dependency matrix, Object C

In the object dependency matrix above you can see that the “Ready” state of object C is caused by sequentially executing Process A and then Process D. This also has to be done for object B of process Y under analysis.

3.8. Building block 8: End-to-end time dependency analysis

The full end-to-end time dependency analysis can be obtained by combining the process time dependency analysis and the object time dependency analysis. The process time dependency analysis delivers the Process dependency matrix. This matrix shows the sequence of the required object state transitions related to the process under analysis. The process dependency matrix of process Y is repeated in table 8.

Objects	Object A	Object B	Object C
Processes			
Process X	1		
Process Y		1	2
Process Z		1	1

Table 8 - Process dependency matrix Process Y

The object time dependency analysis delivers the object dependency matrix. This matrix shows the sequence of the processes which cause an object state transition. The sequence of the processes which cause the state transitions of Object B and C related to process Y, are repeated in the Object dependency matrices in table 9 and 10.

Processes	Process A	Process B	Process C	Process D	Process E
Object B To State					
Created					
Ready					
Released		**(1)**-1			
Loaded					

Table 9 - Object dependency matrix Object B

Processes	Process A	Process B	Process C	Process D	Process E
Object C To State					
Created					
Ready	**(2)**-1			**(2)**-2	
Released					
Loaded					

Table 10 - Object dependency matrix Object C

The information in the Process dependency matrix shows that the state transition of Object B needs to happen before Object C (Object B has the highest rank). The rank of an object in the process dependency matrix can be added to the object dependency matrix of that object. This is shown in bold in the object dependency matrices above, where rank 1 of object B and rank 2 of Object C, are added to their respective object dependency matrix. In this way, each process in the object dependency matrices gets a sequence which is formatted as (X)-Y. Where X is the rank coming from the Process dependency matrix, which is the sequence of the object state change. And Y is the rank of the process that causes the state change, coming out of the object dependency matrix.

The final end-to-end dependency matrix is provided by listing all processes in vertical direction and the process under analysis in horizontal direction. The sequence number of each process coming from the object dependency matrices, can then be filled in next to the process in vertical direction in the end-to-end dependency matrix. In our example, for process Y, the sequence numbers of process A, B and D coming from the object dependency matrices can be filled in. This results in the end-to-end dependency matrix as shown below in table 11, where the results for process Y are filled in. Finally, the result for the process under analysis can be obtained by sorting the column of the process under analysis, process Y in our case, by sequence.

Processes	Process Y
Process A	2-1
Process B	1-1
Process C	
Process D	2-2
Process E	

Table 11 - End-to-End dependency matrix

Sorting the column of process Y provides the dependencies and sequences as shown in table 12. Table 12 shows the final dependency result for Process Y. It shows that Process Y is sequentially dependent on Process B, then on process A and finally on Process D. This means that before process Y can be executed, first Process B, then Process A and finally Process D need to be executed.

Processes	Process Y	Object	Process
Process B	1-1	1	1
Process A	2-1	2	1
Process D	2-2	2	2

Table 12 - Process dependency result, Process Y

An important part of the end-to-end time dependency analysis is the process instance verification. This needs to be performed in case a process appears multiple times in the process dependency result like process B in table 13. The instance verification entails that it needs to be verified whether all identical processes in the result are actually the same instance of that process. If yes, only one instance of the process needs to be retained in the result because only one instance of the process is actually executed. If no, the process is actually executed multiple times and every instance needs to be in the dependency result. In our example in table 13, process B changes objects with different object sequences and thus different object instances. Therefor process B needs to be executed twice and cannot be left out of the result. In most cases not only the object sequence has to be taken into account, but also the manufacturing context. Especially when the related objects have the same sequence. As a best practice, when a process appears multiple times in the process dependency result, the manufacturing context of each possible instance of the process needs to be analyzed.

Processes	Process Y	Object	Process
Process B	1-1	1	1
Process A	2-1	2	1
Process D	2-2	2	2
Process B	2-2	2	2

Table 13 - Process dependency result, multiple processes

3.9. Building block 9: Global plant objects

One of the research questions stated the following. Which global business objects, related to the CPG plant but not to a manufacturing operation, need to considered? Because the dependency discovery method does not discriminate between object types, global business objects are implicitly part of the analysis. Global business objects are documented during business and plant object identification.

4. The dependency discovery method steps

Finally here all steps to be executed which form the dependency discovery method are summarized:

1. Select process model
2. Determine all plant product families
3. Select product family under analysis
4. Select process under analysis: going backwards through the process model
5. Identify plant objects
6. Identify business objects
7. Perform process to object state mapping
8. Determine object lifecycles
9. Perform object transition analysis
10. Perform process time dependency analysis
11. Perform object time dependency analysis
12. Perform end-to-end time dependency analysis

The dependency discovery method as shown above needs to be executed going through 2 loops. The first loop exists when a product family is selected in step 3, all processes of the process model have to be analyzed for that family going from step 4 until step 12. When all processes have been analyzed, the next product family can be selected in step 3 and again all processes need to be analyzed for that specific product family.

DEMONSTRATION

In this part the usage of the dependency discovery method will be demonstrated. The method will be demonstrated by analyzing a number of processes of a CPG plant. In this case it will be a generic CPG plant. The generic CPG plant is specifically defined for this master dissertation in order to be able to demonstrate the dependency discovery method.

1. The generic CPG plant

The generic CPG plant which is shown below, has been defined using the constraints (Kallrath, 2002) used by planning and scheduling. The model of the generic CPG plant acts as a complete CPG manufacturing plant including all its processes. The purpose of the generic CPG plant is to serve as a generic example and test environment for the demonstration of the dependency discovery method.

The generic CPG plant model is shown below, where raw materials are stored in storage 1, 2 and 3. These raw materials can have any pack type, bulk or packed. The raw materials are used in 2 production processes of 2 product families where each production process is having its own and shared raw material (RM) storage. The production processes also have their own and shared production machines. At the end of the production process the produced semi-finished (SF) goods are stored for 2 possible purposes: make to stock (MTS) or make to order (MTO). These two types indicate two types of storage facilities. The produced semi-finished goods can then be turned into finished goods (FG) by packing machines which support different pack types. After that, the finished goods can be stored in separate or shared storage facilities.

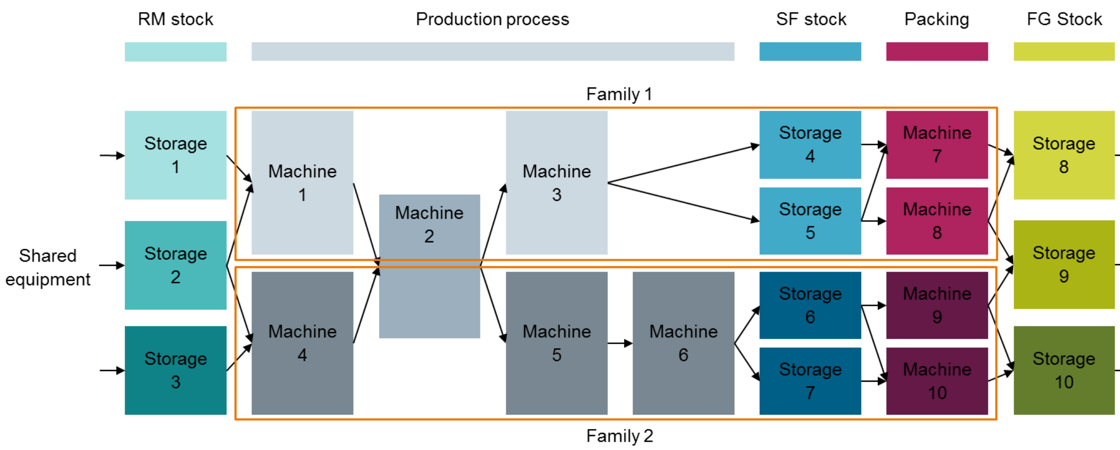

Figure 21 - Generic CPG plant model

2. Discovering process dependencies

Within this section the use of the dependency discovery method will be demonstrated by executing the method for a number of processes.

Initially, the first steps of the dependency discovery method which are common for all processes are shown. These steps are:

1. Select process model: the ISA 95 model will be used
2. Determine all plant product families: these are determined in the generic plant model
3. Select product family under analysis: family 1 of the generic plant model will be analyzed

The next step in the dependency discovery method is to select the next process under analysis. The next process is identified by going backwards through the processes of the ISA 95 model.

4. Select process under analysis: the last process is outbound inventory tracking (See figure 18)

In the next paragraphs the specific steps of the dependency discovery method which have to be executed for each process under analysis will be demonstrated. These steps are:

5. Identify plant objects
6. Identify business objects
7. Perform process to object state mapping
8. Determine object lifecycles
9. Perform object transition analysis
10. Perform process time dependency analysis
11. Perform object time dependency analysis
12. Perform end-to-end time dependency analysis

2.1. Outbound inventory tracking

The Outbound inventory tracking process is the first identified process when going backwards through the ISA 95 model (See figure 18). Outbound inventory tracking is the activity that manages information of material movements related to outbound shipment orders. An outbound shipment order exists out of one or more material movements, where each material move has a dataset related to it. The dataset contains the start and end time of a movement and updates on lot and sub-lot quantities. Outbound inventory tracking manages those datasets of material movements related to outbound shipment orders.

2.1.1. Identify plant objects

There are no plant objects involved in order to execute the outbound inventory tracking process.

2.1.2. Identify business objects

Following business objects are related to the outbound inventory tracking process:

- Shipment order
- Material Move(s)
- Material Move(s) data

An outbound shipment order contains one or more material moves. Each material move has its related material move data like source, destination, start time and end time.

2.1.3. Perform process to object state mapping

The object states which are required in order to be able to execute the outbound inventory tracking process are indicated in the table below.

Objects	Shipment order	Material Move	Material Move Data
Outbound inventory tracking	Completed	Completed	Collected

Table 14 - Outbound inventory tracking process object state mapping matrix

The outbound inventory tracking process requires that all material moves of an outbound shipment order have been completed. When all material moves related to the shipment order have been completed, the shipment order gets the completed state as well. The outbound inventory tracking process also requires that all material move data has been collected.

2.1.4. Determine object lifecycles

The previous state of the outbound shipment order and the material move object is "In progress". The previous state of the material move data object is "Created". The actual analysis of the lifecycles of the objects can be found in appendix.

2.1.5. Perform object transition analysis

The matrices which show the object transition analysis for all involved objects can be found below. The matrices show that the outbound shipment order and the material move object transition from the "In Progress" to the "Completed" state during the Outbound inventory execution process. They also show that the material move data object transitions from the "Created" to the "Collected" state during the Outbound inventory data collection process.

Shipment order To state	Completed
Shipment order From State	
In Progress	Outbound inventory execution

Table 15 - Outbound inventory tracking shipment order object transition analysis

Material Move To state	Completed
Material Move From State	
In Progress	Outbound inventory execution

Table 16 - Outbound inventory tracking Movement object transition analysis

Material Move data To state	Collected
Material Move data From State	
Created	Outbound inventory data collection

Table 17 - Outbound inventory tracking Movement data object transition analysis

2.1.6. Perform process time dependency analysis

In order to start the Outbound inventory tracking process, the related objects need to be in their required state. The sequence of the required states is shown in table 18.

Objects	Shipment order	Material Move	Material Move Data
Outbound inventory tracking	2	1	2

Table 18 - Process dependency matrix Outbound inventory tracking

First, all material moves have to be completed before material move data can be collected. The completion of all material moves implies that the shipment order is completed as well.

2.1.7. Perform object time dependency analysis

The state transition of each object is caused by only one process (see object transition analysis). Each transition process will Therefor have 1 as rank.

Shipment order object	Outbound inventory execution
Completed	**(2)**-1

Table 19 - Outbound inventory tracking shipment order object dependency matrix

Material Move object	Outbound inventory execution
Completed	**(1)**-1

Table 20 - Outbound inventory tracking Movement object dependency matrix

Material Move data object	Outbound inventory data collection
Collected	**(2)**-1

Table 21 - Outbound inventory tracking Movement data object dependency matrix

In order to support the end-to-end time dependency analysis, the object ranks (1 and 2) coming from the process dependency analysis have already been added into the matrices in bold.

2.1.8. Perform end-to-end time dependency analysis

Combining all information above provides the following end-to-end time dependency analysis. It shows the sequence of processes on which Outbound inventory tracking is dependent on.

Processes	Outbound inventory tracking	Object	Process	Object name
Outbound inventory execution	1-1	1	1	Material Move
Outbound inventory execution	2-1	2	1	Shipment order
Outbound inventory data collection	2-1	2	1	Material Move data

Table 22 - End-to-end dependency matrix outbound inventory tracking

In the end-to-end dependency matrix above, the outbound inventory execution process is shown twice. This is due to its relation with the material move and shipment order object. These objects are updated during a single instance of the outbound inventory execution process.

The shipment order only transitions to completed when the last material move is completed during the outbound inventory execution process. This explains the different object sequences. It can be concluded that outbound inventory tracking is dependent on sequentially executing one instance of the outbound inventory execution process and then the outbound inventory data collection process.

2.2. Outbound inventory data collection

Going backwards through the ISA 95 model, Outbound inventory data collection is found as the next process to be analyzed. Outbound inventory data collection is the process which collects all information of executed material moves related to an outbound shipment order. The collected material move information will be used by the outbound inventory tracking process where this information is further managed.

2.2.1. Identify plant objects

There are no plant objects involved in order to execute the outbound inventory data collection process.

2.2.2. Identify business objects

Following business objects are related to the outbound inventory data collection process:

- Shipment order
- Material Move(s)
- Material Move(s) data

Again, outbound shipments have one or more material moves with their related material move data.

2.2.3. Perform process to object state mapping

Following object states are required to execute the outbound inventory data collection process.

Objects	Shipment Order	Material Move	Material Move Data
Outbound inventory data collection	Completed	Completed	Created

Table 23 - Outbound inventory data collection process object state matrix

In order to collect all material move data, all material moves of the shipment order must be completed. The completion of the material moves enables that all related material move data can be created. The completion of all material moves also implies that the shipment order is completed as well.

2.2.4. Determine object lifecycles

The previous state of the outbound shipment order and the material move object is "In progress". The previous state of the Material move data object is "Initial". The full object lifecycles of the related objects can be found in appendix.

2.2.5. Perform object transition analysis

Below you can find the tables which show the object transition analysis of all involved objects. The shipment order and the material move object transition from the "In Progress" to the "Completed" state during the outbound inventory execution process. The material move data object transitions from "Initial" to the "Created" state during that same process.

Shipment order To state	Completed
Shipment order From State	
In Progress	Outbound inventory Execution

Table 24 - Outbound inventory data collection shipment order transition analysis matrix

Material Move To state	In Completed
Material Move From State	
In Progress	Outbound inventory execution

Table 25 - Outbound inventory data collection Movement transition analysis matrix

Material Move data To state	Created
Material Move data From State	
Initial	Outbound inventory execution

Table 26 - Outbound inventory data collection Movement data transition analysis matrix

2.2.6. Perform process time dependency analysis

In order to start the outbound inventory data collection process, the related objects need to be in their required state. The sequence of the required states is shown in table 27.

Objects	Shipment order	Material Move	Material Move Data
Outbound inventory data collection	2	1	2

Table 27 - Process dependency matrix Outbound inventory data collection

First, all material moves have to be completed before material move data can be created. The completion of all material moves also implies that the shipment order is completed as well.

2.2.7. Perform object time dependency analysis

Because there is only one process for each object to cause its state change, each process will have 1 as rank.

Shipment order	Outbound inventory execution
Completed	**(2)**-1

Table 28 - Outbound inventory data collection shipment order object dependency matrix

Material Move	Outbound inventory execution
Completed	**(1)**-1

Table 29 - Outbound inventory data collection Movement object dependency matrix

Material Move data	Outbound inventory execution
Created	**(2)**-1

Table 30 - Outbound inventory data collection Movement data object dependency matrix

In order to support the end-to-end time dependency analysis, the object ranks (1 and 2) coming from the process dependency analysis are already added into the tables in bold.

2.2.8. Perform end-to-end time dependency analysis

Combining all information above provides the following end-to-end time dependency analysis. This shows the processes and their sequences where outbound inventory data collection is dependent on.

Processes	Outbound inventory data collection	Object	Process	Object name
Outbound inventory execution	1-1	1	1	Material Move
Outbound inventory execution	2-1	2	1	Material Move data
Outbound inventory execution	2-1	2	1	Shipment order

Table 31 - End-to-end dependency matrix outbound inventory data collection

Outbound inventory execution is shown 3 times. This is due to its relation with the material move, material move data and the shipment order object. It can be concluded that the outbound inventory data collection process is only dependent on one instance of the outbound inventory execution process, which simultaneously sets all three objects. The completion of the material moves and the shipment, as well as the creation of the material move data, happen during one instance of the outbound inventory execution process.

The shipment order only transitions to completed when the last material move is completed. Also, it is only possible to create the material move data when all material movements are completed. This explains the different object sequences.

2.3. Outbound inventory Execution

The next process in the ISA 95 model is the outbound inventory execution process. The outbound inventory execution process performs the actual material movements related to an outbound shipment order.

2.3.1. Identify plant objects

The generic plant model can be used to determine the involved plant objects. In this case the finished goods storage is the most important object because it contains the source tank of the material movements.

There are multiple types of material movements possible in a CPG plant, such as bulk or packed material movements for example. Each material movement type requires a different set of plant objects to perform the actual material movement. A bulk material movement may require a pump, where a packed material movement may require a forklift or an operator. Because each movement type involves a different set of plant objects, each movement type will have a different implementation of the outbound inventory execution process. Each of those implementations of the outbound inventory execution process needs to be analyzed individually. For this demonstration, only a bulk loading process, where a bulk liquid product is loaded to a carrier using a pump, is assumed. The bulk loading material movement requires following plant objects:

- The source tank: where the product is loaded from
- The finished product lot: where the amount of product to be loaded is taken from
- The load equipment (pump, tank and load arm): which is used to perform the load
- The load carrier: where the product is loaded on
- The field operator: who is going to perform the load

The load equipment object contains all equipment needed to perform the material move. This means all equipment needed to provide the complete route from source tank to carrier, including the source tank.

2.3.2. Identify business objects

Following business objects are related to the outbound inventory execution process:

- Shipment order
- Material Move(s)

These objects are again the same objects as seen earlier. However, the material move data object is not required because it is created by the outbound inventory execution process.

2.3.3. Perform process to object state mapping

In order to execute the outbound inventory execution process, the involved business and plant objects require the states as mentioned in the tables below.

Objects	Shipment Order	Material Move
Outbound inventory execution	Dispatched	Scheduled

Table 32 - Outbound inventory execution process object state matrix (business objects)

Objects	Source tank	Product lot	Equipment	Carrier	Operator
Outbound inventory execution	Ready	Available	Ready	Available	Available

Table 33 - Outbound inventory execution process object state matrix (plant objects)

The table of the business objects shows that the outbound shipment order needs to be in the "dispatched" state. This is a state where the order contains all needed information in order to be executed and is made available to operators for execution. One of the preconditions to be able to put the outbound shipment order to the dispatched state, is that all related material moves are in the "scheduled" state. The scheduled state of the material movements is for this reason also required to execute the outbound inventory execution process.

The table of the plant objects shows that the source tank and the equipment need to be "ready" to be used. It also shows that the product, the carrier and the operator to start the actual material movement need to be "available".

2.3.4. Determine object lifecycles

When an outbound shipment order is released by the business, it gets the "released" state. The material moves related to the shipment order will also have the released state and will not have a schedule yet. In order to be able to dispatch the outbound shipment order, it needs to have the "scheduled" state. To obtain the scheduled state, it is required that all related material moves have a schedule. When all material moves related to the shipment order have been "scheduled", the related shipment order will also obtain the "scheduled" state. If the outbound shipment order is "scheduled", it can be dispatched to the operators in order to be executed. In the explanation above, it can be noticed that the previous state of the material movements is the "Released" state and the previous state of the shipment order is the "Scheduled" state.

Before a source tank is "ready" for use, it can have multiple possible other states: "out of service" for maintenance or "in use" for production or loading activities. It will be assumed that the previous state of the source tank is "in use" because material coming from production is being put into the tank. This implies that the product lot which is being put into the tank still has the "in production" state. The equipment, the carrier and operator can also have multiple previous states like the source tank. It will be assumed that the needed equipment is "in use" and that the carrier and operator are "unavailable". Further analysis in this demonstration is based on these assumed states. Normally the analysis needs to be performed for each combination of possible previous states of the involved objects, which will result in multiple end-to-end dependency matrices for the process under analysis.

2.3.5. Perform object transition analysis

Below you can find the tables which show the object transition analysis of all involved objects.

Shipment order To state	**Dispatched**
Shipment order From State	
Scheduled	Outbound inventory dispatching

Table 34 - Outbound inventory execution shipment order transition analysis matrix

The shipment order object is transitioned to the dispatched state by the outbound inventory dispatching process.

Material Move To state	**Scheduled**
Material Move From State	
Released	Outbound inventory scheduling

Table 35 - Outbound inventory execution material move transition analysis matrix

The material move object is transitioned to the scheduled state by the outbound inventory scheduling process, which provides the material move schedule. The material move schedule consist out of a timeslot for the needed equipment, operator and carrier to move the requested product.

Source tank To state	**Ready**
Source tank From State	
In use	Production quality test tracking Production tracking Outbound inventory tracking

Table 36 - Outbound inventory execution source tank transition analysis matrix

In order to be able to load the product, the required product needs to be put into the source tank. Therefor the product needs to be produced and quality tested first. This is finalized by the production tracking and production quality test tracking process. The production quality test tracking process is the quality test tracking process of the quality activity model which is being called from the production execution process (see figure 18). When production of the product is finished, the product can be moved into the source tank. The movement into the tank is finalized by the outbound inventory tracking process.

Product lot To state	Available
Product lot From State	
In production	Production quality test tracking Production tracking

Table 37 - Outbound inventory execution Product lot transition analysis matrix

Because the product lot is related to the required product that needs to be put into the source tank, it also requires the production tracking and production quality test tracking process as mentioned above for the source tank object.

Equipment To state	Ready
Equipment From State	
In use	Outbound inventory tracking

Table 38 - Outbound inventory execution Equipment transition analysis matrix

The required loading equipment needs to finish any ongoing loading activity which is not related to the shipment order. These loading activities are finalized by the outbound inventory tracking process.

Carrier To state	Available
Carrier From State	
Unavailable	Outbound inventory scheduling

Table 39 - Outbound inventory execution Carrier transition analysis matrix

Operator To state	Available
Operator From State	
Unavailable	Outbound inventory scheduling

Table 40 - Outbound inventory execution Operator transition analysis matrix

Not only the material move itself needs to be scheduled, but also the operator that needs to perform the move and the carrier which needs to be loaded. The carrier and operator also need to be scheduled by the outbound inventory scheduling process which makes them available.

2.3.6. Perform process time dependency analysis

In order to start the outbound inventory execution process, the related objects need to be in their required state. The sequence of the required states is shown in table 41 and 42.

Objects	Shipment Order	Material Move
Outbound inventory execution	2	1

Table 41 - Process dependency matrix Outbound inventory execution (business objects)

Objects	Source tank	Product lot	Equipment	Carrier	Operator
Outbound inventory execution	3	3	4	1	1

Table 42 - Process dependency matrix Outbound inventory execution (plant objects)

The sequence of the object states continues over both tables. The sequences are derived from the following facts. The operator, carrier and material moves need to be scheduled first. As already mentioned, the shipment can only be dispatched if it has the scheduled state. The shipment order will only receive the scheduled state if all related material moves are scheduled. When the shipment order is dispatched, the produced product lot needs to be put into the source tank in order to execute the material movements. These movements can only be performed when the load equipment is finally ready.

2.3.7. Perform object time dependency analysis

In the matrices below you can find the object dependency analysis which is based on the object transition analysis. The processes which cause the transitions have been indicated with their rank. When multiple processes are involved, a small elaboration of the ranks will be provided below the involved matrix. The ranks coming from the process dependency matrices have also been added in bold in the matrices below in order to support the end-to-end dependency analysis.

Shipment order	Outbound inventory dispatching
Dispatched	**(2)**-1

Table 43 - Outbound inventory execution shipment order object dependency matrix

Material move	Outbound inventory scheduling
Scheduled	**(1)**-1

Table 44 - Outbound inventory execution material move object dependency matrix

Source Tank	Production quality test tracking	Production tracking	Outbound inventory tracking
Ready	**(3)**-1	**(3)**-2	**(3)**-3

Table 45 - Outbound inventory execution Source tank object dependency matrix

In order that the source tank can become ready for use, the product which needs to be loaded from the source tank, needs to be produced first. While producing, the product needs to pass the quality test, which is finalized by the production quality test tracking process. When the product has passed the quality test, production can be finalized by the production tracking process. After production, the product can be put into the source tank which is finalized by the outbound inventory tracking process.

Product lot	Production quality test tracking	Production tracking
Available	**(3)**-1	**(3)**-2

Table 46 - Outbound inventory execution Product lot object dependency matrix

The product lot needs to be produced in order to be shipped. As already explained in the source tank dependency analysis, the product needs to be quality tested during production and production is finalized with the production tracking process.

Equipment	Outbound inventory tracking
Ready	**(4)**-1

Table 47 - Outbound inventory execution Equipment object dependency matrix

Carrier	Outbound inventory scheduling
Available	**(1)**-1

Table 48 - Outbound inventory execution Carrier object dependency matrix

Operator	Outbound inventory scheduling
Available	**(1)**-1

Table 49 - Outbound inventory execution Operator object dependency matrix

The ranks of the objects which have only one process causing the object transition obviously do not need any further explanation.

2.3.8. Perform end-to-end time dependency analysis

Combining all information above provides the following end-to-end timing analysis.

Processes	Outbound inventory execution	Object	Process	Object name
Outbound inventory scheduling	1-1	1	1	Operator
Outbound inventory scheduling	1-1	1	1	Carrier
Outbound inventory scheduling	1-1	1	1	Material move
Outbound inventory dispatching	2-1	2	1	Outbound shipment
Production quality test tracking	3-1	3	1	Source tank
Production quality test tracking	3-1	3	1	Product lot
Production tracking	3-2	3	2	Source tank
Production tracking	3-2	3	2	Product lot
Outbound inventory tracking	3-3	3	3	Source tank
Outbound inventory tracking	4-1	4	1	Equipment

Table 50 - End-to-end dependency matrix outbound inventory execution

The end-to-end dependency matrix provides the following conclusions. First of all the operator, carrier and material move need to be scheduled. This is performed in one time and thus by one instance of the outbound inventory scheduling process. When the material move is scheduled, the shipment order is implicitly scheduled as mentioned in the object lifecycle analysis. When the outbound shipment is scheduled, it can be dispatched in order to be executed.

Before the material moves can be actually executed, the product needs to be available and thus produced. This is done and finalized by the production quality test tracking and the production tracking process. The production quality test tracking and production tracking process change both the source tank and product lot objects during the same sequence. The context shows that the product lot which needs to be available for the load, is the same product lot which is produced and which needs to be moved into the source tank after production. Therefor it can be concluded that the production quality test tracking and production tracking process are both one single instance of that process. One instance of the production tracking and production quality test tracking process can be left out of the end-to-end dependency matrix.

Also the outbound inventory tracking process is shown twice in the end-to-end dependency matrix. The first instance is used to put the product into the tank and the second instance to move material out of the tank. Therefor both instances need to remain in the end-to-end result.

The final result of the end-to-end dependency matrix is shown below where the unnecessary outbound inventory scheduling, production tracking and production quality test tracking processes are left out.

Processes	Outbound inventory execution	Object	Process	Object name
Outbound inventory scheduling	1-1	1	1	Operator, Carrier, Material Move
Outbound inventory dispatching	2-1	2	1	Outbound shipment
Production quality test tracking	3-1	3	1	Product lot, Source tank
Production tracking	3-2	3	2	Product lot, Source tank
Outbound inventory tracking	3-3	3	3	Source tank
Outbound inventory tracking	4-1	4	1	Equipment

Table 51 - Final End-to-end dependency matrix outbound inventory execution

As a final conclusion, it can be stated that in order to be able to execute the outbound inventory execution process, following dependencies have to be taken into account. First the operator, the carrier and the actual material move need to be scheduled. If the material moves are scheduled, the shipment can be dispatched. When the outbound shipment is dispatched, the product can be made available in the source tank, but therefor it needs to be produced and quality tested first. When the product has been produced, it can be put into the source tank. If the product is finally in the source tank, it can be moved to the carrier if the needed equipment is available and is not executing any other material movement.

3. Evaluation

The main objective which was set for the design of the dependency discovery method was that the method should have a small footprint. This small footprint refers especially to the reuse of existing information and methods as much as possible. Looking at the method, it can be concluded that it meets this requirement by reusing: the ISA 95 model, value stream mapping and object lifecycle analysis.

Looking at the new elements which have been added in order to create the dependency discovery method, it can be stated that they also meet the small footprint objective. This due to the fact that these new elements can be executed with a limited set of prerequisites. This means that no new big models need to be created. The new elements are the following:

- Process to object state mapping: using VSM and ISA 95 model
- Object transition analysis: using VSM and ISA 95 model
- Process time dependency analysis: using ranks of timing dependency rules
- Object time dependency analysis: using ranks of timing dependency rules
- End-to-end time dependency analysis: using ranks of timing dependency rules

Next to each new element it is indicated which information it uses. This shows again the small footprint of the new information and models which are needed for the dependency discovery method. The main new analysis method is provided by the timing dependency rules. These rules are lightweight and easy to use and are used during the time dependency analysis.

However it is shown that the dependency discovery method has a small footprint it also has to deal with some challenges. These challenges are coming from the amount of data it is generating. Even if only the end-to-end dependency results are retained, the method results in a large number of matrices. The method currently does not focus on aggregating these matrices into one big model. This results in the challenge to get a good overview of the most important dependencies.

CONCLUSION AND DISCUSSION

The purpose of this master dissertation was to construct and demonstrate a method to approach the discovery of process and activity dependencies. Within the previous chapters the process and dependency discovery method has been explained and demonstrated. Therefor it can be stated that the goal of this master dissertation has been achieved.

During the literature review, the processes of the CPG industry which need to be analyzed have been identified. These are the processes coming out of the ISA 95 activity models. Also, it has been identified in which order these processes need to be analyzed by going backwards through the ISA 95 models. The research has shown that process dependencies can be detected by analyzing the object transitions which are needed in order to be able to start the process under analysis.

Process dependencies are described by first identifying the objects involved with the process under analysis. Business and plant objects can be identified due to their relation with the process under analysis or using value stream mapping. The identified objects have a sequence in which they need to be put into their required state. This sequence is the first part of the description of the dependency. The second part is coming from the sequence of processes which need to be executed to obtain a specific object transition. The combination of both sequences, which has the format X-Y, describe the process dependency. Where X is the sequence of each object transition and Y the sequence of the process causing the object transition. Because the timing dependency rules, on which the analysis of object transitions is based, do not discriminate between object types, it can be stated that global business objects are supported by the method.

However it is shown that the dependency discovery method is lightweight which was the objective that had been set, there is still room for improvement. One of the main topics which can be improved is to provide an overview of all process dependencies. In this way an analysis of the bottleneck processes and process dependencies can be more easily performed. The main issue for this overview is the large amount of data which is generated by the method.

Related to the large amount of data which is generated by the dependency discovery method, is the issue with objects that might have multiple previous states. Due to this, the time dependency analysis needs to be performed for each combination of possible previous states of the involved objects for the process under analysis. This ensures that even more data is generated and that there are mutiple end-to-end dpendency matrices for the process under analysis. The number of matrices is dependent on the number of previous states of the involved objects. Also it is possible that there are multiple implementations of the process under analysis. For example the inventory execution process which can have an implementation for bulk or packed materials. This implies that the dependency analysis needs to be performed for each implementation of the process and even more end-to-end dependency matrices are generated.

Having multiple end-to-end matrices of process dependencies for each process also makes it more difficult to determine bottleneck processes. This due to the fact that there will be multiple end-to-end matrices for each process dependent on the number of previous states or process implementations. As a final conclusion, it can be recommended that the next step is to build a general overview of the process dependencies, which aggregates all possible end-to-end matrices. Based on this overview a method could be created that shows how to further analyze process dependencies to detect bottleneck process.

REFERENCES

Braglia, Carmignani, & Zammori. (2006, 10 1). A new value stream mapping approach for complex production systems. *International Journal of Production Research, 44*, 3929-3952.

Convergence alimentaire. (n.d.). Retrieved from http://convergencealimentaire.info/

Euromonitor. (2017). *PACKAGED FOOD 2018 EDITION: NEW INSIGHTS AND SYSTEM REFRESHER.*

International Society of Automation. (2007). *ISA 95 - Part 3: Activity Models of Manufacturing.*

Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum, 24*, 219-250.

Langstrand, J. (2016). *An introduction to value stream mapping and analysis.* Linköping University.

Meyer, A., Pufahl, L., Fahland, D., & Weske, M. (2013). Modeling and Enacting Complex Data Dependencies in Business Processes. *Lecture Notes in Computer Science, 8094*, 171-186.

Mozota, B. B. (1998, 06). Structuring Strategic Design Management:Michael Porter's Value Chain. *Design management journal, 9*, 1-8.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45-78.

Pont, J.-P. D., & Azzaro-Pantel, C. (2014). New approaches in the process industry.

Rother, J., & Shook, M. (1999). *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda.* Brookline: Lean enterprises institutes.

Sanz, J. L., & Nandi, P. (2012). *Entity-centric operations modeling for business process management. A multi disciplinary review of the state-of-the-art.* IBM Research division.

SAP University Alliances. (2015). *Sales and Distribution (SD).*

Siemens. (2018, 1 08). *Siemens MOM*. Retrieved from http://siemens.com/mom

APPENDIX

1. Outbound shipment order

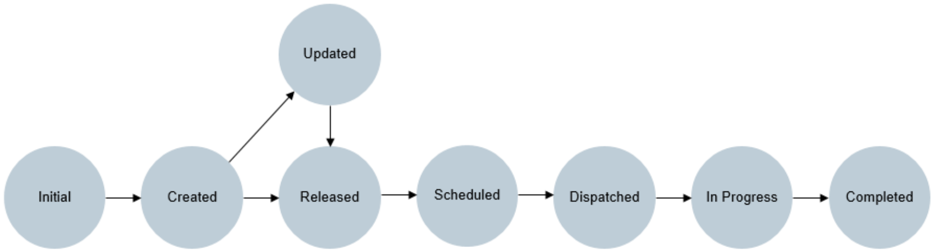

Figure 22 - Outbound shipment order object lifecycle

2. Material move

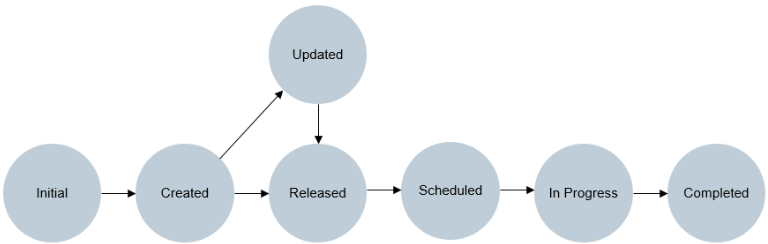

Figure 23 - Material move object lifecycle

3. Material move data

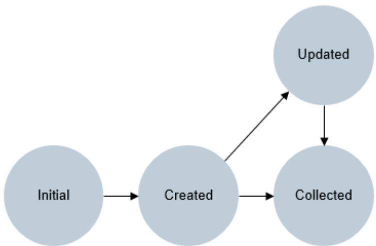

Figure 24 - Material move data object lifecycle

4. Source tank

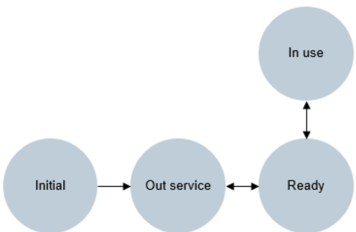

Figure 25 - Source tank object lifecycle

5. Product lot

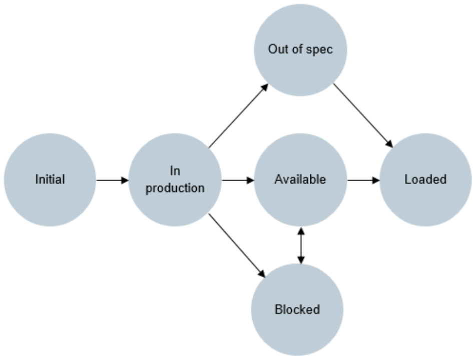

Figure 26 - Product lot object lifecycle

6. Load equipment

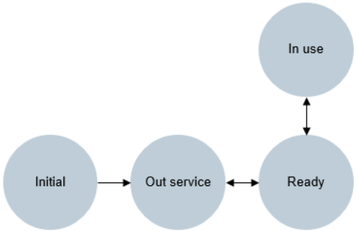

Figure 27 - Load equipment object lifecycle

7. Load carrier

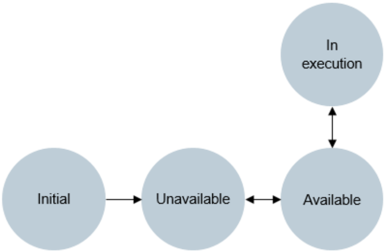

Figure 28 - Load carrier object lifecycle

8. Field operator

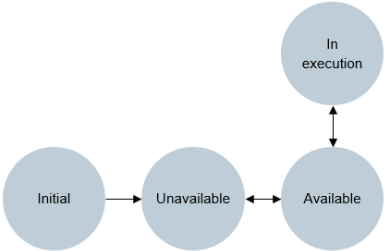

Figure 29 - Field operator object lifecycle

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Discover process dependencies of manufacturing operations in the CPG industry

Richting: **Master of Management-Business Process Management**
Jaar: **2018**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Verwimp, Frederik

Datum: **1/06/2018**