Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 196 (2022) 314–323

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Using a graph database for the ontology-based information integration of business objects from heterogenous Business Information Systems

Carolin Blankenberg, Berit Gebel-Sauer, Petra Schubert*

University of Koblenz, Center for Enterprise Information Research, Universitaetsstr. 1, 56070 Koblenz, Germany

Abstract

This paper reports on findings from a project on *information integration* from *multiple* Business Information Systems with the help of a *user-specific Enterprise Knowledge Graph*. Most ERP systems currently in use store information objects in relational databases. Research in Web Sciences has shown that *graph structures* present information in a *more intuitive way* that is easier to interpret for humans. Following a DSR approach, we developed a concept for storing an ontology in a graph database that allows us to map ERP objects and load them at runtime. This allows the end user to navigate through the graph structure, thus providing an intuitive and quick access to essential job-related information. We evaluated the suggested concept with a prototype following the paradigm of polyglot persistence; the prototype was equipped with a graph database to store the company-specific ontology in its native form. The program code was encapsulated into a separate module following a service-oriented software design.

Peer-review under responsibility of the scientific committee of the CENTERIS –International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Keywords: IS Integration; Enterprise Knowledge Graph; Ontology; Graph Database;

* Corresponding author. Tel.: +49 261 287-2525; fax: +49 261 287-100-2525.
E-mail address: schubert@uni-koblenz.de

1877-0509 © 2021 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the CENTERIS –International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021
10.1016/j.procs.2021.12.019

1. Introduction

The research reported in this paper is part of a longitudinal research project on Enterprise Knowledge Graphs. In the first phase of the project the research team developed a concept and a software prototype for ontology-based data integration [1]. The concept is called “SoNBO” (Social Network of Business Objects). The complementary software provides a navigation interface to an Enterprise Knowledge Graph and is called “SoNBO Explorer”. Following the principles of Design Science Research (DSR), the software was implemented in a company and the implementation project was used to evaluate the SoNBO Concept. The findings were documented in an in-depth case study [2]. The original prototype was built using a document-based database which suited the specific IT requirements of the evaluation company. The software solution was positively received by the employees because the user-specific Enterprise Knowledge Graph represents a unique and valuable access to information that is scattered in the heterogeneous information systems (ERP/CRM) of the company.

The first prototype was built on proprietary software and whilst providing an ideal testbed is not fully built on open standards. The research team identified the use of *open standards* and a *modern programming architecture* as an open requirement and remaining research gap. In this paper, we describe the exploratory work that addresses some of these identified open issues, in particular the storing of the ontology in a graph database. An in-depth literature review in the area of enterprise ontologies showed that the principle of polyglot persistence [3] provides us with the necessary guideline for a next round of improvements. Polyglot persistence recommends that data be stored in the type of database that is best suited for its data structure. Ontologies can be best represented by graph structures and call for a graph database. We started another cycle in our DSR project and extracted and encapsulated the ontology part of the application into a separate software module which, at the same time, is a step towards service-oriented software design. During the DSR “suggestion phase” we identified open standards from Linked Enterprise Data (LED) and Ontology-based Data Access (OBDA) that we included in our further development of the EKG prototype. The externalisation of the code for the ontology into a separate module also allowed us to implement the OBDA principle of working with a harmonisation layer that makes it easier to map to the structures of (multiple) heterogenous backend systems. The functionality of the resulting software module has similar characteristics to an EDI gateway in which the structure of incoming documents is converted to the structure of the internal recipient system (by means of mapping tables).

The development of the prototype was iterative and required multiple cycles. The results showed that the externalisation of the ontology in a graph database is a promising solution. The new module “intermediates” between the software that generates the knowledge graph for the user and the heterogenous information systems that provide us with the information objects.

The paper is organised according to the phases of the DSR process: We begin with the suggestion phase, in which we present an excerpt of the findings from our literature review. The general idea is to show how we transferred concepts and technologies from the Open Web, in particular the research field of Web and Data Sciences, to the area of Business Information Systems. This is followed by the Research Design. We present our suggested solution, the use of a graph database and the modularisation of the code that stores the ontology following the principle of polyglot persistence. The prototype and its new Java class the “OntologyService” is then presented. We discuss our findings from the evaluation of the prototype and summarise what we learned in the conclusion.

2. Awareness of Problem: Literature Review

The following sections describe the most important findings from the literature review in different topic areas: 1) Business Information Systems as sources of enterprise information and 2) technologies for knowledge graphs developed for the Web (graph databases). The idea is to transfer and apply ideas from the open Internet (Web Sciences) to a company context.

2.1. Enterprise environment: Business Information Systems (BIS)

Business Information Systems (BIS) provide the digital support of all activity in an enterprise. The term BIS is an umbrella term and comprises different types of software [4]: *Process-aware* systems for *Enterprise Resource*

Planning (ERP, such as SAP ERP) and *Customer Relationship Management* (CRM) consist of different modules that support functional areas of a company such as financial management, human resource management, logistics, warehouse management or sales/marketing [5]. The information about all business objects, such as people or products, which is crucial for a company, is stored in these systems. For example, without an ERP system, no invoice can be issued to the customer and therefore no sales can be generated.

Enterprise Collaboration Systems (ECS, such as HCL Connections) complement ERP systems by supporting employees in their joint work. ECS provide modules for communication, cooperation, coordination and the creation and management of shared content [6]. The forerunner of the ECS was groupware (e-mail, group calendar, group support systems), which has in recent years been extended by social software components (such as social profiles, wiki, blog) [7], [8].

Enterprise Content Management Systems (ECMS) (such as Alfresco) are aimed at the management of content (files, text/multimedia documents). This category also includes Web Content Management Systems (CMS). *Business Process Management* (BPM) systems (such as Camunda) are geared towards the support and coordination of business processes and are controlled by explicit process representation/definition [9]. In addition, most companies still have specific production systems (e.g. for production control). There is no clear distinction between software types in companies, as some software products offer functionalities from different types of software [4].

The prototype discussed in this paper is embedded into an ECS (user interface) and the Enterprise Knowledge Graph draws data from multiple ERP and CRM systems (source systems).

2.2. Web environment: Graph-based and Ontology-based Information Access

The data from ERP systems (Business Objects) [10] is usually stored in *relational databases*. Saving business objects in a table-like structure affects the user front end, which are often complex. For example, the employee must click down several levels to receive relevant information (e.g. the payment status of an invoice). This is a time-consuming process in everyday work. The challenge of an efficient information access has been discussed in semantic web research. Semantic web research pursues the goal of aggregating information that is stored and managed by a plethora of different users. The aim of knowledge graphs is to aggregate and present this scattered information in a coherent form for the end user. The basic idea is to represent information in a graph, as this is more intuitive for humans than acquiring information in other structures (e.g. tables) [11]. Semantic web research introduced this idea to the web by linking the information from various web pages to a graph that is presented to the end user to access information. A prominent example is Google's Knowledge Graph [12], but DBPedia ("semantic sister" of Wikipedia), YAGO or Facebook's Entity Graph are also examples of implementations in the community. The common denominator is the use of linked data [13], [14]. With linked data, a knowledge graph is generated using four principles: In a nutshell, the principles are about describing the information on the websites in a uniform language and then providing them with links or links to other information. In this way, information is also integrated under one surface. The idea of displaying information in a graph is not new and has been intensively researched in computer science since the turn of the millennium and implemented in practice, when the influential W3C was able to provide the necessary standards for it. Such standards are particularly important for the web community since the translation of the websites with linked data must be implemented in a uniform data schema in which many different people are independently involved.

The requirements of the semantic web also include the integration to be *ontology-based*. This means that the data should not only be translated into a uniform language (linked data), but that an ontology should be used as a schema that transfers the information directly with the help of higher-level concepts (e.g. painter, painting, artist, ...) in a classified and structured form [15]. The idea of the semantic web is that there is an ontology, whose structure defines how web pages are translated [16]. The ontology in connection with the information on the websites (= fact base) then results in a knowledge graph and ideally, a comprehensive knowledge graph then exists for the entire web. The knowledge graph, therefore, continues the idea of linked data using an ontology. In reality, ontologies and thus knowledge graphs have emerged in various domains and not all websites have been translated so far [16]. As already mentioned, DBPedia exists as a "semantic sister" for large knowledge databases such as Wikipedia.

A common definition for ontologies is according to Studer et al. [17] (in Guarino et al. [18]): "An ontology is a formal, explicit specification of a shared conceptualization." A conceptualization is an abstract and simplified view of the world (or specific domain) that is to be represented for a specific purpose [18], [19]. This should be described with the help of a formal and explicit specification using standards to enable a common ("shared") perspective. Standards are, for example, OWL, RDF or SKOS, which serve as the ontology description language. These are characterized by a high semantic expressiveness, which is required for increased complexity [18], [20]–[22]. The ontology-based integration approaches KG (Knowledge Graph; explained in the previous paragraph) and OBDA (Ontology-based Data Access) use this ontology principle to integrate and make the information from the source systems available to the end user. OBDA is understood to be an overlay of the usual data layer of an information system with a conceptual layer (= ontology) that is to be made available in the client. Such a layer enables the user to get a conceptual view of the information in the system. This view abstracts the way in which this information is managed in the data layer of the system itself [23]. The scientific literature demands that the knowledge graph concept should be transferred to companies as an *enterprise* knowledge graph or linked *enterprise* data using linked data, which is still too little done in practice. These two approaches should replace previous integration approaches in the area of EAI (Enterprise Application Integration), which have been used since the 1990s for the integration of information in the distributed and complex business software (= source systems in the context of integration). Since most of the older EAI approaches cannot react flexibly to the changing needs of a company, EKG (= Enterprise Knowledge Graph) with LED (Linked Enterprise Data) is intended to address these challenges. In practice, however, these remain an abstract idea [24]. The research described in this paper addresses the missing transfer from research to practice. We apply the principles of LD and OBDA to an operational enterprise software that provides the user with a user-specific Enterprise Knowledge Graph. The revised SoNBO Explorer will be described in the next section.

2.3. From Web to Enterprise: Ontology-based Information Integration

In ontology-based integration approaches for enterprises in general, an ontology is used to describe the data necessary for the integration. For example, it is specified that the information about an invoice requires information about customer, invoice date, amount and date of receipt of payment. In addition, we need to store the information about its location (where is this information stored? e.g., in which SQL table). If this abstract information is now linked with the concrete data (= fact base) in the source system (e.g. the ERP system), a knowledge graph is created, which is composed of the ontology and the fact base. If the requirements for the display of information change, for example, we want to add the invoice *items* to the knowledge graph, this only needs to be adjusted in the ontology. This change in the ontology is only a matter of configuration ("customization") and there is no need to change the program code. The ontology-based integration approaches EKG or OBDA generally use semantic web technologies to describe the ontology and thus standards that have been defined by the W3C. These standards include, for example, the ontology description languages RDF(S) or OWL, which can be called up with the query language SPARQL or OWL-QL [25]. For the storing of the ontology, which is described with RDF, for example, NoSQL databases are used as RDF store, which can store the RDF triples.

The above-mentioned ontology-based integration approaches are firmly rooted in the research area of the semantic web and they use the described technologies to enable further possibilities such as reasoning. The ontology-based integration approach used in our research is called SoNBO (Social Network of Business Objects). SoNBO differs from the above concepts in terms of the structure of the ontology and the resulting knowledge graph. As the name indicates, the structure of the knowledge graph follows the ideas of a social network known from Enterprise Social Software and from public Social Media platforms such as Facebook. In social networks a node is always a person. SoNBO extends this idea by adding "things" (such as a building or an invoice) as nodes. The principle of the navigation in the network (surfing from node to node along the edges) is identical in both approaches. SoNBO is a concept for enterprises in which access to the ERP information is provided in a Web interface (the SoNBO Explorer) using a knowledge graph instead of using the source applications and running a query against the tables of an SQL database. This gives the user a new and intuitive access in the form of a graph structure. The concept was implemented in practice using proprietary software that does not make use of semantic web standards. As already mentioned, SoNBO was built for the evaluation company and since the company uses HCL Domino technology, a document database (nsf file) was implemented to store the ontology.

2.4. From Web to Enterprise: Databases

In this section, we are taking a closer look at the different types of databases already mentioned above. Most BIS are provided as commercial standard software and are developed by large software vendors (SAP, IBM, MS) for use in many user companies. The majority of these BIS store data in *relational* databases, which emerged in the 80s [26]. The data is stored in tables where the rows represent objects and the columns their respective attributes [27]. At the same time, other database types such as Lotus Notes or Berkerley DB came up which do not use a relational data model. Such systems, summarised under the term "NoSQL", have become increasingly popular with the advent of Web 2.0 and the resulting large amounts of heterogeneous data. In the following, big Internet companies such as Google, Amazon and Facebook developed their own databases such as BigTable, Dynamo and Cassandra to suit their specific needs [28]. Most NoSQL systems have in common that they have no relational model, do not use a relational model, are designed for clusters, are open source accessible, do not specify a schema restriction or only a weak one and are designed to process large amounts of data. Further, they support a simple data replication, they offer a simple API, and a different consistency model (BASE or Eventually Consistent) is available instead of ACID [28].

In the research described in this paper, two types of NoSQL databases are of interest: *document databases* as they are used by Notes and *graph databases* that natively store graph structures. Like relational databases, document databases have their beginnings in the 80s with Lotus Notes as leader. They store semi-structured data as key-value pairs in documents which should have a standard data exchange format like XML or JSON. Therefore, a document database is a collection of semi-structured documents [26].

Another type of NoSQL database are *graph databases* which are based on graph theory and store graphs or graph-like structures. They were also developed in the 80s and were used to design and to manage networks. Due to research in semantic web, more and more graph databases were developed. With the emergence of smartphones and the *location-based services* for these, the need for such systems for processing web data and their popularity grew. The most popular type at the moment are native graph databases which model property graphs [28].

The *Resource Description Framework* (RDF) is an influential standard model in the research field of semantic web that represents graph-like structures. It describes data in the form of triples: 1) subject (vertex), 2) predicate (edge), 3) object (vertex). Every component is expressed as URI where the vertices point to a resource while the predicate describes the relation of the vertices [29]. Such triples could be stored in RDF stores, which could be any kind of store if it is able to handle RDF data. This includes input and output of serialised RDF data as well as an API for integration with applications [30].

2.5. Research Questions

In summary, it can be said that existing BIS are mostly based on relational database tables for storing business objects. However, it is more in human nature to represent information as a graph [11], for which graph databases offer a suitable technology. This approach has been driven by the ontology-based integration approaches from the semantic web since the 2000s, and during this, various technologies and standards have emerged from the influential W3C community. There, the ontology, which contains the schema for access to the source systems, is usually stored in an RDF store, which could be any kind of NoSQL database as long as it can handle RDF data like it is described in the previous section. Other web applications apart from information integration also use the advantages of graph databases by storing network structures between people. An example is Facebook (use of the self-developed graph database "Cassandra"). Our own ontology-based integration concept (SoNBO) uses a document database to store the ontology. The aim of this paper is to incorporate the open standard ideas of the field of web science into our SoNBO application. Up to now, a graph database has generally been used to store web applications and less in the corporate context to store business objects (which are traditionally relational in SQL databases) as graphs. By changing the storage, the business objects can be visualised as a graph (also: social network) for the end user, which is revolutionary for enterprises and a promising avenue for IS Research.

Our research for this particular DSR improvement cycle is guided by the following research questions:

1. Following the principle of *polyglot persistence* [3], [28], what are the (positive or negative) effects of storing the ontology in a *graph database* and leaving the other contents of the application in the document database?

2. Following latest ideas from *service-oriented software design*, how can we *encapsulate the ontology part* in a separate software module?

3. Research Design

The research design is based on design science research (DSR) suggested by Vaishnavi and Kuechler [31], which is also called “improvement research” [31, p. 9]. The suggested methodology follows an iterative design with five process steps: awareness of problem, suggestion, development of artefact, evaluation, and conclusion. Within one DSR iteration, it is possible to loop back to an earlier step. Fig. 1 shows the DSR steps and the activities carried out in the phases. We combined the two phases development & evaluation.

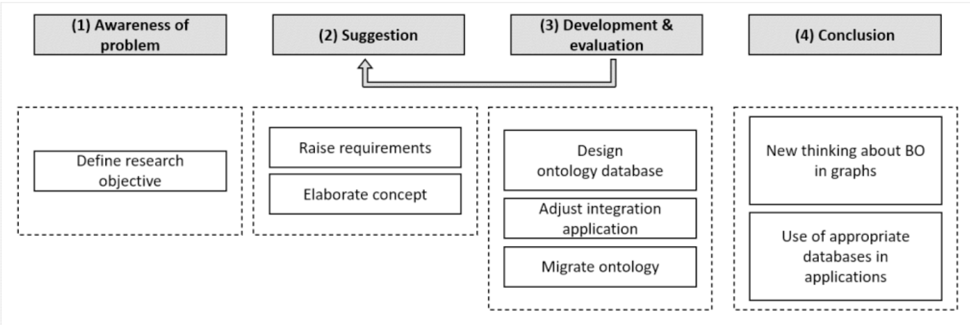

Fig. 1. DSR phases (following [31])

(1) First, the research questions and the resulting research objectives were defined. The research described in this paper was motivated by the perceived need for storing the ontology data in a database that is optimised for its graph structure. A system analysis of the existing SoNBO application as well as literature analysis [32] were performed to identify appropriate concepts from the semantic web for storing graph-like structures. (2) The findings let to the requirements for the new software module and the new concept for the implementation of the graph database. In the next step (3) the concept was implemented in an iterative process of adding changes and running a (self) evaluation. This loop was performed until the results finally met our objectives. In the last step (4), we discussed the findings especially regarding the combination of different database types (polyglot persistence) and the implementation of the service-oriented programming paradigm (modularisation).

4. Suggestion: Using a Graph Database for the SoNBO Ontology

To reach the research goals two perspectives had to be considered, the design of the ontology database and the source code of the SoNBO application which is the integration application. For the design of the database, the characteristics of property graphs were investigated because graph databases are built on this concept. In the application, those operations which belong to ontology database queries were encapsulated in a new Java class. This is a first step to reach independence from the underlying database systems.

4.1. Graph Database: Storing the Ontology for ERP Objects

Different data structures have different requirements for storage and NoSQL environments offer a wealth of different database types. It is important for a developer to know the different database types and choose the appropriate one [28]. Applications often access a combination of different databases to retrieve the necessary information. It is recommended that information is stored in databases that best accommodate the respective data structure. As a consequence, applications must handle different kinds of databases. This phenomenon is called *polyglot persistence* [3], [28].

In this research project, a new database was added to the SoNBO Explorer for the purpose of storing the ontology. The SoNBO Explorer is an integration application that extracts information from the source systems, links it to a social network with the help of a (company-specific) ontology and provides the resulting Enterprise Knowledge Graph to the user. The original software application was built in HCL Domino and used a document database to store the ontology. Fig. 2 shows the new architecture. The ontology was moved to the new graph database (Neo4j). The other information necessary to build the SoNBO Explorer remained in the original Notes database (SoNBO.nsf).

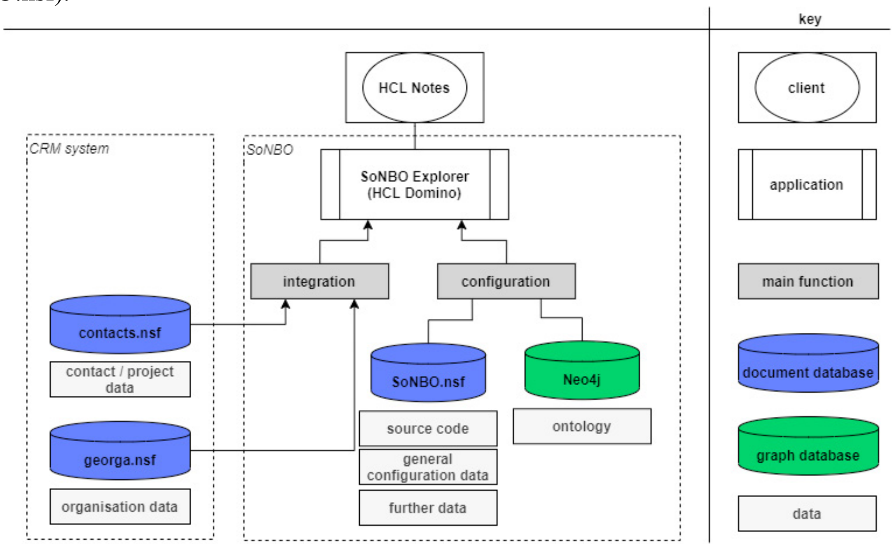

Fig. 2. Architecture necessary to generate the Enterprise Knowledge Graph

Graph databases with the property graph model in mind [28] seem to be the perfect fit for the needs of an ontology database. The vertices of the graph which are called nodes in the database are used to store the information about the different business object types while the edges, called relationships in the database, are used to model the relations between the business object types. Both the nodes and relationships are categorised with the help of labels. Further information about the business object or the relationship, like in which information system the concrete information is kept, are stored as properties of the node or relationship. As Neo4j does not provide *undirected relationships* (which you need to describe an ontology) these relationships are designed by using *two opposing directed relationships*.

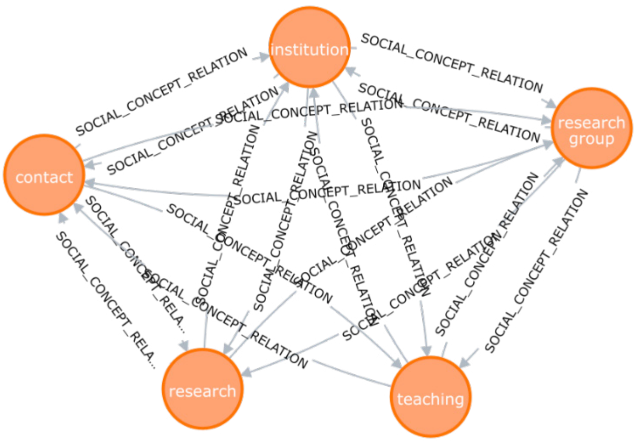

Fig. 3. Graph Database Excerpt

Fig. 3 shows an excerpt of the ontology database of our research group (CEIR, University of Koblenz). This is an instance of the SoNBO Explorer using an operational CRM system (based on GEDYS IntraWare CRM). Before the implementation of Neo4j, the data could only be accessed with the help of views which have a table-like structure. Now, every business object (e.g. the SoNBO research project) is assigned to a concept (research) in the ontology, which is now stored in the graph database (see Fig. 3). The concepts are related to each other (e.g. "research" is connected to the nodes "research group", "institution" and "contact"). A concept can have sub-concepts such as "research project" which are also represented as nodes and can have relationships between each other. These in turn have attributes such as "research name" or "project start" that define from which system the concrete data will be retrieved. So, the concrete research project (e.g. "SoNBO research project") is still stored in the source system (CRM system) but the ontology makes the relationships that already exist in the source systems visible as a graph. This is done during runtime as live access, which means that the knowledge graph is not stored. This implementation type is called *virtualization* of a knowledge graph. It is also feasible to store this knowledge graph. This is then called *materialization* [33]. The SoNBO application has the characteristics of a standard software because it could be adapted to other user organisations simply by changing the configuration of the ontology instead of requiring changes to the program code. Fig. 3 shows the concepts and their relationships (i.e. the ontology). The data describing the ontology is now stored in the graph database in its "native form" instead of a more complicated, converted form in relational tables. The new (graph-based) way of thinking about business objects as a network is now also directly reflected in the stored data structure.

4.2. Development: Java Class: OntologyService

In the second step of our development we turned our attention towards a more service-oriented software architecture. Now that the polyglot persistence had been addressed and the application was running on multiple different databases, it paved the way to encapsulating and extracting the software module for the ontology. In the end, every database could be managed by a specialised software service. This makes the application independent from the database type; changes in the database would not mean that the main application needs to be adjusted [3] but only the specialised database service.

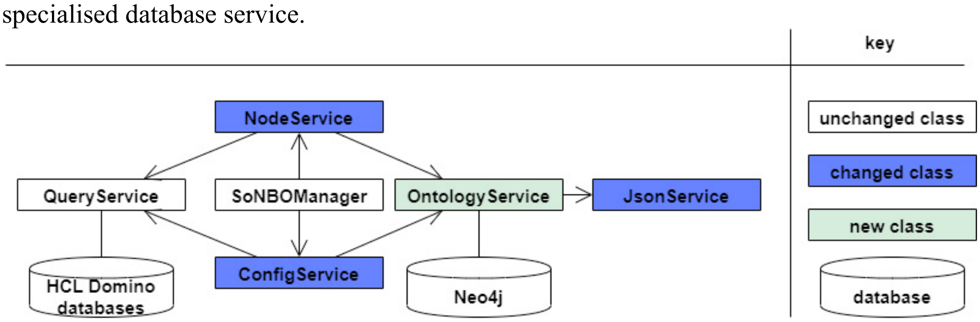

Fig. 4. Java classes

In this phase of our long-term SoNBO project, the first step towards this independency was achieved (Fig. 4). In the Java source code, the methods which operate on the database to get ontology information are encapsulated into a new Java class *OntologyService*. So the application does not operate directly on the database anymore as well as it does not need to handle the resulting data. All database queries and results belonging to the ontology are processed by the new class and are delivered in the necessary format so that the application can work with it. Queries relating to the document databases are still managed by the Java class *QueryService*. However, the results still must be processed by the application itself.

5. Evaluation: Discussion of Findings

The evaluation of the SoNBO Explorer showed that ERP objects can be represented for practical purposes using an ontology as a (social) graph, thus opening up a new way of thinking about ERP objects in IS Research. ERP objects are no longer just to be understood as relational data but also as enterprise knowledge graphs. This graph structure makes human access to information more intuitive because the ERP objects are shown in their connections to one another and such a display is easier for people to grasp [11]. In addition, revealing the links also allows the exploration of new knowledge about the ERP objects (through the links to one another). It is important to note that the graph view is an additional view on the data. The data itself is still kept in its original form (in most cases in a relational model) in the ERP system.

In this section we are now responding to our initial research objectives, the principle of *polyglot persistence* [3], [28] and the *encapsulation of the ontology* as a step towards a more service-oriented software design. Both of our objectives were positively met in the implementation and the evaluation of the enhanced prototype.

The evaluation of the new SoNBO Explorer showed that this new and innovative thinking about ERP objects that are connected through graph structures can also be implemented by using graph-based databases thus following the principles of polyglot persistence. Before this project, the graph-like ontology could only be accessed in the SoNBO Explorer with the help of table-like views which meant that relationships were difficult to identify. We learned that it is important to analyse the structure of the data and choose an appropriate database type. It was interesting to see how new innovative technologies from open Internet initiatives such as NoSQL and open standards from the semantic web can be used in a business environment. Nowadays it is possible to implement these concepts not only in theory but also in practice, as the technological prerequisites (e.g. mature graph databases) are now in place and research progress has been made in other areas of computer science (e.g. semantic web). We have also witnessed that graph databases work well in public social networks (e.g. Facebook) and we believe that the popularity of the Social Media will also pave the way to an increased openness to introduce such concepts in companies.

The ontology data is now stored in a more standardised way, so it is easier to transform it to other standards such as RDF as there are already existing tools for export and import standard forms. The graph database also supports the development of an ontology since the data does not have to be serialised but can be taken over as it is modelled.

Further, the software that manages the databases is now *encapsulated*. This allows the application to be independent of the underlying databases and makes it easier to add changes. This is a step towards the "service wrapped database" described by Sadalage and Fowler [3], which allows different applications to use one database by using the offered services instead of a direct database access. Using these services makes redundant data in different information systems unnecessary. Encapsulating the databases also leads to the ability to choose the database depending on the structure of the data (polyglot persistence). This is a new thinking compared to before where data had to be serialized in a way that the database which is supported by the application could handle it.

6. Conclusion

We believe that the use of open software standards will make our concept more accessible and thus more attractive for a broader use in companies. The use of open standards had two positive side effects: 1) We can now develop the company ontology independent of the main application and 2) the graph can be visualised with existing tools for ontologies. Before, we needed a specialised (self-developed) tool for the visualisation of the graph. Another advantage is that existing tools are constantly further developed and have a greater range of functionality. This will help with the cooperative development of the ontology with the domain experts in companies.

We believe that the modularisation of the application has improved its maintainability. The ontology can now be developed and "replaced/exchanged" in a much easier way. The implementation of the specific company ontology is now similar to a "customization process" for an ERP system.

A possible research issue for the next phase of the SoNBO project is the question of virtualisation versus materialisation. At present, only the *ontology data* is stored in the graph database. The business objects are retrieved at *runtime* and the information is directly queried from the source systems (virtualisation) and displayed for the user. There is no redundant data storing. This leads to performance problems especially with large graphs with many business objects, as large amounts of data have to be loaded at runtime. In order to minimise loading times, the knowledge graph could be (continuously) generated and physically stored (materialisation). As for the ontology, a graph database would be the logical choice for the storing of the knowledge graph.

References

[1] B. Gewehr, B. Gebel-Sauer, and P. Schubert, “Social Network of Business Objects (SoNBO): An Innovative Concept for Information Integration in Enterprise Systems,” in *Procedia Computer Science*, 2017, pp. 904–912, [Online]. Available: https://doi.org/10.1016/j.procs.2017.11.117.

[2] B. Gebel-Sauer, “Ontologie-basierte Informationsintegration in der Form eines Social Network of Business Objects (SoNBO),” Universität Koblenz, 2021.

[3] P. J. Sadalage and M. Fowler, *NoSQL Distilled - A Brief Guide to the Emerging World of Polyglot Persistence*. Upper Saddle River, NJ: Pearson Education, Inc., 2013.

[4] P. Schubert and A. Winkelmann, *Betriebswirtschaftliche Anwendungssysteme (Fokus: Enterprise Resource Planning)*. 2020.

[5] T. H. Davenport, “Putting the Enterprise into the Enterprise System,” *Harv. Bus. Rev.*, vol. 76, no. 4, pp. 121–131, 1998.

[6] S. P. Williams, “Das 8C-Modell für kollaborative Technologien,” in *Wettbewerbsfaktor Business Software*, P. Schubert and M. Koch, Eds. München: Hanser, 2011, pp. 11–21.

[7] P. Schubert and J. H. Glitsch, “Adding Structure to Enterprise Collaboration Systems: Identification of Use Cases and Collaboration Scenarios,” in *Procedia Computer Science*, 2015, vol. 64, pp. 161–169.

[8] F. Schwade and P. Schubert, “Social Collaboration Analytics for Enterprise Collaboration Systems: Providing Business Intelligence on Collaboration Activities,” in *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*, 2017, pp. 401–410.

[9] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, 2nd ed. Berlin, Heidelberg: Springer, 2012.

[10] B. Gebel-Sauer and P. Schubert, “Entwicklung einer Definition für Social Business Objects (SBO) zur Modellierung von Unternehmensinformationen,” in *International Conference on Wirtschaftsinformatik (WI)*, 2019, pp. 199–213.

[11] C. Hoede, “On the ontology of knowledge graphs,” in *Conceptual Structures: Applications, Implementation and Theory. Third International Conference on Conceptual Structures, ICCS ’95*, 1995, pp. 308–322, doi: 10.1007/3-540-60161-9_46.

[12] A. Singhal, “Introducing the Knowledge Graph: things, not strings,” *Google Official Blog*, 2012. https://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html (accessed Oct. 20, 2017).

[13] L. Ehrlinger and W. Wöß, “Towards a Definition of Knowledge Graphs,” 2016, Accessed: Sep. 10, 2017. [Online]. Available: http://ceur-ws.org/Vol-1695/paper4.pdf.

[14] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, “Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO,” *Semant. Web*, vol. 9, no. 1, pp. 77–129, 2018.

[15] B. Villazon-Terrazas *et al.*, “Knowledge Graph Foundations,” in *Exploiting Linked Data and Knowledge Graphs in Large Organisations*, J. Z. Pan, G. Vetere, J. M. Gomez-Perez, and H. Wu, Eds. Cham: Springer, 2017, pp. 17–55.

[16] K. Weller, “Ontologien,” in *Grundlagen der praktischen Information und Dokumentation. Handbuch zur Einführung in die Informationswissenschaft und -praxis*, 6th ed., R. Kuhlen, W. Semar, and D. Strauch, Eds. Berlin, Boston: De Gruyter, 2014, pp. 207–218.

[17] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data Knowl. Eng.*, vol. 25, no. 1–2, pp. 161–197, 1998, doi: 10.1016/S0169-023X(97)00056-6.

[18] N. Guarino, D. Oberle, and S. Staab, “What Is an Ontology?,” in *Handbooks on Ontologies*, 2nd ed., S. Staab and R. Studer, Eds. Dordrecht, Heidelberg, London, New York: Springer, 2009, pp. 1–20.

[19] J. L. G. Dietz, *Enterprise Ontology: Theory and Methodology*. Berlin, New York: Springer, 2006.

[20] C. Feilmayr and W. Wöß, “An analysis of ontologies and their success factors for application to business,” *Data Knowl. Eng.*, vol. 101, pp. 1–23, 2016, doi: 10.1016/j.datak.2015.11.003.

[21] M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, “Introduction: Ontology Engineering in a Networked World,” in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds. Berlin, Heidelberg: Springer, 2012, pp. 1–6.

[22] J. M. Gomez-Perez, J. Z. Pan, G. Vetere, and H. Wu, “Enterprise Knowledge Graph: An Introduction,” in *Exploiting Linked Data and Knowledge Graphs in Large Organisations*, J. Z. Pan, G. Vetere, J. M. Gomez-Perez, and H. Wu, Eds. Cham: Springer, 2017, pp. 1–15.

[23] D. Calvanese, G. de Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati, “Ontology-based database access,” in *Proceedings of the 15th Italian Symposium Advanced Database Systems (SEBD 2007)*, 2007, pp. 324–331.

[24] S. Auer, R. Petzsch, and J. Unbehauen, “Datenintegration im Unternehmen mit Linked Enterprise Data,” in *Linked Enterprise Data*, T. Pellegrini, H. Sack, and S. Auer, Eds. Berlin, Heidelberg: Springer, 2014, pp. 85–101.

[25] A. Blumauer, “SKOS as a Key Element in Enterprise Linked Data Strategies,” 2014.

[26] A. B. M. Moniruzzaman and S. A. Hossain, “NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison,” *Int. J. Database Theory Appl.*, vol. 6, no. 4, pp. 1–14, 2013.

[27] J. J. Miller, “Graph database applications and concepts with Neo4j,” in *Proceedings of the Southern Association for Information Systems Conference*, 2013, vol. 2324, p. 36.

[28] S. Edlich, A. Friedland, J. Hampe, and B. Brauer, *NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken*. München: Hanser, 2010.

[29] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data - The Story So Far,” *Int. J. Semant. Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, 2009.

[30] G. E. Modoni, M. Sacco, and W. Terkaj, “A survey of RDF store solutions,” *2014 Int. Conf. Eng. Technol. Innov.*, pp. 1–7, 2014, doi: 10.1109/ICE.2014.6871541.

[31] V. K. Vaishnavi and W. Kuechler, “Design Science Research in Information Systems,” 2004. http://www.desrist.org/design-research-in-information-systems/ (accessed Jan. 16, 2017).

[32] J. vom Brocke, A. Simons, K. Riemer, B. Niehaves, R. Plattfaut, and A. Cleven, “Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research,” *Commun. Assoc. Inf. Syst.*, vol. 37, no. 9, pp. 205–224, 2015.

[33] J. F. Sequeda and D. P. Miranker, “A Pay-As-You-Go Methodology for Ontology-Based Data Access,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 92–96, 2017.