

# Chapter 3

## Life Cycle Management for Product-Service Systems



**Stefan Wellsandt, Laura Cattaneo, Daniele Cerri, Sergio Terzi,  
Donatella Corti, Christian Norden and Reinhard Ahlers**

**Abstract** Product-Service Systems (PSS) and the Internet of Things (IoT) are two related concepts. This chapter describes an approach to manage PSS along its life cycle. It includes a design methodology for PSS and a systems modelling method. The former supports designers in defining PSSs that incorporate monitoring, control, optimization or autonomy. It includes a new method to assess a product's functionality in terms of the data needed for its realization. The latter adopts life cycle thinking and employs a modelling language to outline the PSS and its various components and actors. A life cycle performance analysis could benefit from the model by extracting cost information from it for further analysis. This chapter highlights challenges related to PSS life cycle management observed during the Manutelligence project. They concern the design methodology and the applied life cycle modelling method.

---

S. Wellsandt (✉)

BIBA—Bremer Institut für Produktion und Logistik GmbH at the University of Bremen,  
Hochschulring 20, 28359 Bremen, Germany  
e-mail: wel@biba.uni-bremen.de

L. Cattaneo · D. Cerri · S. Terzi

Department of Economics, Management and Industrial Engineering, Politecnico di Milano,  
20156 Milan, Italy  
e-mail: laura1.cattaneo@polimi.it

D. Cerri

e-mail: daniele.cerri@polimi.it

S. Terzi

e-mail: sergio.terzi@polimi.it

D. Corti

Department of Innovative Technologies SUPSI, University of Applied Sciences and Arts  
of Southern Switzerland, 6928 Manno, Switzerland  
e-mail: donatella.corti@supsi.ch

C. Norden · R. Ahlers

Balance Technology Consulting GmbH, 28203 Bremen, Germany  
e-mail: christian.norden@bal.eu

R. Ahlers

e-mail: reinhard.ahlers@bal.eu

© The Author(s) 2019

L. Cattaneo and S. Terzi (eds.), *Models, Methods and Tools for Product Service  
Design*, PoliMI SpringerBriefs, [https://doi.org/10.1007/978-3-319-95849-1\\_3](https://doi.org/10.1007/978-3-319-95849-1_3)

### 3.1 Product-Service Systems and the Internet of Things

A PSS is an integrated product and service offering that delivers value in use to the customer. There are several potential benefits of PSSs. First, there is an increase of revenues, because services tend to have higher profit margins and can provide a stable source of revenues. Second, they are means to differentiate offers in mass-markets, which are typically characterized by commodities and technologies. Third, services bring a decrease of variability and volatility of cash flow throughout the life of a product resulting in a higher shareholder value [1]. PSS is a complex concept, because it is composed of several parts that need to be managed: a product and one or more related services, a network of stakeholders and a supporting infrastructure [2]. The different product aspects, such as the concept and design phases and the life cycle, have been analyzed deeper by many authors [3], while the service topic is more recent and first attempts exist for structuring the service design [4].

The IoT technologies promote the business model of PSS. They enable new services related to the connectivity and the usage of environmental data. Connected smart products offer expanding opportunities for new functionality, greater reliability, higher product utilization and capabilities that cut across and transcend traditional product boundaries [5]. Terms like Cyber-Physical Systems (CPS), IoT and Virtual Reality have become ordinary in industries. IoT products raise a new set of strategic choices related to

- how value is created and captured,
- how the amount of data they generate is utilized and managed, and
- what role companies should play as industry boundaries are expanded [5].

Smart products have three main components: physical components, smart components and connectivity components. Smart components are directly connected with services related to the physical parts, while connectivity allows exchanging information between the product and its environment, and enables services to exist outside the physical product itself [5].

Intelligence and connectivity enable an entirely new set of product functions and capabilities, which can be grouped into four areas: monitoring, control, optimization, and autonomy [5]. A product can potentially incorporate all four. Each capability is valuable in its own right and sets the stage for the next level. For example, monitoring capabilities are the foundation for product control, optimization, and autonomy.

- **Monitoring:** smart, connected products enable the monitoring of a product's condition, operation and external environment through sensors and external data sources. A product can alert users or others stakeholders to changes in circumstances or performance. Monitoring also allows companies and customers to track a product's operating characteristics and history. It improves the understanding of how the product is actually used. The collected data has implications for: (1) design by reducing over-engineering and improving market segmentation through the analysis of usage patterns by customer type; (2) after-sale service by allowing the dispatch of the right technician with the right part to improve the first-time fix

rate. Monitoring data may also reveal warranty compliance issues, as well as new sales opportunities, such as the need for additional product capacity because of high utilization.

- **Control:** smart, connected products can be controlled through remote commands or algorithms that are built into the device or reside in the product cloud. Algorithms are rules that direct the product to respond to specific changes in its condition or environment (for example, “if pressure gets too high, shut off the valve” or “when traffic in a parking garage reaches a certain level, turn the overhead lighting on or off”). Control through software embedded in the product or the cloud allows the customization of product performance to a degree that was not cost effective or often even possible before. The same technology also enables users to control and personalize their interaction with the product in many new ways. For example, users can adjust their smart light bulbs via smartphone, turning them on and off, programming them to blink red if an intruder is detected, or dimming them slowly at night.
- **Optimization:** smart, connected products can apply algorithms and analytics to in-use or historical data to improve output, utilization, and efficiency. In wind turbines, for instance, a local microcontroller can adjust each blade on every revolution to capture maximum wind energy. Each turbine can be adjusted to improve its performance and to minimize its impact on the efficiency of those nearby. The rich flow of monitoring data from smart, connected products, coupled with the capacity to control product operation, allows companies to optimize product performance in numerous ways, many of which have not been previously possible.
- **Autonomy:** monitoring, control and optimization capabilities combine to allow smart, connected products to achieve a previously unattainable level of autonomy. At the simplest level an autonomous product does operation using sensors and software on real time. More-sophisticated products are able to learn about their environment, self-diagnose their own service needs, and adapt to users’ preferences. Autonomy not only can reduce the need for operators, but it can improve safety in dangerous environments and facilitate operation in remote locations. Autonomous products can also act in coordination with other products and systems. The value of these capabilities can grow exponentially as more and more products become connected.

Connected products require the design of a new technology infrastructure made of multiple layers. It is necessary to identify sensors, software, data storage, user interfaces as well as ports, protocols and kind of connections to design the PSS.

The IoT could underpin the PSS design by using information feedback at any stage of the Product-Service life cycle. Moreover, IoT provides information for continuous improvement, closer relationship with stakeholders, resource efficiency and the ability to meet sustainability. IoT provides the opportunity to obtain information about some relevant parts of the system in real time, during all the life cycle processes, thus representing a differential in order to build, design, implement and improve a PSS.

### 3.2 Life Cycle Thinking, Modelling and Management

The following paragraphs adopt the findings of a literature survey published in the International Journal for Product Life cycle Management [6]. The study investigated the constituents of life cycle models.

**Life cycle** is a concept used in Biology where it describes the recurring change of states for certain organisms (see [7] for biological life cycle). In engineering, the organisms are exchanged for tangible goods (products). Processes, such as design, production, use, repair, recycling and disposal substitute the organism's state changes. From the perspective of business economics, the concept of the product life cycles was introduced by Theodore Levitt in the 1960s [8]. Levitt's approach describes the life cycle of products by four phases: introduction into the market, rapid growth of sales volume, market saturation and market decline. The introduction of Levitt's product life cycle model supports the strategic management of products. Manutelligence focused on the engineering life cycle.

**Life cycle thinking** means that people have a life cycle model in mind that affects the scope of their activities during, for instance, system development and production. In Manutelligence, a **life cycle model** describes activities related to a PSS in a simplified way. It consists of elements that represent the transformation of the product, software or service over time, whereas the notion of 'transformation' is adopted from operations management [9]. A *transformation* changes inputs (e.g. raw material) into outputs (e.g. machine) of a higher quality and a higher value. Processes and/or the stakeholders performing specific processes represent the transformation.

Technological advances, such as the IoT, resulted in developments that influence life cycle thinking. Effects of these developments and common characteristics of a PSS's life cycle are summarized in Table 3.1. The presented concepts are not meant to be comprehensive.

Examples for the influence of life cycle thinking on engineering are Life Cycle Assessment and Life Cycle Costing. The adoption of life cycle thinking in industry and academia led to the emergence of a plethora of different life cycle models as investigated by Wellsandt et al. [6]. Manutelligence focused on the flow of codified information along the life cycle.

**Life cycle management**, in the Manutelligence project, focuses on the management of information along the PSS life cycle. The specialization on PSS is a rather new discipline, because the adoption of Servitization, Intelligent Products and the Internet of Things is not very high in most industries. Consequently, the experience with this PSS life cycle management is low, compared to the information management for tangible products or software. One of the challenges of PSS life cycle management is the fact that a PSS consists of integrated hardware, software and service components. This is especially true, if the PSS has connectivity-based features. In this context, integration has several meanings, for example, the:

- 1 Integration of ICT into traditional products (Intelligent/Smart Products).
- 2 Integration of hard- and software development (Concurrent Engineering).
- 3 Integration of PSS software into third party software (Interoperability).
- 4 Integration of services into existing business processes.

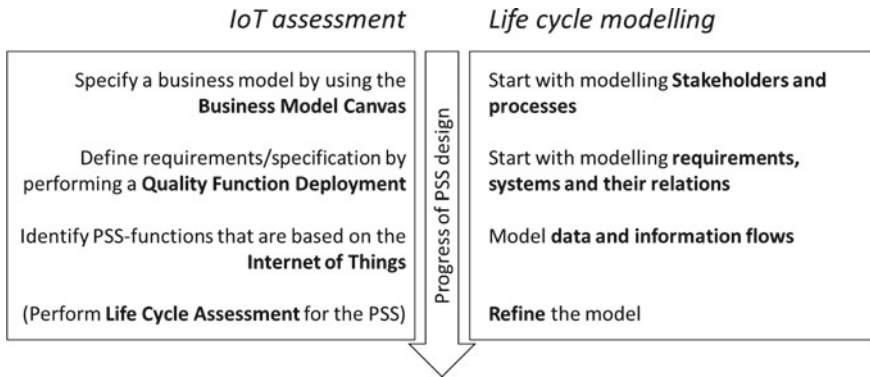
**Table 3.1** Concepts of PSS life cycles

Name	Description
Processes and stakeholders	The cornerstones of the life cycle model are processes and the stakeholders being affected by them
Flows and stocks	Information, material and energy are the constituents that are exchanged between processes to create value and quality
Process abstraction	The life cycle model consists of different process abstraction levels. In Manutelligence, three layers of abstraction were differentiated. These are life cycle phase, process, and activity
System abstraction	Complex systems consist of sub-systems. They can be represented on different abstraction layers in a life cycle model. A common system abstraction is the differentiation of product, assembly and component
Information abstraction	A system class describes systems with shared characteristics. It represents a high level of abstraction. An instance describes an entity that features a unique characteristic, such as a product identifier. It is the lowest abstraction layer for products used in Manutelligence
Product states	A product's state is described by using item-level information, collected at a specific time and place [10]. Each state consists of a set of relevant "state characteristics" measured at specific checkpoints during the life cycle
Structure	Life cycle processes can be structured as a linear or circular sequence. If two linear life cycles share a process, a cross structure is established
Nesting	A system's life cycle can be described as a nested structure of sub-system life cycles. This way, a complex structure of several related life cycle models is created

Each of these integrations represents a complex problem with many variables that can relate to each other. The problems are dynamic because they change with the rapid advances in ICT technology (e.g. the fast spread of Blockchain technology). Solving the integration problems benefits from an extensive understanding of, for instance, the PSS's components and stakeholders. For this reason, one assumption of Manutelligence was that life cycle modelling helps developers and other stakeholders to understand the complexity of a PSS during the design phase. The actual modelling task is a supplement to standard procedures in product design. These procedures include, for instance, the collection and analysis of requirements, and the development of system components and their interfaces.

The Manutelligence partners developed an approach to support the design of IoT-related PSSs. It consists of a PSS design methodology and the concurrent life cycle modelling. Figure 3.1 illustrates this approach.

The following sub-chapters describe the proposed design approach. It starts with the new design method for the IoT Assessment. This chapter omits a description of the Business Model Canvas (BMC) and Quality Function Deployment (QFD) methods, because literature has plenty of descriptions for them already [11, 12]. The second part of this chapter introduces the life cycle modelling and the life cycle performance analysis approach.



**Fig. 3.1** Manutelligence PSS design approach

### 3.3 Approach to Design IoT-Related PSS

#### 3.3.1 Design Methodology

This part of the chapter is based on the Manutelligence public Deliverable 2.2 [13]. The main goal of the methodology is to structure the PSS's concept and design processes with a focus on the identification of the IoT technologies needed to catch and elaborate data for customer need satisfaction, and for the design of suitable services. The idea is to connect different activities and methods, to elicit data and information. The methodology is modular, different activities are optional, and they do not need to be performed in a specific order. Within these main activities, we include the definition of the business model, the definition of the stakeholders' requirements and the definition of the IoT capabilities. Besides these three tasks, there are other tasks to be considered, especially concerning the design of the actual solution components. The main activities should be completed and integrated with other important information, as well as the addition of hard and software functionalities and business processes.

We will focus on the definition of the IoT capabilities, which represents the innovative part of the methodology developed during the Manutelligence project.

#### 3.3.2 IoT Assessment Support

This activity represents the core part of the proposed methodology, since it defines the bases for the PSS design phase. The outputs of this process will be the Bill of Material (BOM) of the product, the definition of the connected services and the identification of data, information and knowledge that should be managed and analyzed during the Product-Service life cycle. Connect products require the design of a new technology

infrastructure made up of multiple layers. It is necessary to identify sensors, software, data storage, user interfaces as well as ports, protocols and kind of connections to design the PSS. Furthermore, it is necessary to satisfy the stakeholders' requirements, which should be previously analyzed, for instance with a QFD.

Usually the Product Design and the Service Design processes are separately performed within specialized teams. This produces solutions that often are not really integrated. The innovation of the Manutelligence methodology aims to improve the integration between Product and Service Design, in order to give to customers the best PSS solution.

Starting from the requirements defined in a previous phase, the team will implement the Product-Service Design with the primary aim of defining the IoT capabilities that should be integrate in the product life cycle, in order to properly design the connected services.

The idea is to develop a graphical tool that can be used by designers to identify and classify the IoT capabilities that should be embedded in the product. This tool has to include:

- The classification of the capabilities of the smart product, which can be grouped into four areas [5]: monitoring, control, optimization and autonomy. This classification specifies the kind of service a company wants to implement thanks to the IoT technology.
- The definition of data destination. It is necessary to define the stakeholder who needs a specific kind of information (e.g. customer, designer, and maintainer). This is important also to identify the format with which data or analysis results should be shown and the support that must be used, in order to read and share data and results. For example, customers would like to have access to data directly from their smartphone.
- The identification of the product parts in which sensors should be installed in order to collect data.

To implement the graphical tool, we exploit the BOM, which should be developed in the Product Design phase. The BOM is particularly useful to identify the product's parts. Parts and components define a specific number of details and levels. For each of these levels, we design a specific graphic structure. For instance, for a smart car, at level 0 we have the vehicle. Level 1 composes of the Power Unit, the Gear Box and the Car Body (Fig. 3.2).

For each level, the product's parts are reported as input for the designed table. The first column identifies the data destination and the stakeholders involved in the service. The entries of the table are colored following the capabilities classification (i.e. monitoring is green colored, control is yellow colored, optimization is orange colored and autonomy is red colored,) and the entries are filled indicating the variable. If a box remains white, this means that no IoT capabilities are defined for that entry. The IoT capability is selected, in order to satisfy the product and service requirements. This classification is an important initial stage, since a specific IoT capability defines the correspondent technology to select, for example, sensors or data storage systems and clarify the kind of connection that must be developed.

		# Level				
		Part 1	Part 2	Part 3	Part 4	Part 5
Data destination	Software	Control		Monitoring		
	Designers					
	Maintenance				Monitoring	
	App		Monitoring		Optimization	

Fig. 3.2 Illustration of the IoT graphical tool

### 3.4 Life Cycle Modelling Support

This chapter is based on a journal paper published in the International Journal for Product Life Cycle Management [6] and the Manutelligence public Deliverable 2.1. The paper’s goal was to investigate life cycle models and identify common elements and structures of these models. For this purpose, 71 models from literature were investigated leading to the aforementioned concepts of PSS life cycle models.

A modelling language that supports many of the concepts described in Table 3.2 was used for the actual modelling task of the use cases. For this purpose, the Lifecycle Modelling Language (LML) was selected in Manutelligence [14]. It is a derivative of the Systems Modelling Language (SysML). LML introduces an Ontology that describes the system life cycle with specified entities and their relations amongst each other. The modelling task was supported by the browser-based systems engineering tool Innoslate (<https://www.innoslate.com/>). It has native support for LML—the software provider appears to be involved in the development of the LML standard. LML’s features and the capabilities of the modelling tool Innoslate were mapped against the aforementioned life cycle concepts. A summary of this mapping is presented in Table 3.2.

LML suggests 12 entities and several visualization techniques for system life cycle modelling. The Innoslate-functions used in Manutelligence mainly base on the entities defined in LML. Instances of the entities build the life cycle model. Some Innoslate-functions relate to visual techniques used by the software (e.g. abstraction layers). The mapping between life cycle concepts and LML is mainly achieved by using existing entities and their relations from the LML Ontology. Process and system abstraction is achieved through the relations between activities and assets (e.g. decomposes and decomposed of). Different structures of a life cycle can be created by linking activities through an activity diagram. No support was identified in LML for the nesting concept. Innoslate allows users to name a process as “life



**Table 3.2** Mapping between life cycle concepts, LML and Innoslate

Life cycle concept	LML features	Innoslate functions
Processes and stakeholders	Activity and Asset	Via LML
Flows and stocks	Input/Outputs and Activities	Via LML
Process abstraction	Relation between Activities	Visual abstraction layers
System abstraction	Relation between Assets	Visual abstraction layers
Information abstraction	No support identified	No support identified
Product states	No support identified	No support identified
Structure	Activity diagram	Via LML
Nesting	No support identified	Visual separation of processes

cycle” and decompose it. In this case, several life cycle containers can be created with related Activity entities. A shortcoming of LML and Innoslate is the lacking support for item level information. This also limits the support for the product state concept. LML and Innoslate can be replaced by other modelling approaches that support the concepts presented in Table 3.1.

### 3.4.1 Life Cycle Analysis Support

This chapter is based on a conference paper for the International Conference on Engineering, Technology and Innovation held 2017 [15]. The paper’s goal was to present and discuss an approach for life cycle analysis using a shared life cycle model. Life cycle analysis for traditional products includes methods, such as “Life Cycle Costing” (LCC) and “Life Cycle Impact Assessment” (LCA). These methods were selected for Manutelligence, because they represent widely acknowledged life cycle analyses.

**Life Cycle Costing (LCC)** is an accounting method, which considers every cost flow throughout the life cycle of a product [16]. **Life Cycle Performance Assessment (LCPA)** enhances the LCC-approach by extending the assessment to cash inflows. The key performance indicator of the LCPA-approach is the net-present value. It reflects the point of time of each considered cash flow throughout the life cycle. Consequently, the cash flows of the assessed object discount with an interest rate depending on its specific point in time. Moreover, LCPA pursues an approach that allows to compare investments with each other.

The quality of LCPA results depends on the input data quantity and quality. Every cash flow throughout the life cycle has to be identified with its cash flow type and its point in time in the life cycle. Usually, the information needs to be tied to a specific life cycle phase and quantified with its time-based value. In many cases, this requirement leads to a time series of cash flows, which needs to be archived in the

ex-post perspective or forecasted in the ex-ante perspective. In the next steps, the gathered data has to be translated into a software-tool to perform the LCPA.

The general LCPA approach can be adapted for the assessment of PSS. Besides the relevant input data for products, such as investments, energy costs, maintenance and other operating costs, the analysis of the service part requires input data that is more focused on, for instance, personnel costs, equipment usage to perform the service, travel costs, as well as service fees as revenues. As a result, the comparative LCPA approach enables the evaluation of different PSS concepts with each other.

The main reference to carry out a **Life Cycle Impact Assessment (LCA)** is the ISO 14000 family of standards. They provide a set of international guidelines quite stable, subjected to periodic updates, internationally recognized and used as reference at global level. The identified and renowned structure of an LCA study is organized into four phases:

- **Definition of the goal and scope of LCA.** It defines why the LCA is performed and what the system boundaries are.
- **Life Cycle Inventory (LCI) analysis.** The exchanged natural elements and their quantities of the resources and emissions entering (e.g. raw material, energy and ancillary material) and leaving the system of interest (e.g. emission, waste, products and co-products) have to be defined. This step is complex and time consuming, since it involves data collection from several different actors and related processes along the supply network. The allocation process of shared resources is a critical point and has an impact on the final result.
- **Life Cycle Impact Assessment (LCIA).** It is meant to calculate the impacts on the environment generated by the identified LCI data.
- **Interpretation.** In the final phase, the report with the quantified impacts is prepared and the critical review of the LCA results is performed.

The ISO 14050 standard has been developed with a physical product focus, even though the provided definition of “product” states “any good or service“. This means that services are conceptually considered, yet the PSS concept is not considered explicitly. For this reason, the ISO-based LCA approach might not be applicable to the PSS context directly. The main difficulty when carrying out an LCA for a PSS is how to integrate the service component into the LCI. Corti et al. proposed an approach to support the LCI phase for PSSs [17]. It could formalize the integration of information related to the service part of the offer. Categories of information and their positioning along the life cycle of the PSS are taken into consideration showing the backbone structure that is recommended to carry out the LCI for a PSS. The life cycle divides into its three main phases, namely Beginning of Life (BOL), Middle of Life (MOL) and End of Life (EOL), whilst the data categories are related to either the product component or the service component.

Guidelines for gathering the information to support the LCA methodology are still missing [18, 19]. Dal Lago et al. propose a conceptual work in this direction [20]. The authors develop a set of guidelines to gather the information needed to instantiate and support an LCA analysis from a PSS life cycle perspective.

**Table 3.3** Approach for model-based life cycle analysis for PSS

Step	Realized
Identify elements needed for analysis	Manual task
Create life cycle model	Modelling tool
Export model into xml file	Modelling tool
Import model from xml file	Analysis tool
Run analysis	Analysis tool

A shared life cycle model supports the PSS life cycle analyses in Manutelligence. Table 3.3 summarizes the proposed approach.

The proposed approach is most efficient, if multiple analyses refer to the same life cycle model. In this case, redundant information collection and modelling is avoided. The *first* step of a model-based life cycle analysis for PSS is the identification of relevant elements from the PSS life cycle. Relevance depends on the necessity to represent an element in the analyses. For instance, it might be necessary to differentiate a product and its components, stakeholders, and activities during the usage phase. In some cases, it could be relevant to omit activities from the PSS life cycle, because their expected influence on cost or environmental impact is low in the targeted application case. The *second* step is the creation of a life cycle model that contains information about the analysis elements. Manutelligence realized this through a model grounded on the LML Ontology. It was developed and maintained in Innoslate. The tool features an xml-file export of the entire model and the Ontology—this effectively represents the *third* step of the proposed approach. Once the model is in a common data format, it can be imported into the actual analysis tool. The import of the life cycle model represents the *fourth* step. Finally, the analysis is initialized with the information stored in the imported life cycle model. This *fifth* step is the final one of the proposed approach for model-based life cycle analysis of PSS. The approach was tested with an LCPA tool.

### 3.5 Challenges for Product-Service Systems Management

This chapter partially bases on a conference paper for the International Conference on Engineering, Technology and Innovation held 2017 [15]. The following paragraphs cover different challenges concerning PSS life cycle management. The first parts summarize issues encountered during the evaluation of IoT components of a PSS. Challenges related to creation of life cycle models follow it. Each of the paragraphs aims to provide suggestions how the encountered challenges could be addressed in the future.

### 3.5.1 IoT Assessment

The IoT assessment method is the original contribution of our methodology, since other phases are actually well-known methodologies already used in several different fields [11, 12]. The IoT assessment graphical tool should be able to design how information and data collected through IoT technologies have to be integrated along the PSS life cycle. We present a list of questions that the IoT assessment tool should be able to answer and we identify which points need further developments and improvements.

**Why?** *Why we want to use IoT technologies? Which is the service we want to develop? Which particular function we want to perform?* Even if these points should come straightforward from the HoQ requirements, the IoT assessment should highlight these goals, in order to keep focus on them. The graphical tool registers the IoT capabilities that should be implemented in the PSS, as suggested in [5].

**Who?** *Who is involved, which are the different stakeholders that take part in this process?* The actual graphical tool, during its application to Manutelligence use cases, turned out to be a little bit confusing, since we have put as data destination designers, maintenance, applications and software. Actually, the presence of hardware and software is always mandatory, because data are collected and analyzed with software support. Only after these first steps, processed data could be visualized/used by someone else, for example, designers could use analytics results to improve product design and product development.

**What?** *What we want to measure? Which data and information we want to collect?* Looking at the graphical support, we have no information regarding which variables and data should be collected along the process.

**Where?** *Where we have to insert and install sensors? Which components of the product are involved in the collection of information?* We answer to this question using information coming from the BOM. At this moment we investigate all the different BOM levels, but how it turned out from its application to Manutelligence use cases, this could request too much effort than what is really needed. For example, the level of details in a car's BOM could be deep and even unnecessary for this aim. In principle, the BOM could be used only to identify and take note of parts involved in the data collection process.

**When?** *Which phase of the product life cycle is involved? When information is collected? When information and data could be used?* It could be necessary to highlight, which are the PSS life cycle phases involved in the process, also to easily identify stakeholders and different flows of data and information between them. For example, data collected from machine-embedded sensors (MOL phase) could be sent to maintainers to realize predictive maintenance (MOL phase). Data collected during the product usage (MOL phase) could be sent to designers, in order to improve product design and product development (BOL phase).

**With What?** *What kind of technology should be developed? What about hardware and software installed and used to collect, analyze and share data? Which data analysis should be performed?* In the graphical tool, we do not mention the tech-

nologies that should be put in place, in order to collect data and allow connectivity (hardware, ports, protocols and kind of connections) even if this point is important, in order to check if all the different devices are able to be connected one with each other. As some of the use cases have highlighted during project development, it is also important to perform a feasibility analysis, to compare benefits and costs of the technology investment. The graphical tool should also contain an initial definition of the statistical techniques that should be used to analyze data. When machine learning techniques are implemented from some software, data are directly processed from an algorithm and results are made available for further analysis (for example, after Cluster Analysis it could be decided to perform a Classification Rules Analysis). In some cases, the presence of a data scientist could be requested, which could take some raw or processed data and perform further analysis or post processing operations, in order to organize the output such that it is easily understandable. Only after these intermediary steps, data could be available to be read and used by other stakeholders.

### 3.5.2 *Life Cycle Modelling*

**Model complexity.** Complexity refers to the number of elements and relations in a life cycle model. The investigated use cases in Manutelligence feature around 100 elements and more than 100 relations (using the LML Ontology). The required number of elements depends on the model's purpose and on the PSS. A support of multiple life cycle analyses (and other tasks) may lead to the need to include additional elements and relations. Higher complexity will likely make the management of the model more difficult (e.g. adding, removing and updating elements). Adding elements to the life cycle model can be avoided if two or more tasks share an element. Additional relations, however, might be necessary in the case of element sharing.

**Model dynamics.** The concept of dynamics refers to the need that a model's elements and relations may evolve over time (depending on the model's purpose). In the course of the PSS design and operations, for instance, the model is subject to extension or reduction. Elements and relations change depending on the needs of the decision makers involved in the related tasks. This is especially the case when the model is also used as a PSS planning tool—as experienced in Manutelligence. Maintaining a life cycle model, i.e. keeping it updated to the most recent planning state, can be a time consuming task. The time needed to maintain a model depends, among others, on the complexity of the model. For instance, it is a difficult task to add an element and its relations to a model with several hundreds of elements and even more relations. In the case of Manutelligence, the maintenance of the model was not experienced as difficult. Possibly, because the model is not yet complex enough to affect maintenance.

**Modelling tools.** Creation and maintenance of a life cycle model can be difficult tasks that should be supported with software tools. The decision to use LML for our study limits the number of tools to support this task. The only tool that integrates it

“out of the box” is called Innoslate. Key persons from the company behind this tool are contributors to the LML specification. The logic of LML may be replicated with other tools, such as Enterprise Architect and the ARIS toolbox. Life cycle modelling tools should, in general, satisfy the needs of the stakeholders involved in the PSS life cycle. In the case of life cycle analysis, the stakeholders (analysts) need to access (read) life cycle model information that is relevant for their analysis task. A complete model export with subsequent file parsing, as realized in Manutelligence, is only a temporary solution. This is, amongst other reasons, because the parsing could miss an element that is not following the convention of the parsing mechanism. A simple example is a typo in the element name. In order to solve this and other data quality problems, a reliable life cycle model exchange mechanism is needed.

## References

1. Cavalieri S, Pezzotta G (2012) Product–service systems engineering: state of the art and research challenges. *Comput Ind* 63(4):278–288
2. Goedkoop MJ (1999) Product service systems, ecological and economic basics. Ministry of Housing, Spatial Planning and the Environment, Communications Directorate
3. Ulrich KT, Eppinger SD (2008) Product design and development, 4th edn, Internat edn. Boston: McGraw-Hill/Irwin
4. Aurich JC, Mannweiler C, Schweitzer E (2010) How to design and offer services successfully. *CIRP J Manuf Sci Technol* 2(3):136–143
5. Porter ME, Heppelmann JE (2014) How smart, connected products are transforming competition. *Harvard Bus Rev* 92(11):64–88
6. Wellsandt S, Nabati E, Wuest T, Hribernik KA, Thoben KD (2016) A survey of product lifecycle models: towards complex products and service offers. *Int J Prod Lifecycle Manage* 9(4):353
7. Bell G, Koufopanou V (1991) The architecture of the life cycle in small organisms. *Philos Trans R Soc Lond B: Biol Sci* 332(1262):81–89
8. Levitt T (1965) Exploit the product life cycle. *Harvard Bus Rev* 43:81–94
9. Grabner T (2014) Operations Management: Auftragserfüllung bei Sach- und Dienstleistungen, 2 edn (Aktualisierte Aufl.). Wiesbaden: Springer Gabler
10. Wuest T, Irgens C, Thoben K-D (2014) An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *J Intell Manuf* 25(5):1167–1180
11. Osterwalder A, Pigneur Y (2010) Business model generation: a handbook for visionaries, game changers, and challengers. Wiley, New York
12. Akao Y (2004) Quality function deployment. Productivity Press
13. Manutelligence consortium, “Manutelligence project” (2015) [Online] Available: <http://www.manutelligence.eu/>. Accessed 21 Jan 2016
14. Lifecyclemodeling.org, “Lifecycle Modeling Language” (2017) [Online] Available: <http://www.lifecyclemodeling.org/>. Accessed 29 Aug 2017
15. Wellsandt S et al (2017) Model-supported lifecycle analysis: an approach for product-service systems. In: International conference on engineering, technology and innovation, Madeira
16. International Organization for Standardization (2008) ISO 15686-5:2008—buildings and constructed assets—service-life planning—part 5: life-cycle costing. International Organization for Standardization

17. Corti D, Fontana A, Montorsi F (2016) Reference data architecture for PSSs life cycle inventory. *Procedia CIRP* 47:300–304
18. Kjaer LL, Pagoropoulos A, Schmidt JH, McAlloone TC (2016) Challenges when evaluating product/service-systems through life cycle assessment. *J Cleaner Prod* 120:95–104
19. Dal Lago M, Corti D, Wellsandt S (2017) “Reinterpreting the LCA standard procedure for PSS. *Procedia CIRP*

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

