A Graph-Based Knowledge Model for Value-Oriented PLM Implementation

Bas Koomen[1,2](✉), Marcel de Klein[2], and Mayke Vergouwen[3]

[1] University of Twente, Enschede, The Netherlands
s.p.koomen@utwente.nl
[2] Cadmes B.V., 's-Hertogenbosch, The Netherlands
{bas,marcel}@cadmes.com
[3] Avans University of Applied Science, Tilburg, The Netherlands

Abstract. Product Lifecycle Management (PLM) has matured over the last two decades. The software and the market have evolved, resulting in a small number of dominant vendors offering an array of functionality in their software platforms. Moreover, PLM software is becoming available in the cloud as a service (Software-as-a-Service, SaaS). These developments impact the Value Added Resellers (VARs) who deliver software and services to the users of the PLM software. The VARs need to focus more on their added value by providing contextual application knowledge for PLM software in specific business environments. The management of this knowledge is a challenge for VARs because this relies strongly on tacit, empirical knowledge that resides in the heads of their staff. This paper describes a methodology that uses Benefit Dependency Networks to capture, manage, and reuse this knowledge. The methodology is based on the literature and then modified with insights from case studies with a PLM VAR. All relevant contextual elements in an extended Benefit Dependency Network of a PLM implementation are registered in a Graph Model, allowing consultants to find answers within a given industry context and configure a customer-specific implementation plan. The paper also describes a software design built on a Graph Database to use this knowledge in an operational VAR environment and build a foundation for future machine learning in implementation knowledge.

Keywords: PLM · Product Lifecycle Management · Graph model · Knowledge management · Implementation

1 Introduction

The concept of *Product Lifecycle Management* (PLM) has been around since the end of the 20th century. PLM is primarily a *business approach* to manage a product's lifecycle from conception to recycling [1–3]. Nevertheless does the term PLM very often stands for software. Indeed, PLM is mainly aided by *PLM software* consisting of various domains, i.e., Computer-Aided Design (CAD), Product Data Management (PDM), Computer-Aided Manufacturing (CAM), Finite Element Analysis (FEA), Project Management, Product Management, Requirements Management, Service Management, and many others. There are definitions of PLM that also include Customer Relations

Published by Springer Nature Switzerland AG 2022
O. Canciglieri Junior et al. (Eds.): PLM 2021, IFIP AICT 640, pp. 225–239, 2022.
https://doi.org/10.1007/978-3-030-94399-8_17

Management (CRM) and Enterprise Resource Planning (ERP) [4]. PLM (and ERP) software has allowed companies to develop, produce, and maintain more complex products at lower cost, less time, and higher quality [5]. Nowadays, the creation of industrialized products is hard to imagine without using some combination of PLM and ERP software.

In the early ‘90s of the 20th century, the PLM software market knew many software vendors, represented by small and specialized Value-Added Resellers (VAR). A VAR is a reseller of a vendor’s software who delivers services in addition to the software, i.e. training, consultancy, support, customizations. To serve their customers with automation, these VARs could deliver various software and computer hardware products coming from different vendors.

Today, a few prominent vendors lead the PLM market. They have grown by the acquisition of many other vendors [6]. A vendor can acquire a different vendor in a completely different market, for example, the acquisition of healthcare software vendor Mididata by Dassault Systèmes, a typical vendor in the product design and manufacturing industry [7]. Also, a vendor may acquire a vendor in the same market, but with a complementary discipline, for example, Dassault Systèmes’ acquisition of ERP vendor IQMS [8]. The result is that the vendors’ portfolios have become very large, diverse, and complex. Most large vendors have multiple CAD solutions, analysis options, and data management flavors.

This trend also impacts the VARs who are required to grow with the expanding portfolio of the vendors. To support the diverse portfolio, the VARs need to build expertise in multiple domains to satisfy the goals in their reseller contracts. Since this expertise can be very specialized, the VAR needs to grow above a critical size. The result is that VARs grow by acquiring smaller VARs, and this smaller number of large VARs dominate the market [9].

The introduction of Software-as-a-Service in PLM poses another challenge to the VARs. The VARs’ traditional role as a go-between for customers to obtain PLM software slowly disappears as customers can access software directly on the vendor’s cloud platform [10]. When the vendors and the customers are less dependent on VARs for the software, the VARs have to deliver other value to their customers to stay profitable.

Besides their software-related services (installation, training, and support), the VARs can assist companies in the optimization and transformation of business processes. This way, they respond to technological developments and changing market demands [11], for example, in the context of the Fourth Industrial Revolution (Industrie 4.0), being a critical change driver to transform product and process development [12]. Suppose a VAR specializes in a specific industry segment. In that case, it can build knowledge on how companies can adapt their processes to the capabilities like multidisciplinary product development, use of IoT technology, additive manufacturing, or product configuration. The VAR can make the connection between the market demands and the application of the PLM software. Because VARs execute multiple PLM projects, their knowledge can grow faster than the knowledge within individual industrial companies, especially if these companies are small or medium-sized. However, the development of knowledge is a challenge for VARs for several reasons:

First, the knowledge is captured during the execution of projects and becomes primarily tacit knowledge in the heads of individual consultants. After participation in multiple projects, consultants become senior enough to manage a project themselves. This tacit knowledge becomes the main asset of the VAR but not in a reusable way. The value that the VAR delivers using this asset is discovering a (business) problem and providing a solution for the customer. In this context, the VAR's business fits the definition of a Value Shop [13].

Second, the knowledge crosses multiple discipline domains. To discover the underlying business problem, the consultant must be familiar with specific aspects, challenges, and trends in the customer's industry. Also, he/she must know how processes can be analyzed, redesigned, and implemented in an organization. Additionally, an information system architecture must emerge to assist the execution of those processes. Moreover, the consultant must know the PLM software capabilities and how they should be applied before he/she can propose the best practices within a given software context. Finally, the consultant must define a business case with the desired outcome and a project plan with a budget, timeline, and quality goal. All this can become even more complicated due to specialized subjects within each discipline domain (i.e., CAD, CAM, FEA, PDM).

Literature on PLM implementation exists and describes various guidelines on implementing PLM (changing from a current state to a future state where a company applies PLM processes with the help of PLM software in a PLM organization). A literature review about PLM implementation in SMEs [14] has learned that most guidelines focus on the project approach from a customer's perspective. A large part of the process is before the selection of a software solution. The effect is that future state processes are defined without the context of commercial software, resulting in gaps between software capability and desired processes [15]. There is a knowledge gap on methodology for PLM implementation from the perspective of a VAR.

This paper describes the progress of a developing methodology to reuse implementation knowledge in a modular way [16], ensuring that the delivered PLM implementation has value to a customer (the value in this context is defined as the contribution to a measurable improvement in the customer's processes and organization). This methodology is based on Benefit Dependency Networks [17] and then evaluated in the operational environment of a VAR in explorative case studies. With the results from the case studies, we enhanced the methodology with additional contextual elements described in Sect. 2 of this paper. Section 3 presents the resulting conceptual model in a graph representation to capture the contextual knowledge. In Sect. 4, we explain our approach to use this model in PLIKSS, a software-based solution that will be developed based on this research. A VAR can use this software to capture and reuse PLM implementation knowledge. We present our conclusions in Sect. 5, where we also discuss the next steps.

2 Contextual Knowledge Elements of PLM Implementation

This section describes the research that led to the formulation of our conceptual model. The analysis uses case studies within the operational environment of a VAR that represents Dassault Systèmes. The majority of this VAR's customers are small and medium-sized enterprises (SME) in the machine-building industry.

2.1 Benefit Dependency Networks in PLM

Benefit Dependency Networks (BDN) [17, 18] can describe the relations between business value, process, and IT (Fig. 1).

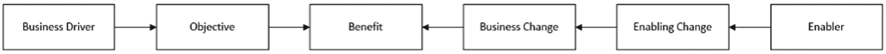

Fig. 1. The essential elements of an IT Benefit Dependency Network.

The PLM Vendor Dassault Systèmes and their VARs use these BDNs (value models) to build a high-level business case for specific customers [19]. Consultancy teams of Dassault Systèmes map customer-specific business drivers via objectives, benefits, business changes, and enabling changes to the enabling software packages. Dassault Systèmes also provides typical BDNs for industry segments, where business drivers, common to that industry segment, are linked to specific goals, benefits, etc. At the other end of the BDN, the elements connect to a small number of standard software solutions. Dassault Systèmes has identified 11 different main industry sectors with 56 sub-segments in total [20]. Figure 2 shows a diagram of the variation of these elements across the 7 sub-segments in the "Industrial Equipment" [21] industry sector. The variation is compared with the commonality (in how many sub-segments this element appears). We observe frequent reuse of the enablers and that the goals are specific to each industry sub-segment.

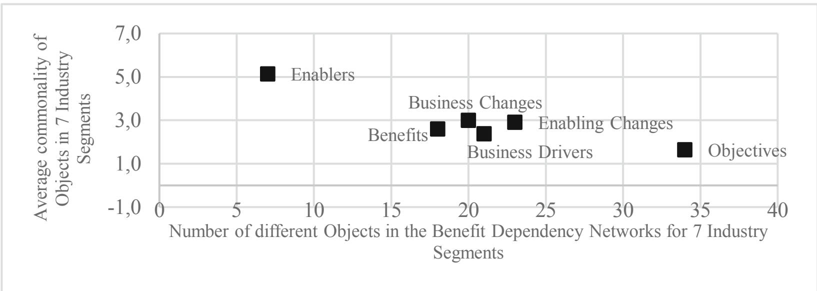

Fig. 2. Variation vs. commonality of BDN elements in industry-specific value models.

Although these value models are beneficial during the high-level definition of a PLM solution with the corresponding business justification and project scope, they lack detail to provide an operational implementation approach. The enablers consist of a variable function set in the software that needs a high level of expertise. Also, the enabling changes and business changes are on a too abstract level for practical use for implementation.

To measure the benefit dependency awareness of practitioners, we have conducted several analyses. 1) A survey with five project managers who are responsible for PLM implementation projects [22], 2) a review of 94 PLM implementation proposals, 3) a survey with 10 PLM consultants who have between 3 and 20 years of experience in PLM implementation, and 4) interviews with seven different functions in the VAR's value chain. The results indicate that the consultants of the VAR are focused on the functional implementation of the software. We also observed that they are primarily aware of operational benefits directly related to the use of the software, i.e., company-wide access to product information, reduced errors, increased efficiency. However, they are less aware of the impact on strategic business aspects and the underlying motivation for a company to change its PLM processes. What also stood out in the results was the lack of quantified benefits. Only 4 out of 94 project proposals contained quantified benefits in their project goals. From the VAR's value chain interviews, we learned a lack of knowledge between functions (departments) in the organization, especially about (new) software capabilities. Moreover, there is a lack of information transfer in the hand-over between functions when a project is sold. (Transfer from sales to planning, from planning to consultants, and from consultants to support). This hand-over leads to mistakes and causes rework in implementations, delays, and increased costs. Onboarding and training new employees is another challenge that was apparent in the same interviews.

2.2 The VAR-Tailored BDN

We have modified the BDN to make it more suitable to reuse operational implementation knowledge in the proposed methodology that we described in our previous paper [16]. We also changed the process elements from a *delta* formulation (i.e., *business change*) to a *target* formulation. (i.e., *business process*). We implemented this structure in a wiki system where we documented and linked typical BDN elements. This wiki resulted in a modular knowledge base with operational implementation knowledge related to benefits, goals, and business drivers.

After submitting the paper that describes this method, we continued our work on the case studies. We collected more insight into the efficacy of the methodology, resulting in several improvements.

2.3 Redefined BDN Elements for Process and Enablers

Based on an exploratory case study preceding this case study, we initially simplified the original elements *business change*, *enabling change*, and *enabler* into *use-case* and *enabling function*. The idea was that this would lower the effort to capture typical processes and link them to the software. However, when the consultants were adding more different processes, it became more complex to extract common practices and reuse knowledge. Therefore, we proposed a new structure, shown in Fig. 3.

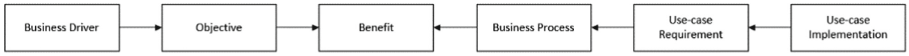

Fig. 3. BDN elements, modified for reusable PLM implementation knowledge.

For the content in the category *business process,* we use 26 different PLM reference processes (i.e., *change management*, *product management*, *document management*) as defined by VDMA [23]. To match the operational detail level of a VAR, we divided the process for integration with authoring applications (*CAx integration*) into specialized processes, i.e., *3D CAD product design*, *3D CAD product engineering*, *CAM programming*, *Finite Element Analysis*. A business process includes one or more *use-case requirements* to define the process at a particular maturity level.

The *use-case requirement* element represents an operational need in a process, for example, "request a change" in change management. A use-case requirement should be described almost as a black box with a desired input and output. It also should not contain any reference to specific software or IT data type. For example, it should reference a "product component" instead of a "CAD assembly". The number of use-case requirements needed in a business process varies per process and maturity level. The lower maturity levels need fewer use-case requirements than the higher levels.

The *use-case implementation* describes the method to realize the use-case requirement to reference specific software and data types. A use-case requirement can have multiple use-case implementations. For example, in our case study, we found eight different use-case implementations for "request change" in the same software. Another possibility is to reference use-case implementation from another use-case implementation. Small use cases are used in many larger use cases, for example, a search function.

The methodology has a set-based knowledge management approach by providing multiple solutions for a single use-case requirement [24]. The selection of a specific use-case implementation depends on priorities in the customer's context. By evaluating the trade-off characteristics of the available use-case implementations, a short-list can evolve. The VAR then evaluates the short-list hand-on with the stakeholders of the customer's organization and selects the use-case implementation that contributes most to the goals.

2.4 Relation Between Use-Cases and Software

We instructed the consultants to keep the information per BDN element compact. The information should fit on one DIN A3-sized page. We noticed that many wiki pages contained the same information about the software in the use-case implementations. Also, selecting one software option in a use case is relevant for the software selection in another. The harmonization of software options has an impact on software investment.

In our case study, we focussed on Dassault Systèmes' 3DEXPERIENCE Platform. This platform contains a structure of *roles* (licenses). Each *role* can include one or more *apps*. In the public cloud version of 3DEXPERIENCE, 525 roles share 390 different apps [25]. Use-case implementations use the apps and, therefore, can be linked. Also, there needs to be a link between apps and roles to get information about the required software licensing.

2.5 Ontology and Organisational Elements

The consultants also pointed out that a reference to a data model was missing in the structure. This feedback made us look deeper into the nature of a use case. A use-case has subjects (actors) and objects (items used or transformed in the use-case). We realized that both the use-case requirements and the use-case implementation refer to objects. Therefore, we came up with a dual ontology. On the one side, there are natural objects, independent from software, like "technical specification", "component", or "customer complaint". On the other side, there are software-specific objects that can represent these natural objects, like "digital 2D drawing", "3D CAD part", or "issue". These ontology objects have a variety of link types, for example, hierarchical links (*part – assembly*), representation links (*component – 3D CAD part*), or informational links (*drawing – <specifies> – component*). The representation links can be one-to-one, one-to-many, or many-to-one. It turned out that this addition was beneficial for the VAR to build application know-how on the large variety of data types in the software (3DEXPERIENCE contains 235 different data types).

On the subject part of the use cases, we propose to add typical organizational elements that act in the targeted industry segments. We decided to add the elements *department* and *(organizational) role*. The roles can be referenced by use case requirements but also by benefits. For benefits, it is relevant who is responsible and who is impacted. Roles can be related to departments and to other roles to describe organizational hierarchy.

2.6 Industry Context

We also involved sales consultants and marketing staff working with the business drivers, goals, and benefits during our case study. In general, they could work with the structure but suggested adding industry segments to filter business drivers relevant to the customer's context. Also, knowledge of the industry is essential to formulate typical *business drivers*, *goals,* and *benefits*. We added the seven sub-segments of Industrial Equipment as mentioned in Sect. 2.1 to start with, but the VAR's marketing department will make these more specific with their market research.

2.7 Extended PLM Benefit Dependency Network

Figure 4 shows the resulting dependency network after adding the new elements. With dotted lines, we indicated the different knowledge domains that come into play with a PLM implementation.

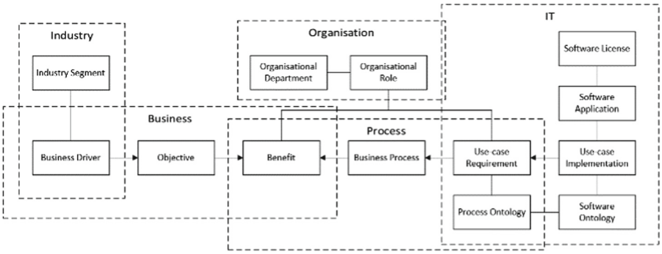

Fig. 4. Extended version of the PLM dependency network.

The PLM dependency network allows the VAR to have different disciplines work with a shared knowledge structure. A sales consultant can use the left side in the diagram of Fig. 4 to build a business case with existing content relevant to a specific industry segment. Business consultants can use prescribed processes and use-case requirements to define a process-oriented solution for a customer, fitting the business case that the sales consultant defined. IT consultants or software specialists can evaluate the functional use-case implementations with the customer and deliver a working environment, reusing predefined procedures and documentation.

Content for customer presentations, quotation, implementation, training, user documentation, and support can be attached to the elements in the knowledge network. Once the knowledge network is filled with this content, the VAR can use it as a configurator for business cases, quotations, implementation plans, customer-specific user guides, and support documentation. Potentially, it can acquire a predictive behavior with machine learning capabilities.

3 Graph Model

3.1 Limitations of the Wiki System

The added categories, described in the previous section, increase the detail level of the reusable knowledge, but they also increase the complexity of the links between the elements. As shown by the quantities of various element categories in the previous paragraphs, the number of relations can become very large. In the case study, we introduced a wiki system. The wiki worked in the first information model, but with the enhanced model, we encountered difficulties. The consultants mentioned several limitations.

The links between wiki pages are uni-directional. Since the knowledge model should allow associating freely through the BDN in any direction, the return link from the target page to the source page also needs to be created. Also, the links do not contain metadata. The consultants also want to add information to the links independently from the individual linked elements. For example, one would like to know how relevant a particular benefit is to achieve a goal related to or whether a use-case requirement is optional or mandatory in a specific business process. Finally, the wiki system lacks the capabilities to add functionality to automate the creation of configured content, as described in Sect. 2.7.

We searched for a more advanced system to capture and access the information model to overcome these limitations. We found that a graph-based system would be the best fit for the extended dependency network [26]. A graph contains *nodes* (or *vertices*) that have metadata. *Edges* connect nodes. The edges can also have metadata. From one node, a user or system can find all neighboring nodes via the edges. Also, routes can be created from one specific node to any other node, such as finding out how a particular software application contributes to a business goal. The graph will show all the use-case implementation, use-case requirements, business processes, and benefits between the two given nodes. In a graph database, it is possible to extend the properties of a node at any time. Graph databases are also used in content management systems with predicting characteristics. This is relevant when machine learning will be used later to guide users to a recommended solution for a given context.

3.2 Design of the Graph Model

As a first step in designing a graph-based knowledge system to manage the PLM benefit dependency network, we created a graph for the primary dependency network and the extended network, as shown in Fig. 5 and Fig. 6. We use the *label* attribute of the node to group nodes of the same type; for example, all nodes representing goals have “Goal” in their *label*. The edges between the nodes are classified with the *relationship type* attribute, for example, “Measured by”. The label for nodes and the relationship type for edges are essential for the knowledge system to filter information.

The labels in a graph data model fulfill the role that classes fulfill in an object-oriented data model. In Fig. 6, some labels are used multiple times to illustrate that nodes with the same label also can have a relationship with each other. For example, an engineer can report to an engineering manager (both have label “role”), a 2D drawing can reference a 3D model (both have the label “data object”), and a technical document can specify a physical component (both have the label “information object”).

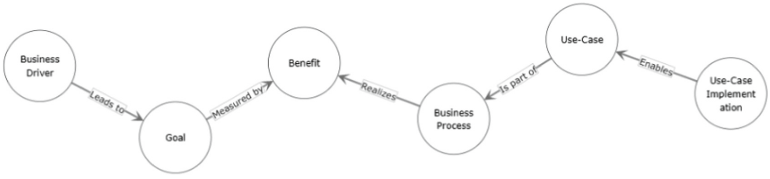

Fig. 5. The basic PLM Benefit Dependency Network as a graph.

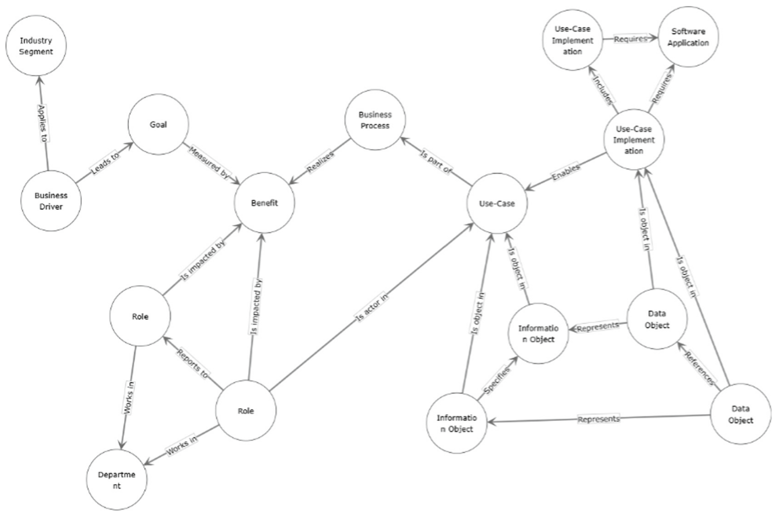

Fig. 6. A graph model of the enhanced dependency network for PLM.

4 Graph-Based Knowledge System Design

We created the first graph model in Neo4j, a widely used Graph Database Management Systems (Graph DBMS) [27]. To build the graph model of the information model, we used Graphileon PE for data entry and visualization. This was sufficient to evaluate the graph traversing capabilities and the modeling options. However, due to the complexity and impracticality of this solution, a tailored front-end needs to be developed later to make it suitable for everyday use and contains specific functionality to support the VAR's use-cases. Therefore, this section includes a few details about the design plans for PLIKSS (**P**roduct **L**ifecycle management **I**mplementation **K**nowledge **S**upport **S**ystem), a graph-based knowledge management and implementation configuration software.

4.1 Graph System Use-Cases

In the wiki-based part of the case study, we were limited to capturing information elements and browsing the structure. In the graph-based approach, more advanced support of additional use-cases is possible.

Implicitly, there must be a process to enter information into the graph. This process concentrates on three use-cases: add nodes, create relationships (edges), and attach documents or media to nodes. Users also enter properties in these nodes and edges.

Then, users must be able to browse the knowledge network efficiently. The basic principle should be that users traverse the graph in any direction and make related nodes appear in the category adjacent to the left or right.

Finally, when the VAR engages with a specific customer, the user must select the nodes that apply to the customer situation. By doing this, the VAR team configures a customer-specific sub-graph within the complete PLM dependency network graph. Figure 7 illustrates how a unique graph of selected nodes can be derived from the (over)complete graph containing every available knowledge object. This filtered customer-specific graph allows the VAR team to generate a customer-specific quotation, implementation plan, training plan, or support plan containing the information related to the selected nodes.

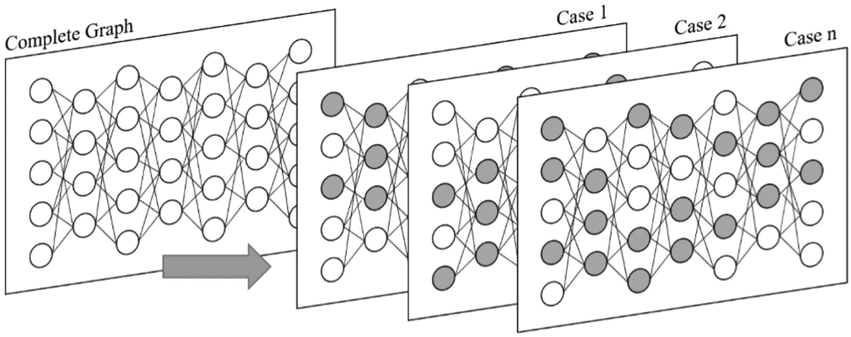

Fig. 7. Simplified illustration of the customer-specific selection process in the PLM dependency network. The user selects only the relevant nodes for the case (grey), resulting in a unique graph for each customer case.

The VAR can also enter specific information about the customer case and the success of a configured implementation, i.e., customer size, product type, the realization of benefits, fit of use-cases with the customer, and accuracy of the project's planning and budget. A future machine learning strategy could help find more successful scenarios and recommend better implementations to less experienced consultants with this information.

4.2 Application Concept for PLIKSS

PLIKSS needs to become an object-oriented graph database in a modular fashion. The graph database system Neo4j supports a binary protocol called Bolt [28]. The binary protocol is enabled in Neo4j by default, which means that you can use any language driver that supports it. Neo4j officially provides drivers for .Net, Java, Python, JavaScript, Spring, and Go. For PLIKSS, we will use the Python programming language to enable machine learning capabilities [29] in the future. PLIKSS will run as browser-enabled software in the cloud. The user accesses the application on an application server through a website hosted on a web server. The application uses the graph data model that resides in a separate server. A file server holds the linked content, i.e., images, documents, videos. All these servers can run in a cloud hosted environment.

The application should also include external information, i.e., YouTube videos, vendor information, Wikipedia pages. Figure 8 shows a conceptual view of this structure.

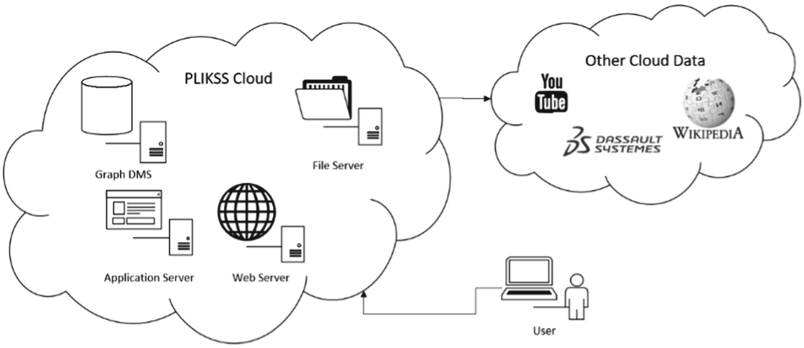

Fig. 8. Conceptual setup of the PLIKSS cloud application

4.3 Machine Learning Strategies in PLIKSS

Machine learning is a method to find complex relationships between input parameters and outcomes. Various forms of machine learning exist to solve multiple types of problems [30].

As a first option, PLIKSS can have a recommendation algorithm in which the experienced users rate the relevance of graph nodes and relationships for individual customer cases. The recommendation algorithm finds combinations of relevant nodes to specific customer profiles based on categorizing the customer attributes (i.e., size, industry, product). After feeding a sufficient number of cases into PLIKSS, the system can recommend graph model configurations for new customer cases based on the customer's profile. This type of algorithm is also used in e-commerce platforms and streaming services, where these systems recommend new items to users based on previous choices and preferences.

A second capability is a mechanism to calculate the probability of success in new PLM implementations. PLIKSS requires data on the outcome of previously configured implementations to learn a parameter set between selected options in the benefit dependency network and a set of project KPIs. These KPIs can be project management indicators (deviation of budget, deviation of hours, deviation of duration, user acceptance, ROI) or value-oriented indicators (achievement of benefits, goals). A detailed study has to determine which algorithm best suits this approach. We will start experimenting with multivariate linear and logistic regression (shallow learning). If this does not result in a reliable model, we will proceed with neural networks (deep learning).

The ultimate challenge for PLIKSS will be building a complete PLM implementation for a given customer profile. This approach requires the users of PLIKSS to register many implementations and their outcomes in a structured way. Then, with reinforcement learning algorithms, the system might find which patterns lead to the desired outcomes.

5 Conclusion and Future Research

The research leading to this paper has shown that acquisition and transfer of knowledge is a significant challenge for VARs. The broadening portfolio vendors and increasing demands of the industry for multidisciplinary process support intensify this challenge. The case studies that we performed with a PLM VAR have shown that the application of reusable modular benefit dependency networks accelerates knowledge transfer and increases insight into the industry value of business processes and supporting software. This insight helps both VAR and customers focus on the elements that contribute most to the desired outcome.

The feedback from the case studies with a wiki-based approach has led to improvements in the knowledge information model and showed limitations of the wiki-based system to manage and use the information model. This feedback resulted in a proposal for a dedicated software that can hold the knowledge information model and is efficient to maintain and operate. The wiki-based case study was convincing enough for the VAR to create a budget to develop the first version of this software. The development of PLIKSS started in April 2020.

The future steps in this research will focus on the development of PLIKSS and define a new case study to measure the efficacy of the knowledge capture and reuse with this software in commercial PLM implementations. After that, we will perform quantitative measurement of the PLM implementation success using PLIKSS and investigate machine learning capabilities in PLIKSS.

This research contributes both to the industry and the academic community. The industry will benefit from capturing and reusing knowledge that was previously only available in experts' minds. The academic community will benefit from the structured data collection of many PLM implementations and learn more about the factors that influence PLM implementation success.

References

1. Stark, J.: Product lifecycle management. In: Product Lifecycle Management (Volume 1). Decision Engineering. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17440-2_1
2. CIMdata. About PLM (2017). https://www.cimdata.com/en/resources/about-plm. Accessed 26 Jan 2019
3. Gartner. IT Glossary - Product Lifecycle Management (2018). https://www.gartner.com/it-glossary/product-lifecycle-management-plm. Accessed 26 Jan 2019
4. Silventoinen, A., et al.: Towards future PLM maturity assessment dimensions. In: International Conference on Product Lifecycle Management, Eindhoven, The Netherlands (2011)
5. Stark, J.: Product Lifecycle Management (Volume 3): The Executive Summary. Decision Engineering (DE). Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72236-8
6. Pang, A., Markovski, M., Markovska, A.: Top 10 PLM and Engineering Software Vendors, Market Size and Market Forecast 2019–2024 (2020). https://www.appsruntheworld.com/top-10-product-lifecycle-management-engineering-software-vendors-and-market-forecast/. Accessed 5 Mar 2021

7. Eddy, N.: Medidata acquired by Dassault Systèmes for $5.8 billion (2019). https://www.healthcareitnews.com/news/medidata-acquired-dassault-syst-mes-58-billion. Accessed 5 Mar 2021
8. Dassault Systèmes Completes IQMS Acquisition (2019). https://www.businesswire.com/news/home/20190107006054/en/Dassault-Syst%C3%A8mes-Completes-IQMS-Acquisition. Accessed 5 Mar 2021
9. McFall, R.: What Happened to the PLM VAR Channel? In Alliance Alchemy. PLM Alliances (2020)
10. Nieuwenhuis, L.J., Ehrenhard, M.L., Prause, L.: The shift to cloud computing: the impact of disruptive technology on the enterprise software business ecosystem. Technol. Forecast. Soc. Chang. **129**, 308–313 (2018)
11. Shilovitsky, O.: How Different Will be a Reseller Model for SaaS PLM? - Beyond PLM (2020)
12. The Fourth Industrial Revolution: what it means, how to respond (2016). Available from: The Fourth Industrial Revolution: what it means, how to respond. Accessed 05 Mar 2018
13. Stabell, C.B., Fjeldstad, Ø.D.: Configuring value for competitive advantage: on chains, shops, and networks. Strateg. Manag. J. **19**(5), 413–437 (1998)
14. Koomen, B.: PLM in SME, what are we missing? An alternative view on PLM implementation for SME. In: Chiabert, P., Bouras, A., Noël, F., Ríos, J. (eds.) PLM 2018. IAICT, vol. 540, pp. 681–691. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01614-2_62
15. Bokinge, M., Malmqvist, J.: PLM implementation guidelines, relevance and application in practice: a discussion of findings from a retrospective case study. Int. J. Prod. Lifecycle Manage. **6**(1), 79 (2012)
16. Koomen, B.: A knowledge-based approach for PLM implementation using modular benefits dependency networks. In: Nyffenegger, F., Ríos, J., Rivest, L., Bouras, A. (eds.) PLM 2020. IAICT, vol. 594, pp. 553–562. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62807-9_44
17. Peppard, J., Ward, J., Daniel, E.: Managing the realization of business benefits from IT investments. MIS Q. **6**(1) (2007)
18. Ward, J., Daniel, E.: Benefits Management: Delivering Value from IS & IT Investments. Wiley (2005)
19. Dassault_Systèmes: 3DEXPERIENCE Value Model App. (2021)
20. Dassault_Systèmes: Transforming Industries, Markets and Customer Experiences (2021). https://ifwe.3ds.com/. Accessed 22 Apr 2021
21. Dassault_Systèmes: Industrial Equipment (2021). https://ifwe.3ds.com/industrial-equipment. Accessed 20 Apr 2021
22. Koomen, B., Hoogeboom, M., Schellens, V.: PLM implementation success rate in SME. An empirical study of implementation projects, preliminary findings. In: Fortin, C., Rivest, L., Bernard, A., Bouras, A. (eds.) PLM 2019. IAICT, vol. 565, pp. 47–57. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-42250-9_5
23. Informatik, V.: Leitfaden zur Erstellung eines unternehmensspezifischen PLM-Konzeptes: Product-lifecycle-Management; transparente Prozesse und konsistente Informationen im Produktlebenszyklus. Informatik. Frankfurt, M. VDMA-Verl (2008)
24. Ward, A.C., Sobek II, D.K., Lean Product and Process Development, 2nd edn. The Lean Enterprise Institute, Cambridge, Mass (2014)
25. Dassault_Systèmes: Portfolio Navigator (2021)
26. Ray, S.S.: Graph Theory with Algorithms and Its Applications. Springer, India (2013). https://doi.org/10.1007/978-81-322-0750-4

27. DB-Engines Ranking - Trend of Graph DBMS Popularity (2021). https://db-engines.com/en/ranking_trend/graph+dbms. Accessed 3 Mar 2021
28. Neo4j_Technology_Inc.: Bolt Protocol (2016). https://boltprotocol.org/. Accessed 7 Mar 2021
29. Srivastava, P.: Why use Python for AI and Machine Learning? (2020). https://technative.io/why-use-python-for-ai-and-machine-learning/. Accessed 30 Mar 2021
30. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Boston (2006). https://doi.org/10.1007/978-1-4615-7566-5