



# GRAPH DATABASES FOR EXPLOITING USE PHASE DATA IN PRODUCT-SERVICE-SYSTEM DEVELOPMENT: A METHODOLOGY TO SUPPORT IMPLEMENTATION

C. Hollauer, B. Shalumov, J. Wilberg and M. Omer

## Abstract

Factors such as cost reductions in sensors lead to an increase of use-related data of Product-Service-Systems (PSS) users. PSS use data is a challenge as well as competitive advantage, enabling companies to better understand customers. Graph databases can be used to amange and analyze highly connected data. However, their implementation is a complex process, with low dissemination and few documented experiences for PSS companies. Consequently, we present a methodology supporting their implementation, which has been evaluated in a qualitative interview study as well as an industrial case study.

*Keywords: product-service systems (PSS), innovation, big data analysis, data driven design*

## 1. Introduction

### 1.1. Motivation

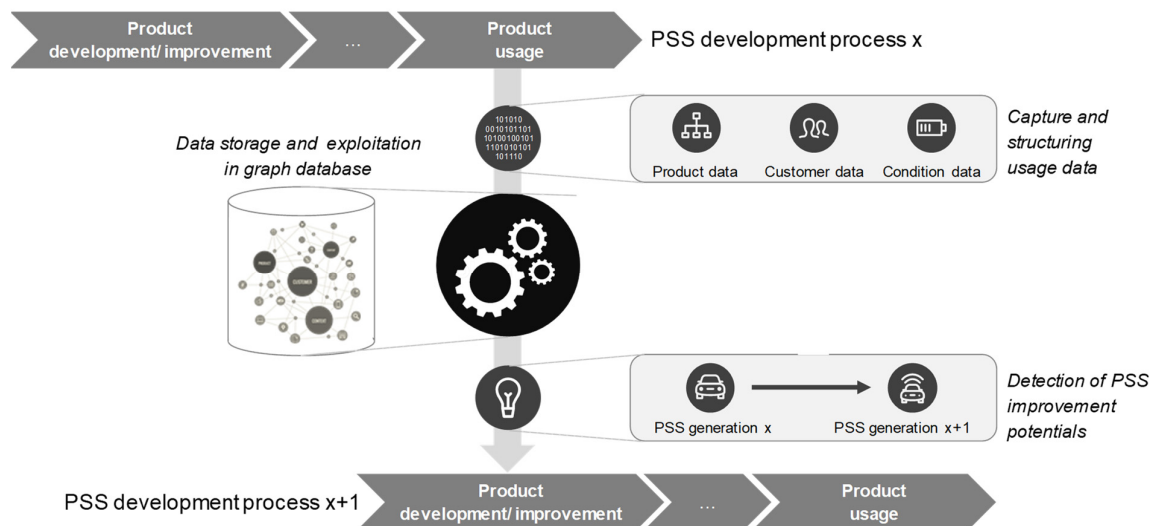
Internet giants like Google and Facebook early realised the potential of data acquired from users and used it to improve and customise their services (Marzi, 2012). Data became the most valuable asset for those companies (Sullivan, 2015). However, handling Big Data with traditional databases quickly runs up against its limits (Robinson et al., 2015). Both Google and Facebook took advantage of another trending technology: graph databases. Meanwhile, the application of graph-based data management has widely spread among software, telecommunication, and internet companies that have access to large amounts of customer usage data (Marzi, 2012). Due to increasing prevalence of low-cost micro sensors as well as future advances in information and communication technology, data acquisition in the field of Product-Service-Systems (PSS) evolved rapidly (Abramovici et al., 2011, p. 1). Industrial enterprises, following internet companies with a few years of delay, are getting overwhelmed with Big Data. Sensors not only generate data during every product lifecycle stage from the idea to the market-ready product, but also enable access to data generated during the product usage (Abramovici et al., 2016, pp. 1-2). This data could contain information about customers' preferences and usage behaviour, some of which may be even hidden for the customer himself. Kemper (2016, p. 1) points out that objects within companies become more and more connected. Nonetheless, the exploitation of usage data is still a serious challenge and often limited to department specific purposes (Ehrlenspiel, 2014, p. 30). Traditional databases do not allow fast information retrieval within the entire product data across all departments. Thus, data is often kept within a department, which hampers product developers' access to data generated during the usage of the product (Ehrlenspiel, 2014, p. 30). Reiner (2004, p. 10) emphasises the importance of early integration of customer information into the product

development process, while Barc (2014) states that only a negligible share of companies describes their technology research and product development processes as data driven, despite of clear cost advantages. Thus, there is a gap between the awareness of hidden potentials of data exploitation and their realisation.

Graph databases are said to address one of the main trends of today's businesses: leveraging highly connected data (Robinson et al., 2015, p. 1). The adaption of proven graph-based techniques, developed by pioneers from other sectors, seems to be comparatively easy and low-risk opportunity to move ahead in the competition. However, the implementation of existing graph-based solutions is a highly complex and delicate process. Graph databases are hardly deployed in industrial companies, lacking documentation of use cases and experience for a successful application (Robinson et al., 2015, p. 65; Sullivan, 2015; Kemper, 2016).

## 1.2. Objective

PSS-developing companies increasingly realise the need for exploitation of connected data. This paper aims to fill the gap between PSS development and potentials of exploitation of data generated by PSS. The main objective is to enable companies to leverage connected data generated by PSS through graph databases in order to improve present PSS systems in future generations. Consequently, a generic and flexible graph database implementation process is required that enables the efficient and effective exploitation of PSS data. The process has to provide examples and guidelines that support its implementation and specification. Figure 1 shows the function and position of the resulting graph database. Data collected during the initial product lifecycle is transformed into knowledge through graph-based technologies. This knowledge will be used in order to improve next generation products. The graph implementation process has to result in a technology capable of fulfilling this task. On the whole, the superordinate goal of the developed process is to contribute to technology transport from data driven industries and create a basis for the exploitation of PSS data.



**Figure 1. Role of graph databases within the product lifecycle**

The remainder of this paper is structured as follows: In Section 2, the domain of PSS as well as Graph databases are introduced in order to form the basis for further explanations. The research methodology is presented in Section 3, which leads to the solution described in Section 4. We evaluate the solution in Section 5 using an interview study as well as an industrial case study and conclude in Section 6 with a summary and an outlook.

## 2. Background and related work

### 2.1. PSS

Although a number of slightly different definitions of a mixed set of tangible products and intangible services exist, distinctive aspects are shared by the majority of authors (Wuest and Wellsandt, 2016). In the course of the thesis the term “PSS” will be used to refer to the following concept: A PSS is “a marketable set of products and services capable of jointly fulfilling a user’s need” (Mont, 2001, p. 2). Abramovici et al. (2016, p. 1) use the term Industrial Product Service Systems (IPSS) to describe the concept of combinations of products and services with “high customer life cycle orientation”. Physical components of PSS can be mechanical, electrical/electronics, as well as software (Schenkl et al. 2013, p. 919) The phenomenon of PSS is of equal importance for both product and service providers (Baines et al., 2009, pp. 1-2). Integrating product and service offerings has the potential to improve efficiency, which can lead to positive economic and environmental effects for industry and society (Mont, 2001, p. 4). Such improvements tend to add uptime or total-care services, which can lead to intensified use of products and timely replacement with newer, more efficient, and innovative products (Rajkumar et al., 2009, p. 36). Thereby the target is to “focus on the delivery of value to the customer and their continuous satisfaction” (Wuest and Wellsandt, 2016, p. 1).

The trend to Product Service Systems leads to an increasing determination of a product’s value by the services supporting them (Hribernik et al., 2016, p. 1). In order to design a profitable PSS, product development makes certain predictions about product manufacturing and usage. The impact of the product development phase on the whole lifecycle requires a high degree of accuracy regarding the predictive models used (Igba et al., 2015, p. 2). A common strategy to survive in the competitive environment is to involve users in a company’s business and development processes to increase knowledge about product usage (Wellsandt 2016, p. 1). An increasing range of products is capable to collect data during the use phase, as by design they are equipped with embedded sensors and software applications (Wellsandt 2016, p. 2). But not only product data collected by sensors can be used to create knowledge. Maintenance reports as well as all activities surrounding a customer interacting with the PSS can “lead to an identification of demands connected to the providers’ product (Abramovici et al., 2016, p. 1). All those data can be used in order to derive knowledge for product development in order to increase the attractiveness of a product in future product generations (Abramovici et al., 2014, p. 1).

Challenges in product development can be divided in two groups: global uncertainties and product lifecycle challenges. Development projects are exposed to a large number of uncertainties, which can be traced back to the following sources (Ulrich and Eppinger, 2012, p. 20-22): market, technical, resource, and schedule uncertainties. According to Artmann (2009, p. 12) “market and technical uncertainties are often regarded as the most decisive ones”, while budget and schedule exceedances are either “induced by the former two (...) or arise from managerial or organizational deficiencies”. Euringer (1995, pp. 2-5) stresses the crucial role of a market-orientation in product development. Market-orientation considers customers’ needs as well as competitors’ activities and technical trends (Bullinger, 1995, p. 18). Empirical studies have demonstrated the importance of a timely identification and evaluation of external trends in order to “update and revise current project targets during the development process” (Artmann, 2009, p. 12).

The field of PSS is shaped by an increasing share of services as well as the prevalence of low-cost micro sensors. Providing services that increase customers value and making products more attractive requires insights into customer behaviour and preferences. Product-embedded sensors as well as the interaction of users with remote service systems offer an opportunity for the collection of data needed to gain those insights.

### 2.2. Graph databases

Nowadays data management does not only have to deal with isolated pieces of information but with highly interconnected domains. Only database technologies that embrace relationships between data points as a core aspect of their data model can store and process large data amounts efficiently. Against this backdrop, graph databases have been targeted by companies from different industries.

Graph databases apply graph theory to store information about relationships between entities. Before discussing the particular types, purposes and benefits graph databases it is important to understand the term database itself. At its most fundamental “a database is primarily a means of organizing information” in a structured way. It allows to “interact with the data stored within it via a predetermined language, dictated by the type of database. (Kemper, 2016)

Kemper (2016) distinguishes four types of interactions: data definition, updating and manipulating existing data, retrieval and reuse of data stored in the database and administration of data that includes all the remaining actions as user management, performance analysis or security.

Graph database management systems (henceforth, graph databases) are generally optimized for use with transactional systems and thus optimized for transactional performance and availability.

A graph database “*is an online database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model*”. They are the “*equivalent of [conventional] online transactional processing (OLTP) databases in the relational world*” and typically accessed in real time from an application. (Robinson et al., 2015)

Graph databases contain explicit semantic relationships between entities. In relational databases, these relationships are not represented directly, but two entities share a common attribute value (key) instead. (Sullivan, 2015). Semantic graph data models can be based on an underlying ontology. Ontologies are considered a “specification of a conceptualization” (Gruber, 2009, p. 1). Ontologies are semantically enriched networks. Semantic networks consist of concepts, individuals, and links between them. Concepts can be organized in a specialisation hierarchy by using subclass and superclass links and given properties. In contrast, ontologies are a graphical formalism that has been created to represent word concepts in a definitional way (Baader, 2003).

Despite their name, relational databases are poor in dealing with relationships, as “JOIN” tables are required to connect tables to each other, which dramatically increases the complexity of the data model for highly connected data. Joins entail “looking up a value from one table in another table” (Sullivan, 2015).

The usage of traditional relational databases as a technology for knowledge discovery is widely spread among industrial enterprises. They are strong in handling highly structured data but, ironically, very poor in handling relations in complex, volatile data environments. Leveraging connections between data has been proved to be extremely important in competitive environment as shown by Google, Facebook, PayPal and other internet giants. It is obvious that PSS developers sooner or later must keep up in order to stay competitive. Wasting such an important resource as data may be destructive in times of Big Data. PSS developers do not have to reinvent the wheel to handle Big Data. Graph databases were created to manage highly connected data. Plenty of them are available on the market and supported by powerful graph analysis algorithms, with the flexibility, intuit, and querying speed making them popular in many areas.

### **2.3. Related work and research gap**

There are several research teams dealing with graph-based data management in mechanical engineering. Theoretical models as well as data utilisation have been developed in order to improve PSS and extend data managements’ focus on the complete lifecycle instead of concentrating only on early phases. The Laboratory for Computer-Aided Design and Production at Ecole Polytechnique Federal de Lausanne published several papers in the last eight years concerning the usage of ontologies for semantic management of product data. Furthermore, the team of Prof. Dr.-Ing. Michael Abramovici from the Chair Mechanical Engineering IT at the University of Bochum together with Susanne Dienst from the University of Siegen adapted different graph-based concepts as ontologies and Bayesian networks for feeding back usage information into product development.

The objective of our work is to enable knowledge exchange throughout and between PSS lifecycles, especially to support product developers with PSS usage data. Due to a rapidly changing environment in times of Big Data (the collection of semi-structured and unstructured data characterised by high connectivity, high volumes, and high dynamics) hamper the process of data exploitation and generation of knowledge. In addition to the challenge driven by the environment, industrial companies face internal challenges concerning data management. Lack of interest and sometimes even egoism at department

level lead to data silos. Considering that every department in a company, while contributing towards a common target – satisfying the customer – interacts and impacts all the other departments, encapsulation of information seems dangerous. Nevertheless, the desire for a collaborative data management is not enough to handle Big Data.

Existing approaches are either limited to theoretical models or focus only on narrow purposes. No backbone for the integration of graph-based techniques exists as of yet. The research area is characterised by a loose co-existence of PSS, knowledge generation approaches, graph databases and graph-based data exploitation methods in context of PSS. Thus, the integration of existing knowledge from different research fields is required to enable industrial companies to leverage connected engineering and use phase data.

### **3. Research methodology**

The goal of a research methodology is a systematic development and validation of knowledge. This work takes the Design Research Methodology (DRM) by Blessing and Chakrabarti (2009) as a basis. Blessing and Chakrabarti (2009, pp. 4-5) defined four main stages of research.

1. Research clarification (RC)
2. Descriptive study I (DS I)
3. Prescriptive study (PS)
4. Descriptive study II (DS II)

The first stage (RC) aims to clarify the research target. The presented work focusses on the knowledge discovery from PSS usage data in order to achieve PSS improvement. Within the second step (DS I) factors that influence the target and the way they influence it have to be identified. Therefore, a comprehensive research on the state of the art is required, providing a basis for the development of the targeted process. As explained earlier, this work is placed in the overlap of the trending topics PSS, Big Data and graph databases. Thus, PSS trends and challenges, data exploration and graph-based data management have to be considered. In the PS, a solution addressing the research motivation is developed. Based on requirements derived from the research objective, a solution approach is developed and implemented. The last stage (DS II) includes an application evaluation to identify whether the developed approach contributes to success considering the target defined in RC.

Blessing and Chakrabarti (2009, pp. 4-5) emphasise that “many iterations take place, and some stages may run in parallel”. Thus, they present seven different research types, depending on the depth of the research during particular stages and executed iterations in between (Blessing and Chakrabarti, 2009, p. 6). The presented work can be classified as type six, applying all stages of DRM including a review of existing challenges and solution approaches followed by a detailed PS and comprehensive DS II. In this case, the developed graph database implementation process will be evaluated through expert interviews and a case study, performing iterations between stages three (PS) and four (DS II), using the gained insight to improve the process.

### **4. Solution approach**

Comparing the Knowledge Discovery Processes from Fayyad et al. (1996), Anand and Buchner (1998), Cios et al. (2005), Cabena et al. (1998) and Shearer (2000), a basic common structure can be identified:

1. Understanding problem domain and initial situation
2. Setting target and derivation of procedure
3. Preparation and modelling of data
4. Generation of knowledge out of data
5. Evaluation and usage of knowledge (not considered as beyond of project focus)

Applying the process to graph-based data management requires assistive tools and models for each step. The work aims to develop a detailed process description and required process elements. Any assistive element that is developed and added to the graph implementation process (GIP) will henceforth be called knowledge graph element (KGE). Figure 2 presents the basic knowledge discovery process, refined with respect to the utilised technology (graph databases) and provided KGE's. The subsequent Table 1

explains the individual KGEs assigned to the process phases and Figure 3 highlights two particular KGEs: The underlying and adaptable meta model as well as the time tree-based mechanism to store PSS sensor data.

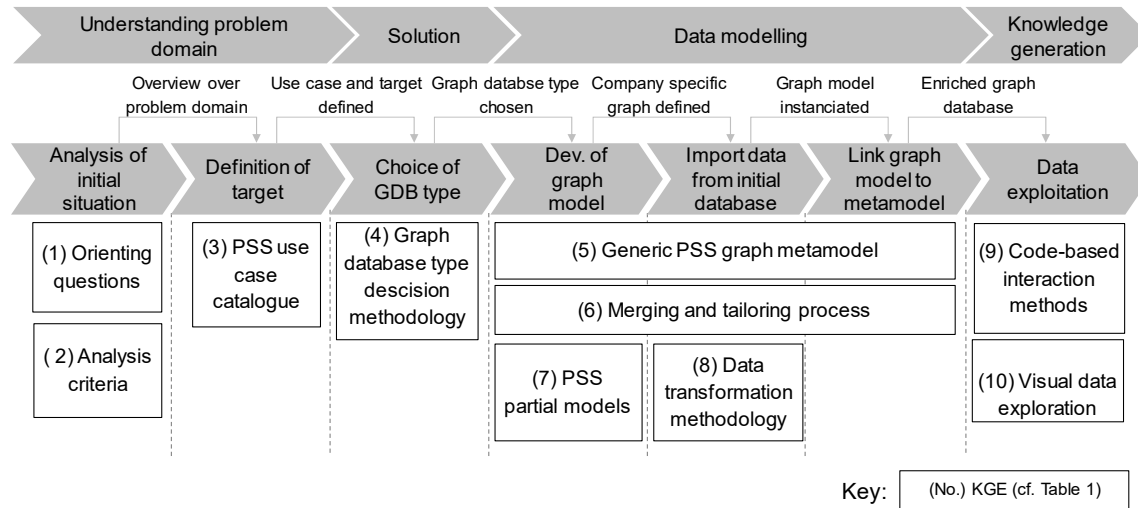


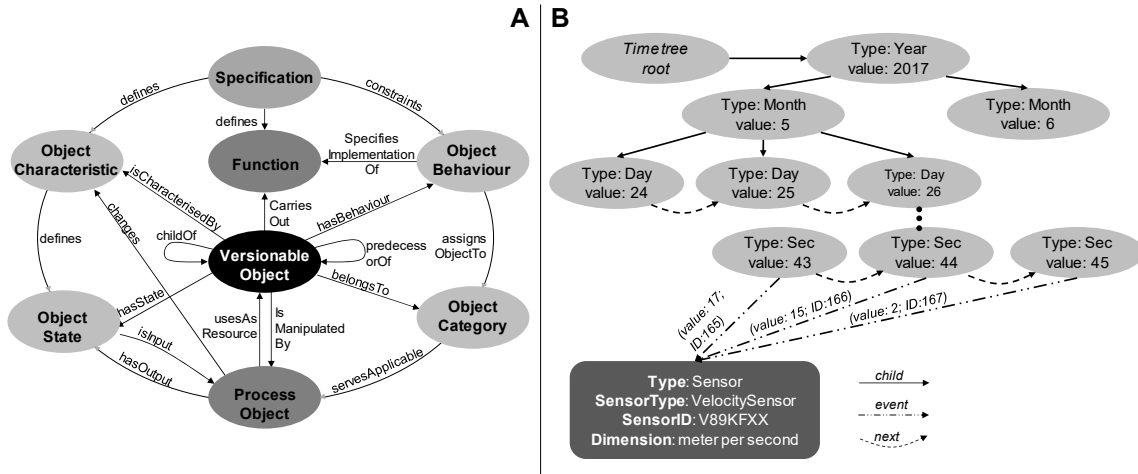
Figure 2. Graph implementation process (GIP)

Table 1. Knowledge graph elements

KGE 1	The understanding of the problem domain requires information from the knowledge carriers within the company. Therefore, 14 <i>orienting questions</i> have been defined as a starting point to analyse the underlying situation. The investigation focus is on the challenges related to data acquisition, data management and data exploitation.
KGE 2	The second knowledge element aims to <i>structure gathered information and evaluate it based on five analysis criteria</i> . Basically, KGE 2 supports the investigation process in order to detect, (a) whether a graph database is required at all and (b) whether the circumstances are favourable for their implementation.
KGE 3	The <i>use case catalogue</i> is an orientation tool that supports the goal setting process and aims to lighten the switch from RDBMS to GDB. It provides enterprises with application examples of graph-based knowledge exploration. Every application example represents a potential use case for GDB. 40 use cases have been defined and assigned to one of the following clusters: Interdisciplinary search, Impact analysis, Cluster analysis, Pattern search, Natural language processing (NLP), Pattern search, Reasoning.
KGE 4	The choice of a graph database is an important step towards the target of the knowledge discovery process. This knowledge element considers strengths and weaknesses of both, Property Graphs and RDF Triple stores and <i>guides the decision process based on pre-defined criteria</i> . The first criterion relates to the required modelling precision and the outcoming complexity of the graph model. Subsequently, the user has to evaluate the importance of automatic knowledge deduction within the database. The importance of performance and usability for pre-defined use-cases are the third and fourth areas of investigation.
KGE 5	KGE 5 describes a <i>meta model</i> , which allows to map every enterprise data point related to PSS and can be used as a meta structure for company-tailored graph models. It is generic enough to fit all areas of PSS – independent of the share of services in the system or the department within the company. Nevertheless, it is also flexible enough to be adapted to the user’s goals (s. Figure 3 A).
KGE 6	Various graph models have been introduced in the literature to represent areas of PSS systems. Most of the models pursue a specific target and do not meet the requirement of generic abstraction. Others serve the hierarchical part of ontologies and demonstrate the inheritance of characteristics rather the focus on relationships. KGE 6 is a partial model catalogue that supports the modelling

	<p>process by providing modelling approaches for specific business areas. Three models have been developed in the course of our project, as the state of the art in research provided no support: Versioning of objects in property graphs, Representation of variants, Acquisition and storage of sensor data via time trees (cf. Figure 3 B)</p>
KGE 7	<p>The creation of a company or use case specific graph-based data model is guided by the three-step <i>merging and tailoring process</i> (KGE 7). The tailoring process starts with the development of a product model. Information collected in the first two stages of the knowledge discovery process is used to abstract required data to concepts. Industry-specific and company-internal terms and connections are implemented. To lighten the effort of developing own models, with respect to the efficiency of graph querying and modelling complexity, or granulating existing ones – the following rules have been defined as guidance:</p> <ul style="list-style-type: none"> <li>- If the range of an object property is unlimited (e.g. a number) it should be modelled as a characteristic with a datatype definition that restricts the type of the range.</li> <li>- If the range of an object property is limited but not important for pattern search (e.g. colour) it should be modelled as a characteristic with a range limited by a list of properties.</li> <li>- If the range of an object property is limited and important for pattern search (e.g. car type) it should be modelled as a node with the class type equal to the properties collective term.</li> </ul> <p>In the second step data points related to the underlying use case are assigned to objects within the product model and thus inherit the frameworks of the concepts. The assertion of data points to ontology objects enables data transformation from existing data bases to the GDB.</p> <p>In step three, the product model, including asserted instances from real data, is connected to the PSS graph metamodel. This step is required to check the plausibility and consistency of the created product model. In fact, the metamodel is designed to guide the user through the first two steps.</p>
KGE 8	<p>Data transformation and deployment in GDB has high potential for inefficiencies and mistakes. We consider <i>three stages of data transformation from RDBMS to GDB</i>: data preparation, data extraction from RDBMS and data import. KGE 8 provides a strategy for each step, in order to achieve an efficient and transparent data transformation process.</p>
KGE 9	<p>KGE 9 refers to <i>code-based data exploitation</i>. It describes and serves examples for graph querying, reasoning, usage of existing open source graph algorithms and demonstrates the creation and usage of user-defined algorithms.</p>
KGE10	<p>The access to information and knowledge is the key factor for a successful business. Availability of knowledge in product development is crucial for the creation of competitive products. However, the need for skills to query databases hamper the access to information. Furthermore, analysis tasks are often communicated to the IT department and after a period of time results are sent back to the requestor. GDB serve an approach to overcome the mentioned challenges and to enable knowledge generation for unskilled data analysis: <i>visual data exploration</i>. Cyber data exploration includes all techniques related to exploration of rendered graph data through interface and is demonstrated in KGE 10.</p>

The operative execution of the GIP is supported by the GIP Use case Map (GUM) (Figure 5). In the first stage, the GUM can be used to gain inspiration by use cases of GDB in the field of PSS. Furthermore, the analysis of the initial situation, which requires the investigation of available data and its quality is assisted by the “Data Types” column which presents data types that will be required in the most variations of a particular use case. The choice of an implementation solution is supported by four decision criteria resulting in a recommendation whether the use case supposes the deployment of a property graph or a RDF Triple Store. Existing graph models are assigned to use cases which might make use of them in case of more complex modelling requirements. The list part of GUM presents recommended graph data exploitation methods with respect to the chosen use cases. In the event that more than one use case has been picked, the usage of GUM has to be carried out with the awareness of the necessity of making trade-offs, as done in the case study.



**Figure 3. (A) Underlying generic metamodel of the graph database (B) Example for sensor data storage within a Property Graph database based on a time tree and weighted relationships**

	Understanding problem domain			Solution			Data modelling	Knowledge generation				
	Use cases			Data types			Graph databases	Graph models	Data exploitation methods			
	Cluster	Title	Description	Product data	User data	...	Importance precision & complexity	Importance knowledge deduction	...	Graph querying	Usage of existing graph algorithms	...
Interdisciplinary search	Use case list			Required data to execute the use case from the product lifecycle			Evaluation of GDB with respect to each use case	Graph models for modelling enterprise data depending on chosen use case		Recommended graph data exploitation methods considering chosen use case		
Pattern search												
...												
Visual data exploration												

**Figure 4. GIP use case map (GUM) (abbreviated)**

## 5. Evaluation and discussion

### 5.1. Expert assessment

For purposes of validating and extending the knowledge initially gained from the theoretical foundations, a qualitative interview study has been conducted (Collis and Hussey 2014). For the collection of primary research data, 20 individuals have been defined. The sample contains large, medium, and small companies as well as GDB affine consulting agencies in order to question experts as well as prospective applicants. Figure 5 demonstrates the sample by mapping every individual according to the company size, the level of overview (based on hierarchy level) of the individual within the company, the company type, and the degree of interest in GDB for the deployment within the company.

During the interview, 65% expressed concerns related to human resources with respect to the execution of GIP and usage of GDB. Another insight concerning the first stage of the GIP is that data access is strongly dependent of the business model: 62.5% of companies with a business-to-business (B2B) model (~78% of which are suppliers) replied that they do not have access to important data. Only 14.3% of companies operating a business-to-customer (B2C) model (~88% of which are OEM's) complained about data access. Nevertheless, the issue of adequate data quality was mentioned multiple times in both cases.



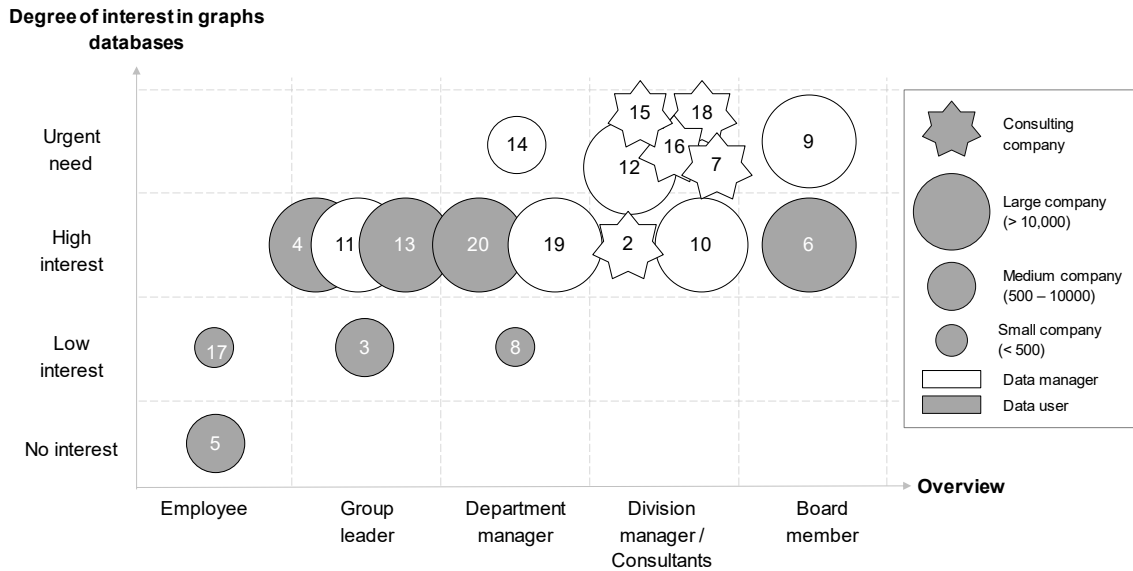


Figure 5. Categorisation of sample population

## 5.2. Case study results

The GIP was executed at eGym, a small to medium-sized German developer and provider of networked fitness machines and associated services. Based on orienting questions from KGE1, the underlying product data management systems have been analysed and evaluated regarding the necessity of implementation of graph databases according to criteria defined in KGE2. The main objectives are to generate insights into user behaviour to be able to provide better services and to understand failure causes of machines to enable predictive maintenance. Both targets can be lead back to three use cases based on algorithmic and visual pattern search form KGE3 contained in GUM: (1) Find customer behaviour patterns, (2) failure cause identification, and (3) visual time pattern search.

The underlying data is highly connected with a low degree of relationship complexity. Thus, the low precision of property graph modelling was considered negligible compared to the performance benefits of property graphs over RDF Triple Stores. The amount of open source visualisation tools for property graphs encouraged the team to choose Neo4j in combination with Graphileon InterActor, a Neo4j graph database interaction tool. Based on the KGEs 5 to 8, the graph model was developed. The high share of "Event"-type objects required the usage of the time tree model for efficient storage of training data acquired from fitness machines.

The user behaviour analysis was based on the assignment of users to groups depending on their membership duration at one of eGym's partner gyms. Behaviour patterns of successfully engaged users were compared to patterns of early quitters. Pattern search for the machine failure identification use case was performed by visual investigation in addition to querying.

In general, findings made during the case study at eGym were separated in three groups: new, confirmed, and potential knowledge. *New knowledge* demarcates insights from data that analysts have not been aware of before. It does not necessarily mean that previously utilised databases are not capable of generating this knowledge but that the process of generating it required a hypothesis to test. The following insights were classified as new knowledge:

- A discovered dependency between goals set by successfully engaged users, the involvement of a trainer, and different training event types. Goals crucially influence the preferred training types of users as well as the necessity for support from a fitness trainer. As a next step, analysts may investigate the order of events regarding their type, to specifically assign trainers to activities.
- Machine issues could be traced back to the following factors: machine type, motor type, motor frequency, and type of events carried out on the machine one month before the issue.

*Confirmed knowledge* created through the implemented GDB considers knowledge that has been previously discovered by analysts using SQL-based analysis in the first place. The same findings were made by using hypothesis-free pattern search and allowed to compare both approaches. Hypothesis-free pattern search allows an impartial user to come to similar conclusions as experience analysts with a high degree of understanding the underlying data structures. Insights related to machine quality issues have been created and compared to existing knowledge.

The execution of the GIP improved eGym's understanding of GDB technologies and opportunities, enabling a targeted realisation of their capabilities and formulation of new analysis ideas. Ideas assessed as promising potentials for future knowledge discovery is understood as *potential knowledge*. eGym agreed to follow up on three use cases:

- A high share of eGym's activities is related to services around their fitness devices. Maintenance services, training support through trainers, and the creation of fitness plans are part of daily business. The implemented GDB provides an interface for the integration of recommendation engines. eGym can use recommendation algorithms in order to suggest services to users based on their training habits and goals.
- Software for visual interaction with graph data was suggested to be used directly by product developers and designers, to increase overview and understanding of available data. The hypothesis being that the communication of data structures can evolve product development.
- Machine issues are currently documented in the form of continuous text by service technicians. Additionally, issues are classified by service technicians and subsequently the maintenance team. However, the categories are too generic and make statistical cause analysis less accurate. Graph-based Natural Language Processing could be used to analyse all issue documentations in order to automatically reclassify and implement them into the ontology model.

The implemented graph database has given eGym' analysts a powerful tool for data analysis. The pilot project, which represents the case study of this thesis, demonstrated not only the advantages of the GDB as a knowledge discovery tool but also the opportunities in usage of GDB as an interface to existing data analysis technologies.

### 5.3. Strengths and limitations

The theoretical basis of the GIP was substantiated by performing a critical examination within a qualitative interview study as well as practical testing within an industrial case study. The GIP enables the application of graph-based knowledge discovery in data generated during PSS usage. As a result, industrial companies are given access to a new field of data management, with the GIP serving as a vehicle to transfer existing knowledge. Subsequently, technologies that have been developed and advanced by internet giants, telecommunication, and healthcare companies become available for mechanical engineering enterprises. The GIP is furthermore supported by the GIP Use Case Map (GUM). It extends the GIP from a general graph database implementation process to a guided and structured methodology. It guides and lightens the effort involved in the execution of implied tasks and thus allows unexperienced companies, with respect to GDB, to estimate results and potentials before and during the project. For that, the GUM provides a collection of concrete use cases as input. The critical review of the process and its potentials confirmed a strong interest among industrial companies and potentials for utilizing the benefits of graph-based technologies. However, it is important to notice that GIP as of now focuses on basic database CRUD operations (create, read, update, delete). More complex computations (i.e. for time-related patterns) should be performed with the additional aid of specific graph computation engines.

## 6. Conclusion: Summary, contribution and outlook

The business environment is increasingly dependent on knowledge besides physical resources. PSS companies realise that sustaining a competitive advantage in global markets requires the maximum utilisation of generated usage-related PSS data. The intelligent use of data about the PSS and its users enables companies to recognise customer's present as well as future needs. Graph databases were proven by internet, telecommunication and healthcare industries to provide ways to exploit usage data and

create knowledge out of it. The objective of this work was to enable the transport of existing graph-based technologies to the field of PSS. The adaptation of technologies to a new area requires a systematic and guided approach. Latter was developed based on existing research on PSS, knowledge management and graph databases. This graph implementation process (GIP) was derived from a generic knowledge discovery process (KDP) and adapted to graph databases as the underlying technology. The initial GIP was reviewed within a qualitative study through expert interviews. Both studies lead to an enrichment of the initial process by an additional supportive element. The final GIP contains four stages of graph-based knowledge discovery. Each stage of the process is supported by specific guidelines and examples contained in the GUM.

The GIP not only addresses the needs of industrial companies that struggle with highly connected PSS data, but also has scientific value. It combines separated research fields by integrating existing graph-based technologies into the generic knowledge discovery process and adapting it to PSS. GIP can thus be regarded as an instantiation of the knowledge discovery process with a specific purpose. Further research could use the GIP as a reference process, assign new technologies and concepts to its stages and extend the GIP with new supporting methods. The GIP has been developed for and evaluated in a PSS context, however, it should be transferrable to purely product-based offerings.

The ability to analyse and understand huge amounts of highly connected data is crucial in determining companies that will outperform their competitors. GIP not only provides enterprises the ability to use a wide range of graph-based technologies but also offers links for further research.

The GIP fills the gap between existing graph technologies and the knowledge management in industrial companies. It enables enterprises to make use not only of existing domain descriptions in ontologies but also of open-source algorithms, developed for the analysis of highly connected data. Based on research results of this work, the following research perspectives have been derived: Performance improvement of the resulting analysis through the integration of computation engines, intelligence through semantic enrichment of graphs, application augmentation, as well as further empirical validation of the GIP and GUM through quantitative studies as well as extended case studies.

## Acknowledgment

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research centre ‘Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service systems based on technical products’.

## References

- Abramovici, M., Dang, H.B, Göbel, J.C. and Savarino, P. (2016), “Systematization of IPS2 Diversification Potentials Using Product Lifecycle Data”, *Procedia CIRP*, Vol. 47, pp. 288–293. <https://doi.org/10.1016/j.procir.2016.03.029>
- Abramovici, M., Dienst, S., Fathi, M. and Lindner, A. (2014), “Development of a knowledge-based feedback assistance system of product use information for product improvement”, *International Journal of Product Development (IJPD)*, Vol. 19 No. 4, pp. 191. <https://doi.org/10.1504/IJPD.2014.063027>
- Abramovici, M., Lindner, A., Fathi, M. and Dienst A. (2011), “A Conceptual Data Management Model of a Feedback Assistance System to support Product Improvement”, *IEEE International Conference on Systems, Man, and Cybernetics (SMC), Anchorage, Alaska, USA, October, 9-12, 2011*.
- Artmann, C. (2009), *The value of information updating in new product development*, Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-93833-0>
- Baader, F. (2003), *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge university press.
- Barc (2014), *BARC Big Data Analytics Survey 2014*. [online] Available at: <http://barc-research.com/wp-content/uploads/2014/06/BARC-big-data-analytics-2014-EN.pdf>
- Blessing, L. and Chakrabarti, A. (2002), *DRM: A Design Research Methodology*, Springer, London. [https://doi.org/10.1007/978-1-84882-587-1\\_2](https://doi.org/10.1007/978-1-84882-587-1_2)
- Bullinger, H.-J. (1995), *Integrierte Produktentwicklung. Zehn erfolgreiche Praxisbeispiele*, Gabler, Wiesbaden. <https://doi.org/10.1007/978-3-322-82712-8>
- Ehrlenspiel, K. (2014), *Kostengünstig entwickeln und konstruieren. Kostenmanagement bei der integrierten Produktentwicklungen*, 7th ed., Springer, Berlin. <https://doi.org/10.1007/978-3-642-41959-1>

- Euringer, C. (1995), *Marktorientierte Produktentwicklung, Die Interaktion zwischen F & E und Marketing. Gabler edition Wissenschaft*, Deutscher Universitätsverlag, Wiesbaden. <https://doi.org/10.1007/978-3-322-99331-1>
- Gruber, T. (2009), *Ontology*. [online] Available at: <http://tomgruber.org/writing/ontology-definition-2007.htm>
- Hribernik, K., Kiritsis, D., van der Vegte, W. and Thoben, K.-D. (2016), "Towards a Methodology for Selecting Product Usage Information Sources for the (Re-)Design of Product Service Systems", *ICE/IEEE International Technology Management Conference, Trondheim, Norway*.
- Igba, J., Alemzadeh, K., Gibbons, P.M. and Henningsen, K. (2015), "A framework for optimising product performance through feedback and reuse of in-service experience", *Robotics and Computer-Integrated Manufacturing*, Vol. 36, pp. 2–12. <https://doi.org/10.1016/j.rcim.2014.12.004>
- Kemper, C. (2015), *Beginning Neo4j*, Apress, Berkeley, CA. <https://doi.org/10.1007/978-1-4842-1227-1>
- Marzi, M. (2012), "Introduction to Graph Databases. Quick look at trends in data, nosql, graph databases, and Neo4j", *Chicago Graph Database Meet-Up, April 29, 2012*.
- Mont, O.K. (2001), "Clarifying the concept of product-service system", *Journal of Cleaner Production*, Vol. 10 No. 3, pp. 237–245. [https://doi.org/10.1016/S0959-6526\(01\)00039-7](https://doi.org/10.1016/S0959-6526(01)00039-7)
- Rajkumar, R., Sundin, E., Lindahl, M. and Ijomah, W. (2009), "Product design for product/service systems", *Journal of Manufacturing Technology Management*, Vol. 20 No. 5, pp. 723–753. <https://doi.org/10.1108/17410380910961073>
- Reiner, D. (2004), *Strategisches Wissensmanagement in der Produktentwicklung, Methoden und Prozesse für kleine und mittlere Unternehmen*, 1st ed., Deutscher Universitätsverlag, Wiesbaden. <https://doi.org/10.1007/978-3-322-81135-6>
- Robinson, I., Webber, J. and Eifrem, E. (2015), *Graph databases*, 2nd ed., O'Reilly Media Inc.
- Schenkl, S.A., Behncke, F.G.H., Hepperle, C., Langer, S. and Lindemann, U. (2013), "Managing Cycles of Innovation Processes of Product-Service Systems", *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics. Manchester, United Kingdom, October 13-16, 2013*. <https://doi.org/10.1109/SMC.2013.161>
- Sullivan, D. (2015), *NoSQL for mere mortals*, Pearson Education, Upper Saddle River, NJ.
- Ulrich, K.T. and Eppinger, S.D. (2012), *Product design and development*, 5th ed., McGraw-Hill Irwin, New York, NY.
- Wuest, T. and Wellsandt, S. (2016), "Design and Development of Product Service Systems (PSS) - Impact on Product Lifecycle Perspective", *Procedia Technology*, Vol. 26, pp. 152–161. <https://doi.org/10.1016/j.protey.2016.08.021>

Christoph Hollauer, Dipl.-Ing., Wissenschaftlicher Mitarbeiter  
 Technical University of Munich, Institute of Product Development  
 Boltzmannstraße 15, 85748 Garching bei München, Germany  
 Email: christoph.hollauer@pe.mw.tum.de