MSc Thesis Mechanical Engineering and Industrial Design Engineering

Improving information traceability in high-tech contract manufacturing environments

Eline Uiterweerd*[a]

[a] University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands

* Corresponding author. *E-mail address:* e.uiterweerd@student.utwente.nl

Abstract

In today's high-tech manufacturing landscape, increasingly complex and customised products are being made. OEMs often outsource the production of (parts of) their designs to contract manufacturers. These contract manufacturers are specialised in transforming an elaborate set of product specifications into a number of physical products, which requires machines and operators to work with extreme precision and within a narrow window of operation parameters. Because of the ambitious targets set by their clients, the high cost of poor quality and the relatively small production series, it is important for contract manufacturers to mature production processes as efficiently and quickly as possible in order to maintain their competitive position. Effective tracing and management of measurement data generated before, during and after the production activities is a fundamental prerequisite for acquiring insights to improve product and process designs. This study proposes a method for improving the traceability of information generated in high-mix, low-volume, high-precision and high-complexity contract manufacturing environments. After identifying the most significant conditions for achieving adequate information traceability, a framework of interlinking information contents is proposed. Furthermore, the research dives deeper into the necessary architecture of various IT systems in order to support this framework. A case study at NTS Norma Hengelo, a contract manufacturer that fits the described profile, is performed to validate the proposed information traceability method and supporting IT architecture.

Keywords: Information traceability; information management; HMLV; high-precision manufacturing; high-complexity manufacturing; IT architecture.

1. Introduction

Quality standards and customer demands are ever-increasing in manufacturing environments in the high-tech industry. Original equipment manufacturers (OEMs) outsource the production of (parts of) their products to contract manufacturers, which fabricate these complex products according to rigorous standards under highly specific conditions, all while maintaining the edge over their competitors. In order to stay competitive, it is essential that the development of the processes that are required to manufacture these products happens as efficiently as possible. This development process relies heavily on the use of collected information to aid stakeholders in making design decisions. The information contents that drive the various decision-making processes are often collected through a wide range of IT applications, each having their own scope and purpose. However, the process of information-driven development and engineering is often hampered by a variety of organisational issues. For example, different departments may unknowingly draw different, contradicting conclusions from the same information, or may not even have the same information to their disposal. Additionally, knowledge from one department is often lost in the transferal to a different department due to lacking documentation. Finally, it is often unclear if certain production targets (such as meeting a fixed delivery date or staying below a threshold for production costs) are attainable until the product is already in production. In other words, the verifiability of important decisions is limited in many cases.

Through mapping out the current data flows in the IT architecture of an existing contract manufacturer in the high-tech industry and conducting interviews with several stakeholders (see, for instance, appendices *Current IT Architecture of NTS Norma* and *Scoping the Case Study*), it became clear that many problems related to the information-driven development process can be reduced to the common denominator of insufficient information traceability. Transparency and traceability of information is essential in order to provide the context and structure to the information contents that is required for decision-making. The foundation for information provision is formed by the IT architecture of an organisation, which should facilitate stakeholders in underpinning and verifying decisions [15]. However, in many cases, this IT architecture has not been designed and maintained with adequate care and is the result of unbridled expansion, with many highly specific software packages having been added over time while old habits and processes have remained in place. The result is an IT landscape in which the overview of information flows is largely unknown and in which every department works within their own domain, making it difficult to trace information contents across departments and IT applications.

1.1. Manufacturing environment description

The topic of information traceability is a widely discussed concept within all kinds of engineering and manufacturing enterprises and is of importance within all phases of a product life-cycle. This research, however, focuses solely on a specific life-cycle phase and type of manufacturing environment. The production phase as observed in high-mix, low-volume contract manufacturers that make highly complex products for OEMs according to high precision and quality standards forms the overall scope of this paper and sets the context for information traceability.

Contract manufacturers are manufacturing enterprises first and foremost. Contract manufacturers do not design and manufacture their own products; rather, they manufacture products and parts that are designed by clients. Many OEMs, especially in high-tech industries like the aerospace and semiconductor industries, design exceedingly complex parts and products. However, with the growing complexity of their products, it becomes increasingly difficult for OEMs to perform every step of their product realisation within their own facility walls. The production of these products requires tools, skills, and expertise that the OEMs do not always possess; hence, such OEMs collaborate with contract manufacturers that manufacture and supply (parts of) their products [3]. This implies that while contract manufacturers receive most documentation that describes the product definition (such as product specifications, an engineering bill of materials, product drawings, CAD-files, standard procedures for assembly and cleanliness, etc.) from their customers, the contract manufacturers themselves are responsible for designing and executing the production process. The degree to which OEMs provide their suppliers with standards and instructions for the manufacturing of their products may vary. Different cooperation agreements between contract manufacturers and their customers include build-to-print (BTP) manufacturing, which sees the customer impose clear restrictions on e.g. the assembly procedures and allowed production methods, and build-to-spec (BTS) manufacturing, in which the customer merely provides a set of specifications which the product must fulfil [19].

In many cases, contract manufacturers have multiple customers and produce a wide range of highly complex and customised products in small batches, and adapt their manufacturing approach accordingly. High-mix, low-volume (HMLV) manufacturing is a production layout strategy that handles this type of production in an efficient manner. Unlike mass production, in which large quantities of products of the same type are manufactured in fixed production lines, HMLV production allows for flexible manufacturing routings and is often characterised by a facility layout that has similar production processes grouped together [18].

Very specific challenges with regards to information traceability and management in this particular type of manufacturing environment can be derived from the aforementioned circumstances. Since the product designs and specifications of clients evolve rapidly and new versions are made in quick succession, the product master data on which the contract manufacturers base their process design is not stable, but subject to constant change. Products are produced in small quantities and oftentimes need to fulfil high quality standards imposed by the customer, which leave very little room for process variations. The production of the intricate part designs often involves a multitude of complex (and, therefore, expensive) manufacturing operations; hence, the cost of part rejection due to poor quality is an especially important driver in optimising and stabilising the production process. It is thus essential that the measurement data recorded during the production process is utilised to its fullest extent; even more so because of the small production quantities. However, in order to draw meaningful conclusions from this data, it must be known exactly under what circumstances the data was generated: what machine settings were used? Which version of the part (based on a particular version of technical drawings and other specifications) was produced? Was the intended production equipment used? In other words, it must be known exactly which process inputs resulted in the recorded process outputs before one can draw any meaningful conclusions about the quality of the process and manufactured product; it must be possible to trace the recorded information back to its source.

1.2. Research aims and structure of this paper

This research aims to identify the most significant conditions for information traceability and, based on these conditions, propose a method to improve the traceability of information in HMLV, high-precision and high-complexity contract manufacturing environments. Section 2 provides a literature review in which the concept of information traceability is further explored. Furthermore, it is explained why information interlinking and structuring are essential to achieve traceability. Finally, the various components of the enterprise IT architecture in which production-related information is stored and managed are discussed, as well as the necessary integration thereof. Based on the outcomes of the literature review, Section 3 proposes a conceptual design for improving information traceability, supported by a proposal for a general IT architecture. Through the use of an ontology and a defined information structure deployed within an integrated IT architecture, it becomes possible to interlink and integrate information contents. This conceptual design is subsequently applied in a case study (Section 4) at NTS Norma Hengelo, a HMLV, high-precision and high-complexity contract manufacturer. Finally, Section 5 concludes the research and Section 6 provides recommendations for future work.

2. Literature Review

This literature review aims to provide more insight into the subject of information traceability, information management and the use of IT systems in a manufacturing context. Section 2.1 covers the definition of information traceability (Section 2.1.1) and the structuring of information contents (Section 2.1.2), which is an important prerequisite for effective information management. Subsequently, Section 2.2 describes the purposes (Section 2.2.1) and integration (Section 2.2.2) of the various IT systems that are involved in recording, storing and transmitting information.

2.1. *Information traceability and management in manufacturing environments*

Traceability is used as a risk management tool that is used to capture the history of a product across its life-cycle in terms of a product's properties and associated transformation throughout the production process [13, 14]. Product life-cycle information that is consistently recorded and easily accessible will result in a number of benefits, such as improved quality control and product compliance, faster product recalls, increased transparency of production processes [7, 10, 13, 14]. Additionally, accurately traced and structured information forms the basis of effective decision support in process optimisation [15]. While many concepts surrounding traceability originate from the food, medical and agriculture industries, its principles are starting to become more widely adopted throughout other industries as well [14].

This section reviews the terminology of information traceability and the prerequisites for implementing traceability systems within a manufacturing environment. After defining the key components of a traceability system, it is argued that concise structuring and formalising the captured information content is essential in understanding and using the traced information.

2.1.1. *Definition of information traceability*

While many definitions for (information) traceability exist across literature (see for instance [7]), the universal definition that is adopted here describes traceability as "the ability to access any or all information relating to that which is under consideration, throughout its entire life-cycle, by means of recorded identifications" [14]. In this definition, the phrase 'that which is under consideration' refers to a *traceable resource unit* (TRU). A TRU is a traceable object of a particular aggregation level; typically, a TRU is a trade unit, logistic unit, or a production unit. Examples include a type of raw material, a specific part, an assembled product, a production batch, or a shipment [10].

Within the framework of information traceability, a distinction is made between different types of traceability. Generally, there are three different aspects that distinguish one application of information traceability from another [7, 14]:

1. *Passive or active traceability*: while passive traceability refers to the act of providing better data visibility (by keeping historic records of TRU transformations), active traceability additionally aims to optimise and control processes in and between the different links of the supply chain.
2. *Backward or forward traceability*: backward traceability revolves around tracing a TRU back to its origin; for instance, it traces which components were used to create a certain assembly. This is considered as a 'top-down' approach for traceability and is used to, for instance, trace back quality problems of a product to its origin. Forward traceability does the exact opposite and 'tracks' a TRU forward through time in a 'bottom-up' manner (investigating in which higher-level assembly a specific component was incorporated). Forward traceability is especially useful when product recalls have to be made. Once a faulty batch of parts is found, it is possible to investigate quickly which products contain parts of this faulty batch and recall those products.
3. *Internal or external traceability*: internal traceability is concerned with keeping records of product transformations within the scope of one production process. External traceability has a wider scope, tracking information from multiple parties across the supply chain of the TRU (which may include suppliers and customers).

Since traceability systems clearly are concerned with recording information and linking information to specific products, there must be some general system components that are required to enable such activities. Fig. 1 displays two different traceability systems.

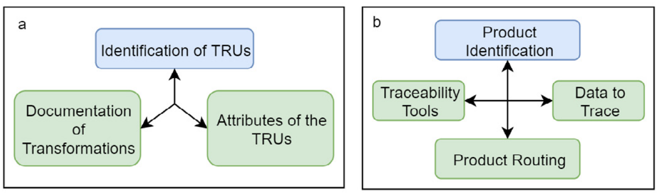


Fig. 1: A comparison between two traceability system models. Left: Olsen and Borit's three-component system [10]. Right: the four-pillar model by Regattieri et al. [13]. Schematic retrieved from [14].

Olsen and Borit [10] mention three general system components:

1. A mechanism for identifying TRUs; binding a unique identifier to a TRU.
2. A mechanism for documenting transformations. It is necessary to record what happens to a TRU throughout the production process.
3. A mechanism for recording TRU attributes and binding these to the TRU identifier, such as any documentation that was used to process and create the TRU.

Regattieri et al. [13] take a slightly different approach and suggest a four-pillar model:

1. Product identification. It is essential to identify the product and its properties. The bill of material (BOM) structure is used to condition the tracing system and to connect TRUs that ended up in the same parent product or assembly.
2. Data to trace. Apart from the product properties, it is required to establish which data needs to be recorded for a TRU.

3. Product routing. This is the mechanism that is responsible for recording the transformations of the TRU across its production process.
4. Traceability tools. This concerns all the required hardware for the traceability, such as physical part markings that link the unique identifier to the TRU (examples: barcodes and RFID tags).

Although both models have slightly different formulations, it is clear that a traceability system requires a number of general components, which are displayed in Fig. 2. First of all, it must be possible to link the physical TRU to a unique identifier (for instance by placing barcodes or RFID tags on the TRU or its packaging). This unique identifier will be used to link process inputs and process outputs (transformations) to the individual TRU. Hence, the linking of process inputs and outputs are two additional general functionalities. Here, process inputs concern all TRU product and production process properties. This includes all documentation that is required to produce the TRU. Examples include the BOM structure, technical drawings, machining code, routings and work instructions. Process outputs are the information contents related to the transformation of TRUs. This entails all information that was recorded during the production of the TRU, including measurement data, cycle times, assembly configurations (linked TRU identifiers).

In this context, it is useful to introduce the distinction between the product type (or 'product object') and the product instance. The process inputs together fall under the category of 'master data', i.e. data corresponding to the generic product type. Therefore, the product type can be seen as a virtual product definition [2, 9]. The product instance is the physical realisation of the product type. In a manufacturing context, the product type has a one-to-many relation to the product instances. Nigischer et al. compare this relation to the instantiations of a class in object-oriented programming; although all of the instances are based on the same product type, they all are independent from one another and may have different process outputs [9]. Hence, while multiple TRUs may have the same or highly similar process inputs, each TRU will have a unique set of process outputs. The collection of process inputs and outputs of a TRU together forms the configuration as-built for a TRU. The configuration as-built resembles all information associated with a unique product instance, specifying the unique configuration of the product instance and its corresponding production history.

Additionally, there must be some way to establish which degree of precision is required when tracing TRUs. For instance, the aggregation level of the TRU must be determined; it may not be necessary to identify individual parts. Instead, identifying TRUs at the level of production batches may suffice. Furthermore, the required traceable process inputs and outputs must somehow be established for each TRU. The traceability protocol identifies and describes these requirements.

To conclude, in order to acquire meaningful insights from a traceability system, it is necessary that the identification, properties and transformations of traceable resource units are linked

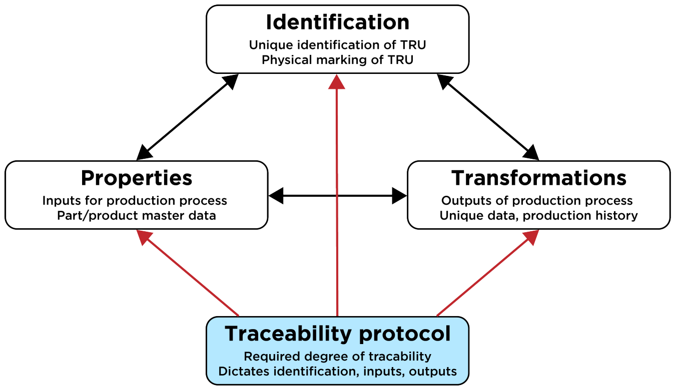


Fig. 2: The required components for a traceability system.

to one another. Integration and structuring of the information content of the traceability system is thus essential.

2.1.2. *Information structures and ontologies*

In order to better understand and apply the information content that is collected, it is necessary to present the information in a structured manner [8]. It is the processing of large quantities of information rather than the collection that forms a problem in companies, resulting in companies not utilising the information that is available and even losing insight into the internal processes due to the added complexity [16]. Using an information structure that allows for the identification of mutual relations between information entities and the adoption of different domains, viewpoints and filters on the information content helps to overcome these problems [8].

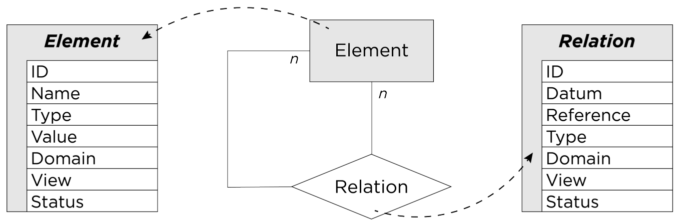


Fig. 3: The basic building blocks for information structures: elements (information content) and relations (between elements). Redrawn from [8].

Information structures relate different entities of information to one another in conceptual graphs (see Fig. 3). A conceptual graph consists of a network of two types of building blocks: elements and relations. The elements capture information about one specific entity, e.g. a part, machine, or a technical drawing. Relations connect elements with another and specifies how their connection is defined; e.g. an element 'product instance' may be connected to an element 'product type', with their relation specifying that the product instance is 'based on' the product type. The relation thus specifies the type and the direction of the connection between two elements.

As the conceptual graph is a graph that captures all information content (and thus grows exponentially throughout a manufacturing process), it is nigh impossible to interpret the contents of the graph as a whole. Hence, multiple levels of structuring are required. Firstly, the information structure may consist of several ‘domains’ of information. The domains refer to a distinct area of interest within the information structure; the domains may be related, but their information contents are mutually independent.

Secondly, for each domain within the information structure, multiple viewpoints on the information contents can be defined. Naturally, different interpretations of the same information contents can co-exist within a manufacturing context and two different stakeholders might use the same information contents for different purposes. For instance, a manufacturing engineer may use a set of processing times for a certain component to evaluate and improve the corresponding processing operation for that specific component, while a process planner uses the same set of processing times to re-evaluate the feasibility of the existing production planning. These respective analyses require different views on the same set of information, which must be accommodated for by the information structure.

Finally, the use of filters is required. Due of the sheer amount of information that is captured even within the specific views, it can still be difficult for users to interpret the information contents. Oftentimes, a user only wants to consider a specific subset of information; for instance, a manufacturing engineer who wants to analyse a set of recorded processing times may want to limit themselves to those parts that were processed in a particular production batch or by a specific machine only.

Besides the structuring and interrelating of information entities, another prerequisite for the utilisation of the information content is the interpretation thereof. As Lutters states, “(...) no matter how comprehensible the information is arranged, if there is no distinct relation between the information content and its denotation, value and use, a useless system results.” [8]. It is thus necessary to use a structure that provides the semantic interpretations of information entities and relations. An ontology provides a conceptual model that captures the meaning and relationships of the elements within an information structure [6]. It specifies the vocabulary, definitions, and constraints that govern the information content. By formalising the content of an information structure using an ontology, it becomes possible to analyse the information in a systematic and consistent manner.

Similar to the information content itself, the ontology of the information structure can be captured in a conceptual graph, displaying the different types of information that occur in the graph of the information content. It depicts the relations between the different typifications of the information content. For instance, the information structure of a product may contain an assembly which consists of several components, but which is in itself also a component for a higher-level assembly. These are two typifications for the same information entity that coexist; whichever typification is relevant depends on the adopted view on the information content. Meanwhile, the accompanying ontology graph represents these relations as one element ‘assembly’, one element ‘component’ and a relation ‘consist of’ that links these two in the correct direction [8]. The types of elements and relations only have to be defined once in the ontology in order to be usable in the information structure.

In conclusion, while the information structure captures the information content and the relations between the information entities, the ontology provides the context that is required in order to be able to formalise and interpret the information content and relations. Both are required for the traceability and interoperability of the information entities across different IT systems and the utilisation of the information contents by stakeholders in the manufacturing environment [6, 12].

2.2. *Integration of supporting IT systems*

In order to manage the vast amount of information that is created throughout the life-cycle of their products, many businesses in the high-tech manufacturing industry use a plethora of IT systems to store, process, retrieve, and analyse their data. This section describes a number of the most important IT systems in a manufacturing context, discussing their functionalities and integration opportunities.

2.2.1. *Definitions of IT systems: ERP, PLM, MES*

Enterprise Resource and Planning (ERP) systems provides a unified enterprise view of the business which encompasses all functions and departments, and a central database in which actions and decisions concerning finance, sales, marketing, purchasing and human resources are stored and traced [4]. ERP integrates and controls various business processes, including the tracking of business resources (e.g. raw materials and production capacity) and the status of purchase orders [1]. Due to its business-driven nature, the concept of ERP has been adopted across all different industries. In many businesses, ERP has emerged as the enterprise management system and as the heart of the information architecture of many enterprises [1, 2].

Product Life-cycle Management (PLM) encompasses the process of managing products and their corresponding documentation throughout its entire life-cycle. Adopting a product-centric approach, PLM systems allow users to manage all information content that is related to the definition of a product [2], including technical design documentation, manufacturing process documentation, and version history. Hence, the primary contents of a PLM system consist of the relevant and required information on product types and, possibly (depending on the industry and manufacturing strategy), corresponding product instances [9]. Since the scope of PLM encompasses the entire life-cycle of a product, one of PLM’s main functionalities is version and change management across the product type’s life-cycle [5].

Manufacturing Execution Systems (MES) are software tools that enable information exchange between the organisational level of a company and the control systems for the shop floor, which usually consist in several different, highly customised software applications [1, 5]. It bridges the gap between design and production and thus serves as a bidirectional communication system between the shop floor, its machines and employees and the other IT systems that contain the information content that is necessary to execute production. In one direction of information flow (from design to production), MES provides the required information to the shop floor employees and machines. In the opposite direction (from production to design), MES records and processes relevant measurement data from the shop floor, which can be used to evaluate the quality of the product and corresponding production processes [5].

2.2.2. *Interaction and integration of ERP, PLM and MES*

When it comes to the division of information storage and functionalities over PLM, ERP, and MES, there is not one 'certain' right approach. Many different task divisions can be designed and adopted, since there is some overlap in the functionalities of the three systems. As a consequence, the same information content can be stored in more than one of the information systems [2]. In practice, businesses have often not made clear decisions about the division of functionalities and information contents across their IT systems; as a result, information is scattered across PLM, ERP, and MES, as well as other systems [2], which is harmful for the traceability and transparency of the information contents; information ends up scattered across the IT systems, without a clear definition of which system is managing what information. This problem demonstrates that it is required to define an enterprise IT architecture in terms of functionalities across the various systems in order to support information traceability and transparency. This section reviews past integration efforts of ERP, PLM and MES.

Ben Kheder et al. stipulate that the manufacturing activity is the intersection of four different life-cycles: the product type, product instance, manufacturing system, and purchase order life-cycles [2]. For each of these life-cycles, a classification of its life-cycle activities is made according to two criteria:

1. Activity type: are the output and the duration of this activity within the life-cycle certain or uncertain?
2. Activity output: does the output of this activity take shape as (virtual) data or a physical effect?

As a result, each activity of each life-cycle can be placed in one out of four categories: data-certain, data-uncertain, physical effect-certain and physical effect-uncertain. Examples include:

- The reception and preparation of a production order: data-certain.
- The mutation (design or process revision) of a product type: data-uncertain.
- The manufacturing of a product instance and delivery of a purchase order: physical effect-certain.
- The maintenance of a manufacturing instance: physical effect-uncertain.

Ben Kheder et al. then propose to manage the data-certain activities in ERP, data-uncertain activities in PLM, physical effect-certain activities in MES and physical effect-uncertain activities in other dedicated applications. By identifying links between the activities of the various life-cycles, the data exchanges between ERP, MES, and PLM become apparent. D'Antonio et al. [4] build further onto this concept and make some additions to it. This results in the general architecture as presented in Fig. 4.

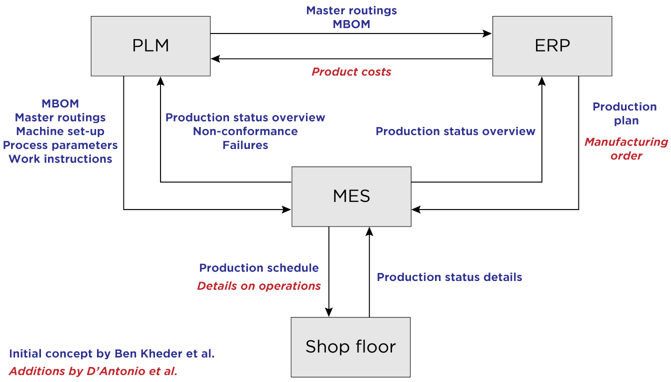


Fig. 4: The integration architecture of ERP, PLM and MES as proposed by Ben Kheder et al. and D'Antonio et al., adapted from [2, 4].

In a different approach, Avvaru et al. [1] suggest using one structured central database as an intermediary between ERP, PLM and MES. This central database is referred to as a 'Knowledge Base System', out of which the IT systems can transfer and withdraw the necessary information. The entity relationship diagram in Fig. 5 displays the information flow between ERP, PLM and MES. All of this information flow would take place via the central database. It is clear that Avvaru et al. shape their integration according to the traditional definitions of ERP, PLM and MES as described in Section 2.2.1 and include explicit data links; however, in contrast to the approach by Ben Kheder et al. and D'Antonio et al., product specifications and product instance data are primarily stored in ERP. Avvaru et al. argue that the ERP system is the 'interface' to the customer, and thus that information received from and meant for the customer

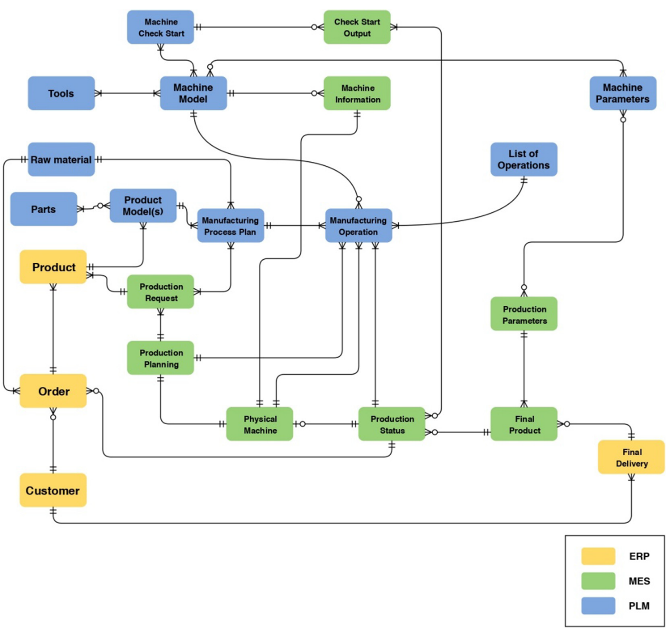


Fig. 5: The integration architecture of ERP, PLM and MES as proposed by Avvaru et al. [1].

must be stored and managed in this system. Ben Kheder et al. and D'Antonio et al. appear to argue that PLM is the more suitable system to store and manage this data, since product specifications and product instance data is primarily relevant for the manufacturing enterprise.

Integration efforts by Nigischer et al. focus mostly on the integration of MES and PDM[1]. The developed architecture aims to establish a closed-loop information flow between the involved information systems and to integrate additional data sources, such as production data of production facilities and sensor data of products in their use phase [9]. The closed-loop information flow is divided into forward and backward integration. The forward direction encompasses the propagation of relevant engineering information from PDM to downstream systems, i.e. MES and (indirectly, via MES) the shop floor applications. Based on measurement data, MES creates relevant performance indicators and forwards relevant information concerning the overall production status to ERP. The backward integration aims to use the collected process data to identify potential improvements for subsequent production runs and design revisions. MES, the shop floor applications and the product instance(s) respectively send planning data, measurement data and use phase data to a data analytics platform, which feeds back product entity information into PDM.

3. Proposed Solution

The literature review has demonstrated that three components are required for the improvement of information traceability in production. Firstly, information entities need to be linked to one another - a manufactured product instance must have a unique identifier to which process inputs and process outputs are explicitly connected. Secondly, the information contents need to be displayed in a structured manner. In order for stakeholders to analyse the gathered information, an information structure must be adopted which makes use of domains, views and filters. Finally, an IT architecture must be used which allows for the integration of information contents as described previously. In order to achieve this, there must be a clear description of each software system's responsibilities and an overview of the required information exchanges.

This section describes the envisaged approach to fulfilling these requirements. The interlinking of information entities is covered in Section 3.1. Section 3.2 describes the information structure and the use of domains, views and filters. Finally, the overarching IT architecture to accommodate for the integration of information contents is described in Section 3.3.

3.1. Interlinked data identifiers

The entity relationship diagram in Fig. 6 shows the core concept of the interlinked data identifier landscape. The landscape can be interpreted as a conceptual ontology for information content generated before, during, and after production. It is important to note that this ontology may not be fully complete; the ontology instead aims to point out the most obvious types of information entities and relationships in a manufacturing context. Whenever a 'real' information structure is filled with information contents, it may become apparent that the proposal in Fig. 6 contains insufficient information to capture and describe all relevant types of information entities and relations. The background of the data identifier landscape provide more insight into which information contents are created in which IT system; more information on this can be found in Section 3.3.

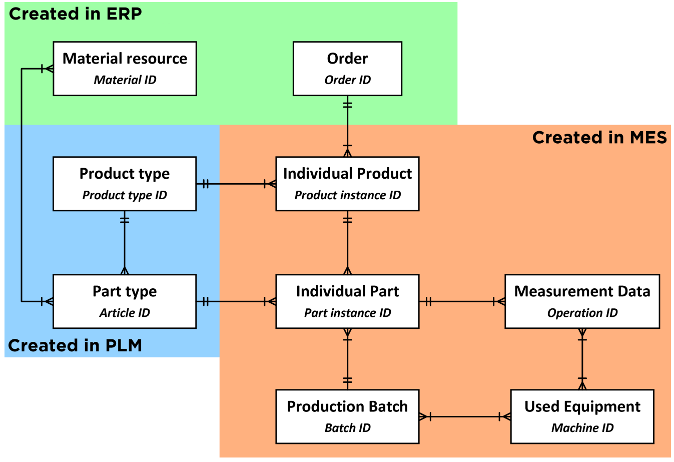


Fig. 6: The interlinked data identifier landscape; relations are indicated in crow's foot notation.

The ontology is limited to the scope of a production order and created using crow's foot notation. A customer places an order for a number of product instances; these product instances each have their own (and only one) corresponding product type. Both the product type and product instance consist of multiple parts; the part types and part instances, in turn, are only part of one product type or product instance. Following this approach, it becomes clear that there are a number of one-to-many and many-to-many relations to be identified. This is illustrated in an example in Fig. 7, in which three part instances are based on the same part type.

What is also important to note is the fact that some information entities are linked to the same identifier. For instance, the information entity 'part type' refers to all of the information content related to the part type - product data (drawings, CAD-files, specifications, etc.) *and* corresponding process data (machining programs, traceability protocol, work instructions, part routing, etc.). This approach closely follows the traceability principles as described in Section 2.1.1 and Fig. 2 in particular.

3.2. Resulting information structure

The example in Fig. 7 demonstrates that the interlinked information content clutters extremely easily. When more information entities are added (in the form of more process inputs and outputs, as well as more part types, part instances, opera-

[1] PDM is an abbreviation for 'Product Data Management' and can be regarded as a subset of PLM that is primarily concerned with managing product data and documents.

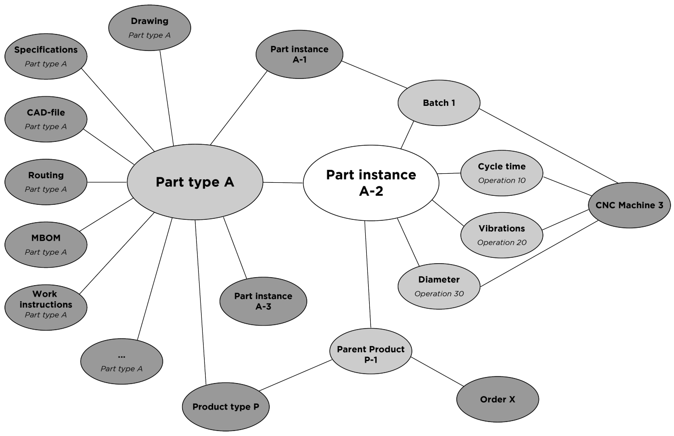


Fig. 7: An example of interlinked product and process information. The chosen focal point is *Part instance A-2*; the colour gradient indicates the proximity of other elements to this part instance.

tions, production batches, etc.), the graph as depicted in Fig. 7 becomes very hard to interpret without any additional structuring efforts.

Fig. 8 displays the information structure. The entirety of the information structure would capture any information contents that describe an incoming customer order. However, since the scope of the research topic is limited to enhancing information traceability within the production phase, only one domain (the production of the order) is considered for the remainder of this section.

Four different aspect views have been chosen for the information structure, three of which (i.e. the views ‘Product’, ‘Process’ and ‘Machine’) are traditionally important focal points for improvements in production. The fourth view (‘Traceability’) is meant to evaluate how well the information has been traced and what adaptations may need to be made. The following paragraphs will elaborate further on these aspect views, and their corresponding analyses and filters.

3.2.1. Product view

The product view adopts the produced product instances as a focal point. For a product instance, it is possible to investigate the Configuration As-Built and thus to investigate which part instances make up a product instance. Additionally, information regarding the production history and quality can be viewed for a particular product instance or a part thereof. The most important filter that can be applied to all of these analyses is the product instance filter, allowing the user to select and view a specific product instance. The product view can be interesting for a variety of stakeholders, such as a manufacturing engineer, a design engineer, a quality engineer, or a client.

3.2.2. Process view

The process view focuses on the information content surrounding the production process and specific underlying operations. This view exceeds the scope of the product view and addresses the production process of the entire production order. For a specific operation, it is possible to retrieve a summary of the recorded measurements (e.g. cycle times, spindle speeds, and product measurements). Filters that could be applied for this type of analysis are a batch filter (allowing the user which production batches are included in the summary) and a machine filter (in case the specific operation was conducted by multiple machines in parallel). Additionally, the process view allows the user to investigate the overall quality of the production order by means of accessing relevant KPIs (e.g. yield rates, first-time-right rates). The process view is of great relevance for manufacturing engineers seeking to optimise and stabilise production processes and production planners aiming to improve the production planning of future production runs.

3.2.3. Machine view

The machine view allows users to retrieve information about the performance of a specific machine or production asset. It is possible to view key performance indicators (KPIs) for individual machines (e.g. *OEE*, yield rate, C_{pk}), as well as to compare multiple machines that executed the same operation(s) through statistical analyses (e.g. ANOVA). Furthermore, problematic production assets can be identified through a problem diagnostics analysis, pinpointing the cause of substandard machine performance. Similar to the process view, the machine view can aid manufacturing engineers in stabilising operations. Additionally, it can be of relevance for machine operators and maintenance engineers when a particular machine shows poor performance or exhibits strange behaviour.

3.2.4. Traceability view

The traceability view has a somewhat different scope than the previous three views. Whereas the product, process and machine views provide opportunities to analyse and improve the quality of the physical production process and resulting product instances, the traceability view is used to judge the quality of the gathered data and to review and adapt the used traceability protocols. Besides the ability to request an overview of the current traceability protocols for a certain product type, the traceability view can provide suggestions for changing traceability protocols based on the gathered data. This specific analysis can suggest a stricter protocol for part types that have a lower overall quality score, or a less strict protocol for a part type that is consistently produced well within all specified tolerances. Additionally, the traceability view provides an overview of the information content that should have been recorded during production, but is ultimately missing. The traceability view is primarily relevant for quality engineers and product data managers.

3.3. Supporting IT architecture

A first indication of the various IT systems’ purposes and control domains is already given in Fig. 6. The definitions of the IT systems as described in Section 2.2.1 were closely followed in the design of the general IT architecture. The following paragraphs describes the functionalities and interactions of these systems. Furthermore, the general task division as pro-

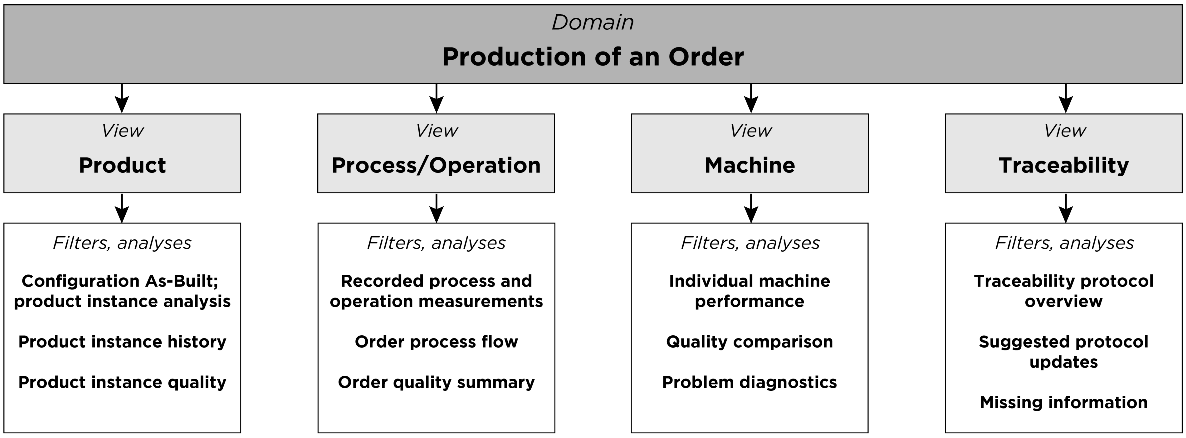


Fig. 8: The envisioned information structure with its defined domain, views, and filters.

posed by [2] (based on the type and output of the tasks) was followed, notwithstanding a couple of notable exceptions.

3.3.1. Enterprise Resource Planning (ERP)

The primary domain of the ERP system in a manufacturing context is the management of incoming orders from customers. Receiving, accepting and processing orders are the core processes of ERP. With this, a couple of affiliated tasks emerge: inventory management and purchase orders for supplier parts and raw materials are managed in ERP, as well as cost calculations (preliminary and subsequent costing). This is reflected in Fig. 6, in which information about orders and material resources are depicted as part of the ERP domain.

3.3.2. Product life-cycle Management (PLM)

The PLM system manages all information contents that concern product and part type data (i.e. master data). As stated before in Section 3.1, this encompasses both product and process data. In the type of manufacturing environment that is under consideration, product and part master data is constantly evolving throughout its life-cycle. This implies that proper version and change management is essential, making PLM the most suitable environment for this type of information management. Furthermore, PLM will contain the information contents related to finished product instances. This is convenient for the analysis of the configuration as-built of the product instances and the comparison of product types to their physical instantiations. Storing the information contents related to finished product instances is feasible due to the adopted manufacturing strategy; only very few unique products are manufactured based on the same set of inputs, since the product and process definitions evolve quickly. Thus, the low production quantities allow for product instance information to be stored and managed in PLM without reaching the limits of storage capacity and losing the overview.

3.3.3. Manufacturing Execution System (MES)

The manufacturing execution system (MES) is designed to do exactly what its name describes: enabling the execution of production orders. Combining product master data and order information, the task of MES is to plan, execute and monitor the orders. A major difference with the proposal of [2, 4] is the incorporation of the production planning functionality in MES. Traditionally often executed in ERP, production scheduling will be shifted to MES. In the stipulated IT architecture, ERP does not contain the necessary information to complete the scheduling activity (i.e. the MBOM and routings). Furthermore, in order to ensure the possibility of real-time adjustments in the planning based on shop floor data, moving the scheduling activity to MES is the better option. More details on how the interaction between ERP and MES takes place with regards to scheduling and monitoring can be found in Section 3.3.5. Furthermore, MES is responsible for creating new part and product instances and assigning a unique identifier to these instances. The measurement data, used equipment and processed production batches will be coupled to the part instances in MES. Finally, upon completion of production, MES ensures that a final configuration as-built is created for the product instances.

3.3.4. Additional IT systems

Besides the triangle of ERP, PLM, and MES, some additional IT systems are required. The two general systems that complete the core IT architecture surrounding production activities are data warehousing and data analytics systems.

A data warehousing system is added to deal with the copious amounts of information generated over time. As can be read in the previous sections, MES collects measurement data which is coupled to product instances. The configuration as-built of the product instances, containing all of the relevant process input and output data, is subsequently stored in PLM. While PLM is a good platform for viewing this information, it is not meant to store extremely large amounts of information. Besides this, it is not necessary to actively maintain product instance information in PLM for products that have been manufactured years ago and are past their end-of-life. While it can still be necessary to access this information, it does not need to be as readily available as recently manufactured product instances. Therefore, a data warehouse is used to store product instance information of

Table 1: A modular N^2 diagram of the proposed IT architecture, showing the information flow between the different IT systems (clockwise).

ERP		Manufacturing order Material resource availability		
Required material resources	**PLM**	Product type information	Old product instance information	
Production status update Material resource update	Product instance information (Configuration As-Built)	**MES**	Raw measurement data	
			Data warehouse	Raw measurement data
	Processed measurement data (enriching Config. As-Built)			**Data analytics**

'old' products. Note that the definition of an 'old' product is fully determined by the life-cycle duration of the product instance. Furthermore, the data warehouse contains the raw measurement data that was recorded on the shop floor and gathered by MES. Though users will usually not need to look at the raw measurement data due to the availability of structured data and various data analyses (see Section 3.2), it must be possible to retrieve the source data on which the analyses are based in case anomalies occur that cannot be explained by the outcomes of the analyses.

In addition to data warehousing, a data analytics application must be used. The raw measurement data that is collected by MES must be processed in order to yield meaningful insights. The types of analyses as described in Section 3.2 must be defined separately from the actual information content in order to ensure repeatability. The raw measurement data is processed by the data analytics application and used to calculate order- and product-specific quality parameters, which can subsequently be coupled to the product instances in PLM and used to update the production status and material resource overview in ERP.

3.3.5. Integration and interaction of IT systems

A number of interactions between the described IT systems have already been touched upon in the previous paragraphs. The modular N^2 diagram in Table 1 depicts a complete overview of the general information transfer streams between all components in the IT architecture. In order to be able to execute the production of an order, MES needs to receive the general order information (client, due date, order quantity, etc.) from ERP and the product type information (drawings, machining programs, MBOMs, work instructions, routings, etc.) from PLM. During production, MES collects measurement data from the shop floor and forwards this data to the data warehouse. The data warehouse stores the unprocessed measurement data and transfers the data to the data analytics application. Here, the data is converted into meaningful KPIs that are coupled to the correct data identifiers. MES also keeps ERP up-to-date on the general production status and the material resource stock levels. Upon completion of production, MES forwards the product instance information (containing the configuration as-built) to PLM, which is enriched by the KPIs from the data analytics platform. Note that the scope of this preliminary architecture is limited to the information streams related to manufacturing activities.

4. Case Study at NTS Norma Hengelo

The goal of the case study is to verify and validate the general solution principles as formulated in Section 3 and the feasibility of their implementation. By means of applying these principles in a realistic scenario, the aim is to investigate the feasibility of the proposed information structure and the supporting IT architecture. The case study is conducted at NTS Norma, a build-to-print contract manufacturing company based in Hengelo, the Netherlands. Following a short company description in Section 4.1, Section 4.2 describes why information traceability is an urgent issue within NTS Norma and how the case study aims to improve this. Section 4.3 elaborates on the results of the case study, which is evaluated in Section 4.4.

4.1. Company description

NTS Norma Hengelo is a HMLV, high-precision, high-complexity contract manufacturer and first-tier supplier to various OEMs in the semiconductor, aerospace and defence industries. NTS Norma classifies as a medium-sized enterprise, employing around 400 FTE.

In most projects, NTS Norma adopts a build-to-print approach. A complete package of technical product specifications is delivered by the client; after issuing the necessary adjustments and verifying the product specifications, Norma designs, tests and executes the corresponding production process. The development teams of Norma work according to a stage-gate model which closely resemble the steps in the design process as formulated by Pahl et al. [11]. The stages can be sub-divided into three different project life-cycle phases: sales, *new product introduction* (NPI), and volume production. The sales phase starts with a customer order request and delivers a business case and corresponding quotation. Upon acceptance by the customer, the project enters the NPI phase. During the NPI phase, the technical product specifications are verified and the process design is created, tested and finalised. The NPI phase ends with the production of a prototype and (when the prototyping stage was deemed successful) the production of a pilot series. When the processes are sufficiently under control, a release-for-volume is issued and the project enters the volume production phase. In this phase, the production capacity is upscaled and the products are manufactured according to the verified processes and specifications.

Since the goal of the NPI phase is to mature the product and production process to a state in which the product can be reliably manufactured to the high quality standards, the various stages in the NPI phase are followed in an iterative manner. Throughout the NPI phase, the information content regarding the design of the product and production process is constantly evolving. For instance, if a pilot series is run and it appears that a certain step in the production process does not meet the set quality standards, the project team may alter this operation and evaluate the new process in a subsequent production run. Similarly, if there is an opportunity to slightly adjust a product feature in order to make it more easily manufacturable without compromising its functionality, the design of the product may be adjusted in close cooperation with the client. Especially during the prototype and pilot series production stages, the new versions of product and process specifications follow each other in rapid succession, with only few products being manufactured during each iteration. Since the products and associated production processes are highly complex and thus costly, it is important that the information contents are managed effectively and efficiently in order to progress to a state of confirmed product and process maturity as quickly as possible.

4.2. Initial problem statement and scope of the case study

As was already stated in the introduction of this paper, information traceability and management are an influential factor in the success of information-driven development and engineering activities. Due to the rapidly evolving, highly complex products and process that characterise NTS Norma's manufacturing environment, achieving adequate traceability of information is especially important to enable efficient operations within development projects - particularly those in the NPI phase. However, a clear approach and support structure to achieve this are currently lacking. Through interviews with various managers, team leaders and other stakeholders at NTS Norma, it was discovered that information traceability is very limited. Three main causes of poor information traceability were uncovered: the lack of a strategic IT architecture, paper-based information handling in production, and human errors and negligence.

Firstly, the current IT architecture lacks a clear strategy. A multitude of IT systems is used to manage the enormous amounts of information content; however, the responsibilities of the IT systems and the overall information flow between systems have not been explicitly defined. Instead, the current IT architecture at NTS Norma is a circumstantial result of how the systems are being used. Consequentially, none of the employees at NTS Norma have a precise overview of this architecture and clear guidelines regarding information handling across the IT systems are missing. Boundaries of several IT systems are vague and mostly undefined; as a consequence, there are major inter- and intradepartmental differences regarding the standard procedures of handling and storing information.

Efforts to map out the current IT architecture by means of stakeholder interviews are visualised in Appendix A, *Current IT Architecture of NTS Norma*. What can clearly be seen in this landscape of IT systems is the sheer abundance of applications. Many duplicate systems (e.g. four different CAD programs) are used for the same purpose, since new systems were added over time while old systems were not always completely phased out. In the meantime, NTS Norma relies heavily on the use of spreadsheets to execute processes across all departments and development phases. The user-friendliness and customisability of spreadsheets can be an advantage, but also pose a threat: every department makes use of their own, often locally stored, set of spreadsheets. As a consequence, there is little standardisation and several departments might be relying on different procedures and data sets to perform the same activities - in fact, it can hardly be known if this is the case at all, since there is no central overview of used spreadsheets and their contents. This severely hampers the integration of the information contents and activities of the various departments.

The second cause lies in the fact that the handling of information on the shop floor is predominantly paper-based. Except for information contents which must be presented in a digital format (e.g. NC machining code), all process inputs for the manufacturing operations are presented to the shop floor operator as physical copies. Printed product routings, set-up drawings, and work instructions are bundled into a 'job traveller', i.e. a folder bound to the production of a specific (part of a) production order. Operators rely on and often annotate the documents in this folder. As a consequence, updates to the contents of the job traveller cannot be synchronised immediately and job travellers may thus contain outdated and incorrect information. If a change is made to one of the documents in the job traveller, every individual job traveller must be found and its contents must be corrected - which is a rather time-consuming and error-prone process. Furthermore, job travellers and their corresponding products get lost on a daily basis. Since no digital system accurately tracks the progress of the products, it can be difficult to retrieve a lost job traveller.

Finally, human errors and negligence also are a cause of poor information traceability. This has much to do with the currently established ways of working on the shop floors. Because operators mostly handle paper-based information, registering unique part identifiers is a manual and cumbersome process. Serial numbers and article numbers often have to be copied and written by hand, which is obviously an error-prone procedure. Additionally, some of the necessary data-recording procedures (such as recording production times) are simply very time-consuming and repetitive and are often executed poorly out of convenience.

The problems listed here had already been recognised internally to a certain extent. NTS Norma had, therefore, initially expressed the need to 'redesign the IT architecture'. However, redesigning the IT architecture only makes sense when a specific target is considered - after all, the IT architecture is a means to support and achieve certain commercial goals, but not a goal in and of itself. Through extensive stakeholder interviews, listing current problems and lacking functionalities, and grouping these functionalities into potential case studies, the goal of the case study was specified as 'improving the traceability of information surrounding the production activities at NTS Norma'. A more elaborate description of the scoping process that preceded the chosen case study can be found in Appendix B, *Scoping the Case Study*. The case study is focused on the NPI stages of prototype and pilot production. These two particular stages are crucial in maturing a product and production system for volume production and require the effective use of feedback through measured data in order to be executed in a timely manner. The case study combines multiple types of information traceability, focusing mostly on passive, both backward and forward, and internal traceability.

The case study has three main deliverables that contribute towards this goal. Firstly, a method for applying the principles of information traceability as described in Section 2.1 at NTS Norma must be developed. Secondly, the IT architecture that is designed to accommodate for improved information traceability (as described in Section 3.3) must be applied to the situation at NTS Norma. Finally, the potential of adequate information traceability must be demonstrated by visualising the information structure from Fig. 8 in a mock-up traceability tool.

4.3. *Developing a framework for improved information traceability in production*

The three deliverables of the case study are discussed in this section. Section 4.3.1 describes the developed approach to apply the traceability principles in a practical context at NTS Norma. Next, Section 4.3.2 demonstrates the flow of information contents before, during and after production activities between the different IT systems, following the principles as described in Section 3.3. Finally, Section 4.3.3 describes the development process and outcomes of the 'traceability tool', i.e. the visual representation of the information structure as presented in Section 3.2.

4.3.1. *PDCA approach towards information traceability*

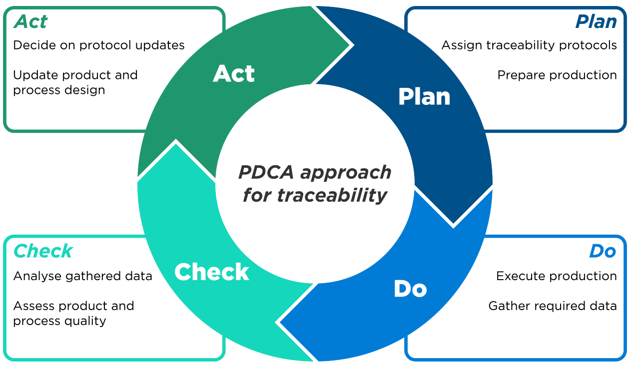


Fig. 9: The suggested PDCA approach for improving information traceability during production.

Tracing information is a continuous activity during a product's life-cycle. Hence, an approach to implementing information traceability must be adopted that takes into account the different product development activities and phases at NTS Norma. Currently, the stages of NTS Norma's development process are being followed in an iterative manner; for instance, a pilot production series is executed, evaluated and adjusted multiple times before it is released for volume production. To accommodate for this method of continuous improvement, a PDCA approach is employed for information traceability. PDCA (*Plan, Do, Check, Act*) is a four-step iterative method used for the continuous improvement of processes, products, and services [17]. Fig. 9 shows the PDCA cycle applied to the topic of information traceability within production. The four steps are applied as follows:

1. *Plan*: before a (first) trial production run starts, determine how and which information must be traced for the produced parts. This is done by assigning 'traceability protocols' to the part types.
2. *Do*: during the production run, record the data that is prescribed by the traceability protocols.
3. *Check*: after the production run, analyse the gathered data and assess the product and process quality.
4. *Act*: in preparation of a subsequent production run, update the product and process design according to the analysis outcomes. Additionally, decide which part types need to receive a traceability protocol update.

These four steps will be discussed in more detail in the upcoming paragraphs.

Plan. The *Plan* phase involves assigning traceability protocols, the importance of which was already discussed in Section 2.1.1. In an effort to distinguish different levels of information traceability, five different protocols have been defined. An overview of these protocols can be seen in Table 2. The protocols use two types of classifications to distinguish different part types: the origin of the part type (manufactured by NTS Norma or purchased from a supplier), and the criticality of the part, the latter of which requires a more elaborate definition. A part type is classified as critical-to-quality (CTQ) when at least one of the following conditions applies:

- The client defines the part as 'critical' and/or requires individual identification of the part type;
- The part type is newly introduced at NTS Norma and is being manufactured for the first time;
- The part type makes use of a 'critical-to-quality' production process, i.e. a process which has shown large variations over time and is not (yet) under sufficient control (low yield rate and/or first-time-right rate), has to yield results within a very small dimensional tolerance zone, or has been newly introduced.

Any part type that is labeled as CTQ requires individual identification of its manufactured part instance, through the use of a part instance (or 'serial') number. Additionally, measurements recorded during manufacturing activities must be identified on an individual part instance level. This is not necessary for non-CTQ part types, for which the production batch is a sufficiently detailed level of identification and measurement recordings. An exception is made for part types that are not labeled as CTQ

Table 2: An overview of the different traceability protocols.

Protocol	**Description**	**Identification level**	**Measurement level**
Make-regular	Non-critical manufactured parts/assemblies.	Batch	Batch
Make-CTQ	Critical-to-quality manufactured parts/assemblies.	Individual	Individual
Make-CTQ-underlying	Assembly structures that are non-critical themselves, but have underlying critical parts. Underlying assembly structure must be captured.	Individual	Batch
Buy-standard	Off-the-shelf, standardised purchased parts.	Order	Order
Buy-custom	Purchased custom-made parts.	Individual	Individual

themselves, but have underlying components which are labeled as CTQ. In order to keep an accurate overview of the configuration as-built of products with CTQ components, it is essential to identify the parent assemblies of these components on an individual level as well. However, it is not strictly necessary to also match recorded measurements to part types that receive the label 'Make-CTQ-underlying'. For purchased parts, a distinction can be made between 'off-the-shelf' standardised parts (e.g. nuts, bolts, screws) and customised parts which have been manufactured by the supplier.

A decision tree for assigning these traceability protocols can be seen in Fig. 10. The decision tree is meant to be used in a 'bottom-up' approach: starting with the lowest-level part types of a product MBOM (individual components), and ending with the highest-level part types (assemblies), each part type receives a traceability protocol. This bottom-up procedure is essential for identifying assemblies with underlying CTQ components.

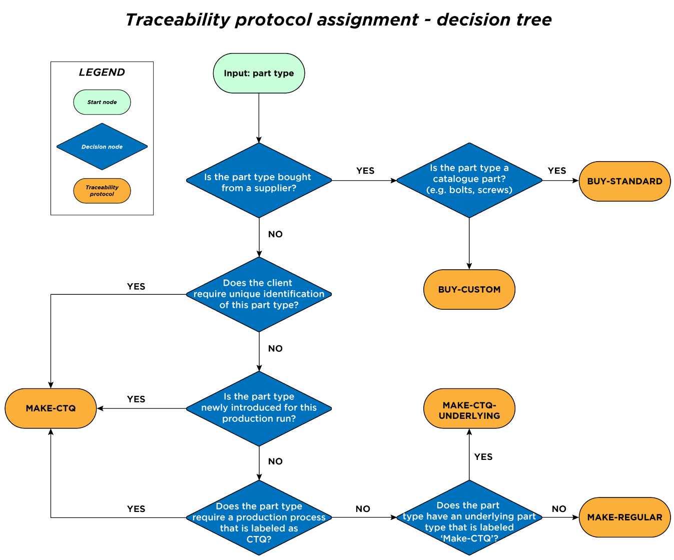


Fig. 10: The decision tree for assigning traceability protocols to part types.

The protocols each also dictate which types of information contents (both process inputs and outputs) are necessary to trace. For instance, part types with a 'Make-CTQ' label may require certain measurements to be recorded for each production step on an individual level. On the contrary, for part types that have been labeled as 'Buy-standard', it is merely necessary to record which supplier or vendor the parts came from and their associated purchase order number. The result of this traceability protocol assignment process is an empty 'traceability structure': a manufacturing bill of materials (MBOM) with an overview, per part type, of the information to be recorded. This traceability structure is filled with all required process inputs (all master data corresponding to the product type) before the manufacturing activities start.

Do. After the traceability structure has been filled with the required information inputs, the *Do* phase can start. This phase encompasses the manufacturing activities and aims to fill the traceability structure with the required information outputs while the product instances are being manufactured, such as assembly structures and measurement data. Here, it is important to note the distinction between the information inputs and the information outputs in the traceability structure - while information inputs are generic and associated with a product (or part) type, the information outputs are unique and associated with a specific instance. Following one-to-many relation between information contents about a product (or part) type and its physical instantiation(s) as shown in Fig. 6, the traceability structure will establish links between all manufactured part instances and their respective types. The traceability structure will thus record all process outputs on the level of the production order. The configuration as-built of a product instance is derived from the traceability structure only upon completion and approval of the product instance. This ensures that no faulty parts or assemblies will be registered into a configuration as-built. An example of a traceability structure, visually represented as a table, can be seen in Table 3.

Check. The *Check* phase takes place predominantly after completion of the manufacturing activities. In this review-oriented phase, the gathered information is analysed in order to associate the quality of the manufactured products, their constituents, and the associated production processes. In order to do so, the viewpoints and analyses as defined in Fig. 8 are used. The information contents of the traceability structure serve as input for these analyses. The analyses are used to derive various order-specific, product-specific, part-specific, and process-specific KPIs, such as yield rates, overall equipment effectiveness, and variations in measured parameters and processing times.

Act. Finally, the *Act* phase aims to implement the feedback derived from the analyses from the *Check* phase. Product and

Table 3: A heavily simplified example of a (filled) traceability structure, represented as a table. The left part of the table represents the (generic) process inputs, while the right part of the table contains the (unique) process outputs. It showcases that the traceability structure captures information about all manufactured part instances in a certain production order - including those that have been scrapped and were not included in any final product configuration.

Level	Part type ID	Traceability protocol	Process inputs	Part instance ID	Batch ID	Process outputs	Parent
1	Parttype-1	Make-CTQ-underlying	Inputs-parttype-1	Instance-1-1	Batch-1-1	Outputs-batch-1-1	N/A
				Instance-1-2	Batch-1-1	Outputs-batch-1-1	N/A
1.1	Parttype-1.1	Make-CTQ	Inputs-parttype-1.1	Instance-1.1-1	Batch-1.1-1	Outputs-instance-1.1-1	Parttype-1-1
				Instance-1.1-2	Batch-1.1-1	Outputs-instance-1.1-2	*Scrapped*
				Instance-1.1-3	Batch-1.1-2	Outputs-instance-1.1-3	Parttype-1-2
1.2	Parttype-1.2	Make-CTQ	Inputs-parttype-1.2	Instance-1.2-1	Batch-1.2-1	Outputs-instance-1.2-1	Parttype-1-1
				Instance-1.2-2	Batch-1.2-1	Outputs-instance-1.2-2	Parttype-1-2
1.3	Parttype-1.3	Make-CTQ	Inputs-parttype-1.3	Instance-1.3-1	Batch-1.3-1	Outputs-instance-1.3-1	Parttype-1-1
				Instance-1.3-2	Batch-1.3-1	Outputs-instance-1.3-2	Parttype-1-2

process designs are adjusted where needed. Simultaneously, the traceability protocols are re-evaluated. The outcomes of the analyses may suggest that a stricter traceability protocol may be required, or that the current traceability protocol is unnecessarily strict. As part of the continuous improvement principle, the completion of this phase will result in a new cycle of production preparation, execution and evaluation.

4.3.2. Proposed IT architecture

The PDCA procedure towards information traceability must be accommodated by a clearly defined and well-integrated IT architecture. Creating the traceability structure and linking the right information contents is meant to be a fully digital process. Fig. 11 displays the information flow between and activities within IT systems during the production activities. This architecture is fully based on the architecture and underlying rationale as described in Section 3.3, but presented here as a process flow diagram. Due to the defined scope, the chronological starting event of this process flow is the acceptance of a client order; likewise, a finished production order defines the end point. Upon order acceptance, general order information is created in ERP. This information is linked to a unique order identifier. In the meantime, the required product type information is created in PLM. This includes product and process specifications. The specifications are linked to their respective part types by means of the part type identifiers. When all of this information is available, the order will be prepared for production. This entails planning the order and creating the traceability structure of the production order. This is done in MES; all required order information, as well as part type information, is thus carried over from ERP and PLM into MES. When the traceability structure has been created and the required inputs have been added, the *Plan* phase has been completed. Next, the *Do* phase is executed by manufacturing the order. During this phase, the traceability structure is gradually filled with the required process outputs. The contents of the traceability structure as a whole are listed in Fig. 11 as 'Production information'. Upon completion of production, the configuration as-built of the product instances are derived from the traceability structure. This product instance information is stored in PLM. The measured process outputs still need to be stored in the data warehouse and processed and analysed in a data analytics application, before they are definitively added to the product instance information in PLM. Relevant KPIs are also communicated to ERP, to update the status of the production order and the physical resources. The *Check* and *Act* phases are less explicitly defined in this process flow; the product instance information is used to evaluate product and process quality during the *Check* phase, and the *Act* phase involves making updates to the product and part type data (also in PLM).

4.3.3. Traceability tool

The *Check* phase makes use of the 'traceability tool'. The traceability tool is a view-only application that extends over the various IT systems in the architecture as proposed in Fig. 11. Using the defined information views and analyses from Fig. 8, it allows users to analyse the gathered production data of ongoing and finished production orders during the activities of the *Check* and *Act* phases of the PDCA cycle, providing a central platform that allows users to access information contents from the various IT systems in a user-friendly overview. The traceability tool was developed as a means to demonstrate the potential of having an adequate information traceability strategy in place and to get feedback on the chosen approach from various stakeholders within NTS Norma.

Requirements. The general requirements for the traceability tool can be found in Table 4. The only software requirement demands interconnectivity and integration with the other IT applications, since it is meant to display the information contents of other systems in a presentable manner, rather than storing its own set of (duplicate) information. For stakeholder-related requirements, it is important that a distinction is made between different roles of users of the traceability tool. A certain viewpoint or set of analyses may be useful for one stakeholder, but irrelevant to another. The scope and structure of the case study (transaction-based information recording, spanning the prototype and pilot production stages) will apply to the traceability tool as well. Additionally, in order to maintain a single source of truth, the traceability tool will be used only to view the information contents. The requirements of the 'Analysis' category contain the various viewpoints and analyses as proposed in Section 3.2. Finally, a set of requirements regarding the prototyping and testing of the traceability tool was added. The aim is to create a prototype mock-up that portrays the most important functionalities of the traceability tool. This prototype will be

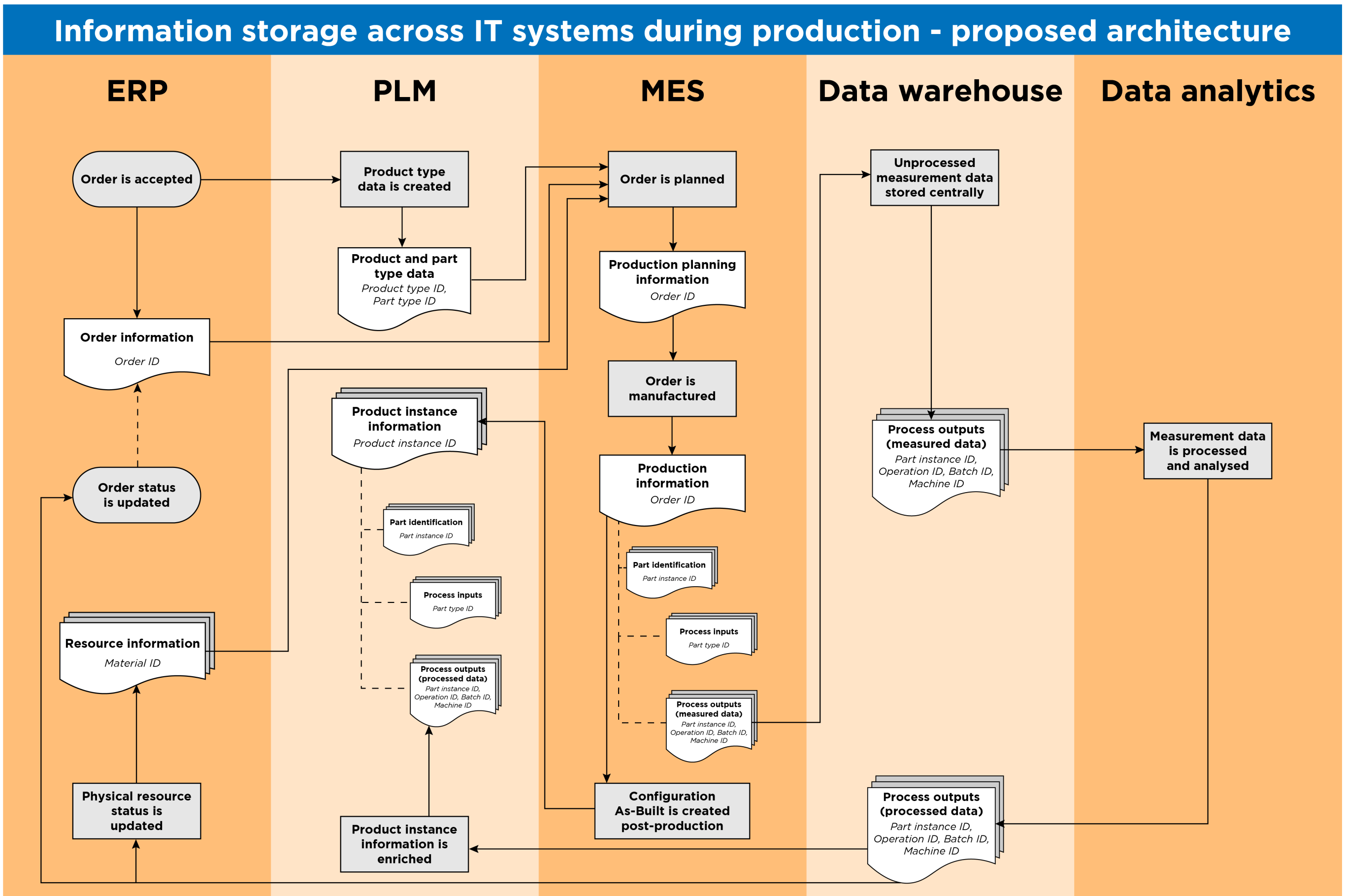


Fig. 11: The proposed IT architecture for NTS Norma pictured as a process flow diagram, based on Fig. 6 and Table 1. The scope of this architecture encompasses the production activity, from the moment of order acceptance until the point of order completion.

demonstrated to the same stakeholders that assisted in scoping the case study, in order to validate the solution principles and to acquire feedback for future development.

Tool design. The traceability tool was designed based on the information structure principles as shown in Fig. 8. The general process flow of the traceability tool is displayed in Fig. 12. From this process flow, it becomes obvious that the tool is intended to be used as an interface between the user and the information contents stored across various IT systems (as displayed in Fig. 11). In summary, the traceability tool allows a user to specify their analysis request through offering a range of views, analysis types and filter options. Subsequently, the tool passes this query to the IT systems that contain the required information contents. From the whole network of information associated with a certain production order, a subset of information is then extracted for analysis. The analysis is executed in the data analytics platform; the outcomes of the analysis are then displayed to the user in the traceability tool. In order to maintain a single source of truth and prevent users from relying on potentially outdated information, it is not possible to export the analysis outcomes directly from the traceability tool to an 'offline' document. This ensures that users will base their insights and decisions on the most up-to-date information at all times, rather than falling back on the old way of working with a plethora of offline spreadsheets. Instead, the exports will be made via the dedicated IT systems in which the information contents are stored and should require some form of authorisation. This way, it is possible to create e.g. customer reports about product instances in PLM.

Furthermore, a slight distinction has been made between the different stakeholders. Because the traceability tool is used on a read-only basis, it was deemed unnecessary to needlessly restrict users in their selection of viewpoints and analyses. On the contrary, having additional access to viewpoints that are not directly of interest for a particular user may even improve collaboration and communication efforts within interdisciplinary project teams. However, in order to guide stakeholders better in their user experience, a selection of potentially interesting analyses can be made for each type of user. These analyses are displayed on the home screen of the traceability tool after a user has logged in with their credentials. The user credentials contain information about a user's role (e.g. operator, manufacturing engineer, or process technologist), which is used for displaying direct links to their most relevant analyses.

The design of the traceability tool is characterised by a need for flexibility. Ideally, it must be possible to use this tool in combination with whatever specific software packages are be-

Table 4: An overview of the general requirements for the traceability tool.

Domain	**Requirement**
Software	The traceability tool should retrieve the information to display from other software platforms, e.g. ERP, PLM, and MES.
Stakeholders	The traceability tool must cater to multiple viewpoints and stakeholder perspectives.
Stakeholders	The perspectives of multiple stakeholders must be defined in terms of required information and useful analyses.
Scope and structure	The traceability structure needs to be ‘transaction-based’, i.e. displaying the transformations of individual parts and (sub)assemblies throughout the production journey.
Scope and structure	The traceability tool must span at least the prototype and pilot production stages (within the NPI phase).
Scope and structure	The traceability tool must be read-only, disallowing the user to alter existing information contents.
Analysis	The traceability tool must be able to display links between product and process specifications, recorded production data and specific traceable resource units (e.g. batch units or individual parts).
Analysis	The traceability tool must be able to link any anomalies or defects that occurred during production to a specific traceable resource unit.
Analysis	The traceability tool must be able to indicate the status and location of a specific traceable resource unit (i.e. show where in the production process the unit currently is).
Analysis	The traceability tool must be able to show individual parts in a Configuration As-Built.
Prototype and testing	The prototype must clarify the working principles of information traceability to the users and provide insight into the potential translation to an IT architecture.
Prototype and testing	The prototype must present a realistic use case of the traceability tool, using a real product in production as an example.
Prototype and testing	The prototype must convey and validate the different (stakeholder) perspectives on the traced information.
Prototype and testing	The prototype must be able to deal with basic user input (e.g. indicating a perspective on the gathered information).

ing used in the corporate IT architecture. Furthermore, if the need arises to add or modify certain viewpoints, analyses, or user roles, it must be possible to do so without encountering too much system rigidity. Hence, the preferable realisation option for the traceability tool would be a web user interface that is to be developed for NTS Norma specifically. This ensures that the tool is not restricted in its communication with other IT systems. Additionally, a web interface would suffice in fulfilling the rather simple and lightweight functionalities of the traceability tool - formulating search queries, retrieving and displaying information content. Finally, it is preferable to display the information from the various IT systems in the most neutral and unbiased manner possible.

Prototype. A simple mock-up prototype was created of the traceability tool, using static dashboards with clickable fields. The main aim of developing this prototype was to adequately portray the intended user experience - more specifically, the navigation and work flow in the traceability tool. The complete mock-up prototype can be found in Appendix C, *Traceability Tool Prototype*. Two sample dashboards of the mock-up are displayed in Fig. 13. Some of the most important elements of the user interface are listed below.

- *Navigation bar*: the bar at the top of the interface allows users to navigate back to the home screen with the general information of the order under review. Furthermore, users can easily navigate between different aspect views via the navigation bar.
- *Location path*: another means of navigating between different viewpoints and analysis.
- *Analysis title and general order information*: used to display the current analysis, as well as the order number, product description and the client.
- *Filter options*: displayed on the left side of an analysis window, the filter options require input of the user and specify the analysis.
- *Output window*: displayed on the right side of an analysis window, the output window shows the outcomes of the analysis in a graphical and/or textual manner.
- *Clickable links and pop-up windows*: identifiers that are displayed in an analysis window (e.g. batch, part type and machine identifiers) contain links to pop-up windows that display specific information about that identifier (e.g. a general description, process inputs and outputs, and other associated identifiers).

4.4. *Evaluation and conclusions*

The evaluation of the case study consists of the verification and validation of the proposed methods and designs. The case study is verified by comparing the solution outcomes to the proposed solution principles, guidelines and requirements; its validation is done by presenting and discussing the outcomes to a multidisciplinary group of stakeholders at NTS Norma.

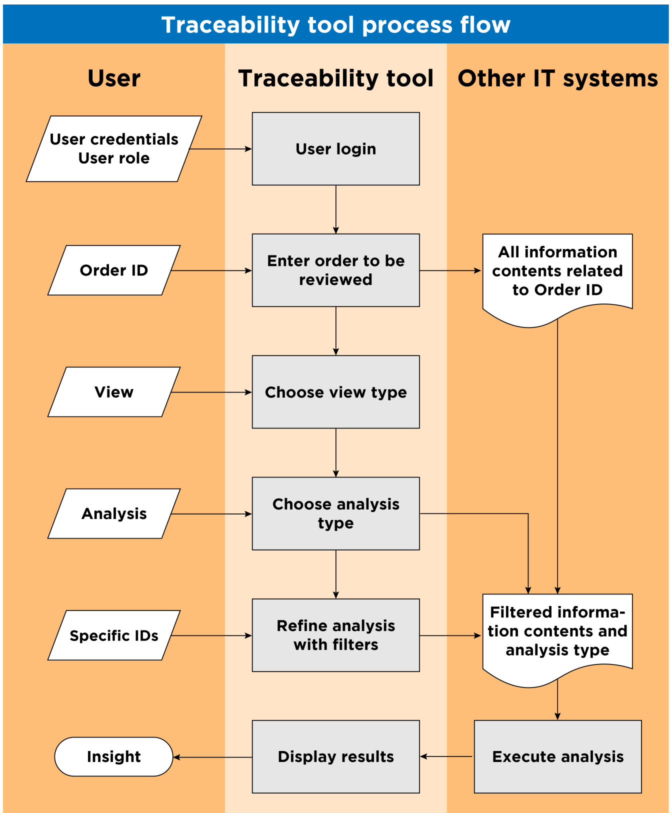
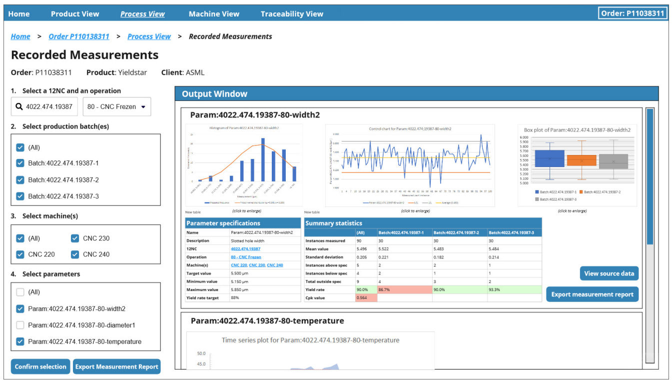
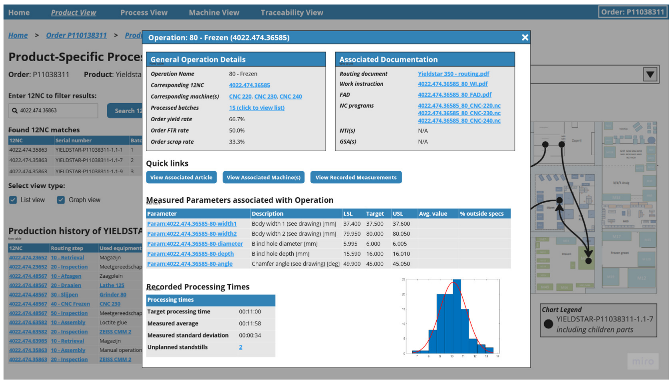


Fig. 12: The general process flow of the interaction between the traceability tool, user, and other IT systems.



(a) A dashboard in the process view (analysis: recorded measurements).



(b) A pop-up window displayed on top of the analysis (analysis: production history, pop-up: operation).

Fig. 13: Two sample dashboards of the traceability tool. All of the dashboards can be found in Appendix C, *Traceability Tool Prototype*.

4.4.1. Case study verification

In terms of verification, the outcomes of the case study seem to fulfil most of the previously proposed design principles, guidelines and requirements. The PDCA approach towards improving information traceability matches the current development strategy of NTS Norma, due to its iterative nature. Additionally, the four components that are required for information traceability (as per Section 2.1.1: identification, process inputs, process outputs, and traceability protocols) are embedded into the PDCA approach. The IT architecture has been designed to support the PDCA approach. The interlinked data identifier landscape of Fig. 6 was leading in the design of the general architecture as proposed in Section 3.3.5; this proposal formed the basis of the IT architecture of NTS Norma, in which system contents and boundaries have been clearly defined. Finally, the traceability tool forms the interface between the various systems in the IT architecture and the user. The traceability tool offers structuring of the captured information content on various levels of aggregation. Viewpoints, analyses and filters have been incorporated in order to allow the user to view the information contents in different perspectives. The tool fully supports the PDCA approach, aiding stakeholders in evaluating the manufactured products as well as their production processes and providing decision-making support for subsequent iterations. The prototype of the tool, despite being a relatively simple mock-up, visualises the basic user experience and makes use of realistic examples.

It is important to note, however, that many of the requirements are difficult to verify without testing. For instance, the design principles as proposed in Section 3 - and their materialisation in the case study - can only be truly evaluated when a sufficiently realistic prototype test is carried out. Within the scope and time of this research, it was not feasible to set up and test the proposed IT architecture in a realistic scenario. Instead, it was decided to focus on developing the traceability tool further, in order to clearly demonstrate to stakeholders at NTS Norma the added benefits of adequate information traceability.

4.4.2. Case study validation

Because verification of the case study outcomes is rather difficult due to a lack of realistic testing, this evaluation relies more heavily on validation by stakeholders at NTS Norma. The outcomes of the case study were presented to the same group of stakeholders that provided input for scoping the case study. The presentation that was shown to this group of stakeholders can be viewed in Appendix D, *Case Study Presentation at NTS Norma*. The goals of the presentation were to show the outcomes of the case study as presented in Section 4.3, to receive feedback and to start a discussion between the stakeholders.

The reactions received from the stakeholders were mostly positive. The manager of the department of manufacturing engineers, for instance, was convinced that the presented approach will have major benefits for the implementation and effective usage of statistical process control (SPC) and will allow them to mature production processes faster - the traceability tool being of particular help for this matter. The manager of the current PLM system confirmed that it is indeed necessary to establish clear boundaries and information flows between the IT systems. The managing director remarked that it would be extremely useful to be able to see the status of currently active production orders and the location of unfinished products on the shop floor, which is enabled by the proposed architecture. Finally, representatives of the IT department saw the proposal as an initial step in the right direction for promoting discussion on the key question that shapes the IT architecture: what information should be stored and managed in which system? This discussion indeed took place after the presentation. Questions that were raised by the attendees built on the ideas that were presented, predominantly asking for more details on certain aspects of the newly proposed IT architecture. The proposal differs greatly from the current architecture, relying much less heavily on ERP (for tasks such as creating MBOMs, product routings and planning production orders) and putting much more emphasis on the integrated use of PLM and MES (the latter of which is currently not being used at NTS Norma) aside from ERP. Because of this, it was quite challenging for stakeholders to formulate their questions from the perspective of the new proposal. Some questions were based on assumptions that only hold in the current 'paradigm'; for instance, one of the stakeholders expressed their concern about assigning the production planning activity to MES, since the required resources for such a planning would be located in ERP. This would be a valid concern for the current IT architecture, but the new proposal assumes a different distribution of information contents across the systems.

However, there were also some questions which resulted in concrete improvement suggestions for the IT architecture. Most notably, stakeholders questioned the necessity and added benefits of dividing all information contents primarily over ERP, PLM, and MES; an example of this is the decision to store product instance information contents in the PLM environment. This gave rise to the question that resulted in a major recommended point of improvement for the IT architecture: if the traceability tool functions as an interface between the IT systems and the user and presents the information content in a neutral manner, why would it be necessary to store all that information in a 'biased', non-neutral environment such as a PLM system in the first place? PLM, MES, and ERP systems are all designed with a particular purpose in mind; as such, the information contents will be structured in these systems to fulfil their intended purposes as good as possible. By default, the PLM system maintains a product-centric view on its information content. This is very useful for answering questions related to the configuration as-built of a particular product instance, but more problematic when different questions are raised (e.g. 'Which parts have been processed by CNC machine *X* during the past two weeks, when it was displaying erratic behaviour?'). The information structure in these application-specific IT systems is quite rigid and does not allow for switching viewpoints easily; hence, the proposed IT architecture may limit the flexibility and ease of using different aspect views. The traceability tool does enable the use of different viewpoints, but should retrieve information that is stored in a neutral structure (i.e. a structure without any 'default' view), as displayed in Fig. 7. In other words, the information contents should not be influenced by their default structuring. Following this logic, it makes little sense to permanently store information contents in a non-neutral environment such as ERP, PLM or MES when this is not absolutely necessary. Instead, it would be favourable to store most of the information contents in a neutral data warehouse that can store and maintain the information contents as an unbiased network.

5. Conclusion

Within today's manufacturing landscape, many difficulties related to complex decision-making processes and achieving competitive production quota share a common characteristic of poor information traceability. Simultaneously, adequate and reliable information traceability and management is a prerequisite for many future development opportunities. This is especially true for HMLV, high-precision and high-complexity contract manufacturing environments. Depending on the client's wishes and the quality of the produced products, the design and production processes of the highly complex and customised products and its components may evolve quickly; hence, the information content associated with a product will continue growing rapidly. Due to the small production batches and restrictions imposed by the high-precision aspect, there is very little time to mature the production processes and the cost of poor quality is high. As such, it is important to utilise the data gathered during prototype and pilot production runs to its fullest extent. In order to do so, effective data management is required; generated information must be traceable to its source.

The review of relevant literature has pointed out that three aspects are essential in achieving adequate information traceability. Firstly, information entities must be linked to one another in a structured manner in order to be interpretable. The required components of a traceability system demonstrate that a traceable resource unit must be identified by means of a unique number and a physical marking. The unique identifier is used to explicitly link the correct process inputs (i.e. product type data) and process outputs (i.e. product instance data) to the associated traceable resource unit. The traceability protocol dictates how strictly the traceable resource unit and its corresponding process inputs and outputs must be traced. Since not all parts of a product are equally difficult to manufacture, it is unnecessary to achieve the same amount of detail in their traced information contents; making a distinction will allow for more efficient information management, avoiding unnecessary data recordings. Secondly, interlinking and presenting the information contents in an information structure with a corresponding ontology, it becomes possible for users to adopt different domains, views and filters to retrieve, analyse and interpret the relevant information. Thirdly, in order to accommodate for the data collection and interlinking that is required for traceability, it is necessary to design an architecture in which the various IT systems that create, store and manage information contents are well-integrated. Out of the many available software applications, ERP, PLM and MES are the three systems that play a pivotal role in the vast majority of manufacturing enterprises. ERP, PLM and MES each have their own purpose, but due to some existing overlap in functionalities between these systems, many different configurations as to which system stores and manages which information contents are possible. However, it is essential that a clear IT architecture is formed, which specifies the responsibilities of and the interfaces between each of the systems.

In an effort to improve information traceability in HMLV, high-precision and high-complexity contract manufacturing environments, an information structure was designed based on the principle of interlinking data identifiers. As presented in the literature review, creating explicit links between a part or product instance, its inputs and outputs is required for achieving information traceability. Furthermore, an information structure has been designed in which multiple views, analyses and filters can be used. As Fig. 7 shows, the created networks of information entities grow exponentially and become hard to interpret extremely fast. By using different aspect views, users are enabled to analyse the quality and history of the manufactured product instances, the executed operations, the used manufacturing equipment and the followed traceability protocols. Finally, in order to support the framework of interlinked data identifiers and the information structure, an IT architecture was proposed that divides the information contents over ERP, PLM, MES, and data warehousing and analytics. In this architecture, ERP covers all matters related to receiving, accepting and processing orders as well as purchasing materials, PLM stores all master data and data of manufactured product instances, and MES is responsible for executing production orders and gathering the necessary data. Additionally, the data warehouse and analytics platform are responsible for storing raw measurement data and outdated information, and processing raw measurement data, respectively. It must be noted that, due to the limited scope of this research, the displayed IT architecture is not representative of the true size of a realistic enterprise IT architecture and interfaces with IT systems outside this scope (e.g. customer relations management software) have yet to be defined.

The case study at NTS Norma Hengelo, a contract manufacturer in the high-tech industry, demonstrated that a well-defined IT architecture is crucial to achieve the information transparency and traceability; conversely, the lack thereof severely hinders traceability and hampers a multitude of business processes. Analysis of the current IT architecture and overall workflow demonstrated that due to the abundance of poorly integrated software solutions, the excessive use of locally stored spreadsheets and lack of document management strategies, the overview and structure of information contents are lost. To overcome these issues, a method for improving information traceability based on the general IT architecture as proposed in Section 3.3 was developed. The work done during the case study adds to the general solution concept as presented in Section 3 by providing a practical implementation approach for the information traceability principles across the life-cycle of products and projects. A 'traceability tool' was proposed as a view-only interface to the gathered data. The traceability tool retrieves information from the various components of the IT architecture and allows the user to conduct a multitude of analyses and to adopt various viewpoints on the information contents. Since the scope and time constraints did not allow for elaborate testing of the proposed IT architecture and workflow in a realistic setting, the evaluation of the case study outcomes relies mostly on validation by stakeholders within NTS Norma. Upon presenting the outcomes of the case study, most stakeholders could clearly see advantages of the proposed solution for NTS Norma in general and their own area of expertise in particular. Overall, the proposed solution was regarded as a good first step in the right direction and was believed to significantly contribute to increased insight in the gathered information and corresponding processes. A major point of discussion was the design decision to store the information contents primarily in application-specific IT systems, such as ERP, PLM, and MES, since this may limit the flexibility of the information network. Due to the inherently biased views on the information contents by these systems, the disadvantage of this architecture is that the method of storing and default structuring the information may influence the contents and links.

While it is too early to verify the general applicability of the presented solution for the analysed type of manufacturing environment, the provided framework has proven to be useful in stimulating and guiding the debate on improving information management and traceability on an enterprise-wide level. In addition, the presented concepts have shown to aid stakeholders within these manufacturing environments in recognising that adequate information traceability is a prerequisite for overall improvement, competitiveness and growth. With an adequate information management strategy and supporting IT architecture in place, high-tech contract manufacturers will possess a potent tool that can assist in decision-making processes and facilitate more efficient and effective navigation through development projects.

6. Recommendations and Outlook

Since the opportunities to test the proposed solution during this research were limited, the main recommendation for future development of this traceability concept is to verify and improve the proposed solution through testing. A plausible testing approach would be to create and implement a prototype of the proposed IT architecture on a small scale at first, e.g. by processing the information contents of one project according to the proposed method. A learning factory or a set of sensor-equipped production machines could aid in gathering measurement data. In multiple iterations, the scale of the project could be increased; for instance, a project may initially comprise of the manufacturing of a (rather simple) prototype, and be increased in scale until it resembles a project with the complexity and life span duration corresponding to HMLV, high-complexity, high-precision manufacturing industry.

While further developing and testing the solution proposed in this work, it is of importance to bear in mind that the eventual implementation of a new IT architecture in a corporate setting is an extremely impactful event. Hence, it is equally important to develop an implementation roadmap. The new IT architecture is part of a larger project that focuses on the improvement of many business processes; in order to achieve this, adequate change management is required across the entire organisation. In order to ensure a smooth transition to the new architecture and procedures, it would be beneficial for all involved stakeholders to cultivate an understanding of how their methods of operation influence the quality of projects, processes, and products. By involving all stakeholders in this transition process and creating a more user-friendly IT environment, the quality of gathered information contents will improve.

During the presentation of the case study outcomes, a major point of discussion was the design decision to store the information contents primarily in application-specific IT systems, such as ERP, PLM, and MES. The traceability tool aims to make use of the neutrality of the network of information contents in order to display the information from different viewpoints and use analyses and filters to provide insights to a wide range of stakeholders. However, the use of application-specific IT systems in storing all of the information contents may limit the flexibility of the information network, since these systems are built to rigidly maintain certain links between information contents, while other links may not be relevant to a particular system. Hence, it is recommended that the proposed IT architecture be altered to store more information contents in a truly neutral environment. The data warehouse that is already incorporated in the IT architecture could store the information contents as a network, maintaining all links between data identifiers. Simultaneously, only the essential information contents will be stored within the aforementioned application-specific systems. For instance, the PLM system will still contain the master data of product types, since revision management is the core task of PLM; however, product instance information will be stored in the data warehouse. When the data warehouse stores the vast majority of information contents, neutrality of the information network and all of its contents and links can be maintained since no pre-structuring is applied. This simplifies the process of adding more viewpoints, analyses, and filters to the traceability tool according to the wishes of stakeholders. Moreover, this method of storing information will likely ease the transition of one IT system to another.

References

[1] Avvaru, V., Bruno, G., Chiabert, P., Traini, E., 2020. Integration of PLM, MES and ERP systems to optimize the engineering, production and business. IFIP Advances in Information and Communication Technology 594, 70–82. doi:10.1007/978-3-030-62807-9_7. ISBN: 9783030628062.

[2] Ben Khedher, A., Henry, S., Bouras, A., 2011. Integration between MES and product lifecycle management, in: ETFA2011, pp. 1–8. doi:10.1109/ETFA.2011.6058993. ISSN: 1946-0759.

[3] Chan, M., Chung, W., 2002. A framework to develop an enterprise information portal for contract manufacturing. International Journal of Production Economics 75, 113–126. doi:10.1016/S0925-5273(01)00185-2.

[4] D'Antonio, G., Sauza Bedolla, J., Chiabert, P., Lombardi, F., 2015. PLM-MES integration to support collaborative design, in: Proceedings of the International Conference on Engineering Design, ICED, pp. 81–90. ISSN: 2220-4334 Issue: DS 80-10.

[5] D'Antonio, G., Macheda, L., Sauza Bedolla, J., Chiabert, P., 2017. PLM-MES integration to support industry 4.0, in: Ríos, J., Bernard, A., Bouras, A., Foufou, S. (Eds.), Product Lifecycle Management and the Industry of the Future, Springer International Publishing. pp. 129–137. doi:10.1007/978-3-319-72905-3_12.

[6] El Kadiri, S., Kiritsis, D., 2015. Ontologies in the context of product lifecycle management: State of the art literature review. International Journal of Production Research 53, 5657–5668. doi:10.1080/00207543.2015.1052155.

[7] Jansen-Vullers, M.H., van Dorp, C.A., Beulens, A.J.M., 2003. Managing traceability information in manufacture. International Journal of Information Management 23, 395–413. URL: https://www.sciencedirect.com/science/article/pii/S0268401203000665, doi:10.1016/S0268-4012(03)00066-5.

[8] Lutters, D., 2001. Manufacturing Integration Based on Information Management. Ph.D. thesis. Universiteit Twente. Drienerlolaan 5, Enschede, The Netherlands.

[9] Nigischer, C., Hennig, M., Gerhard, D., 2019. A concept to integrate manufacturing execution and product data management systems. IFIP Advances in Information and Communication Technology 565, 315–324. doi:10.1007/978-3-030-42250-9_30. ISBN: 9783030422493.

[10] Olsen, P., Borit, M., 2018. The components of a food traceability system. Trends in Food Science & Technology 77, 143–149. URL: https://www.sciencedirect.com/science/article/pii/S0924224417304107, doi:10.1016/j.tifs.2018.05.004.

[11] Pahl, G., Beitz, W., Feldhusen, J., Grote, K.H., 2007. Engineering Design. Springer. URL: http://link.springer.com/10.1007/978-1-84628-319-2, doi:10.1007/978-1-84628-319-2.

[12] Panetto, H., Dassisti, M., Tursi, A., 2012. ONTO-PDM: Product-driven ONTOlogy for product data management interoperability within manufacturing process environment. Advanced Engineering Informatics 26, 334–348. doi:10.1016/j.aei.2011.12.002.

[13] Regattieri, A., Gamberi, M., Manzini, R., 2007. Traceability of food products: General framework and experimental evidence. Journal of Food Engineering 81, 347–356. URL: https://www.sciencedirect.com/science/article/pii/S0260877406006893, doi:10.1016/j.jfoodeng.2006.10.032.

[14] Schuitemaker, R., Xu, X., 2020. Product traceability in manufacturing: A technical review. Procedia CIRP 93, 700–705. doi:10.1016/j.procir.2020.04.078. ISSN: 2212-8271.

[15] Slot, M., Fraikin, M., Damgrave, R., Lutters, E., 2022. Digital infrastructures as the basis for implementing digital twinning. Procedia CIRP 109, 568–573. URL: https://www.sciencedirect.com/science/article/pii/S2212827122007454, doi:https://doi.org/10.1016/j.procir.2022.05.296. 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world.

[16] Slot, M., Lutters, E., 2021. Digital twinning for purpose-driven information management in production. Procedia CIRP 100, 666–671. URL: https://www.sciencedirect.com/science/article/pii/S2212827121006168, doi:https://doi.org/10.1016/j.procir.2021.05.141. 31st CIRP Design Conference 2021 (CIRP Design 2021).

[17] Theisens, H., 2021. Lean Six Sigma Green Belt - Mindset, Skill set and Tool set. Climbing the Mountain. fifth ed., Lean Six Sigma Academy.

[18] Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A., 2010. Facilities Planning. John Wiley & Sons.

[19] Valve, D., 2023-03-29. Build to print or spec for your custom fabrication project. URL: https://dantevalve.com/news/build-to-print-vs-build-to-spec/.