BUDAPESTI CORVINUS EGYETEM

Doctoral School of
Economics, Business
and Informatics

Ph.D. THESIS COLLECTION

Géza Gábor Molnár

Semantic web technologies application possibilities for the „exploratory" OLAP

SUPERVISOR:

Dr Andrea Kő, PhD
university professor

BUDAPEST, 2022

Department of Computer Science

THESIS COLLECTION

Géza Gábor Molnár

Semantic web technologies application possibilities for the „exploratory" OLAP

SUPERVISOR:

Dr Andrea Kő, PhD.
university professor

Table of contents

1 BACKGROUND AND OVERVIEW OF THE RESEARCH

Over the last decade, the amount of unstructured data (e.g., texts) has increased much more than structured data. It is due in no small part to social media and the proliferation of smart devices. As a result, there is now much more unstructured data than structured data. It is estimated that by the end of 2022, their share of total data will be around 93% and 7%, respectively.

The storage of large amounts of data is not a problem for the time being, as the price of storage devices has recently fallen significantly while their capacity has increased. The cost of storing one GB of data has thus fallen to a fraction of its current level. Furthermore, while current storage technologies are expected to reach their limits in a few years, there are already forward-looking developments (e.g. quantum storage) that will be able to keep pace with the increased storage capacity needs.

However, the situation is different when processing unstructured data. They were previously processed manually or simply ignored. However, their growing share of data means that the need to analyse them is increasing. A typical example is when a company wants to aggregate and evaluate feedback from its customers or textual information about a partner. The sheer volume of data often makes it hopeless to perform similar tasks manually. Therefore, it is nowadays particularly important to develop computer procedures that allow the processing and analysis of unstructured data, primarily textual.

The processing of unstructured data as a problem involves many subfields, including knowledge representation, semantic technologies, data, web and text mining, artificial intelligence, including learning algorithms. Semantic technologies, including ontologies, are important tools for structuring and providing a conceptual description of a domain. They help define concepts specific to a domain and the relationships between them.

1.1 Research objective

My research topic is related to the use of semantic technologies (mainly ontologies) in data warehouses. My primary goal is to investigate solutions that enable the combined analysis of structured and unstructured data in data warehouse/business intelligence systems. Such systems are also referred to exploratory OLAP systems in the literature.

The field of exploratory OLAP systems is relatively new. The first models appeared just over ten years ago. It is important to note that most of these models have remained at a conceptual level. In a few cases, prototypes have been developed, but their general applicability has not yet been demonstrated.

For this reason, I aim to develop and evaluate at least one prototype. Working prototypes can also be used to draw conclusions about the feasibility and applicability of a given exploratory OLAP model. I would like to use ontologies as the main semantic web technology in prototyping. Since these are not available in my chosen domain (ticketing system), a further goal is to develop the necessary domain-specific ontology.

1.2 Challenges

There are several difficulties to overcome when developing an exploratory OLAP model or prototype. Some of them are common to all solutions, and others are related to ontologies.

- The first problem is that domain-specific ontologies are generally not available. It is not a trivial task to develop them, preferably automatically, from unstructured source systems.
- The second difficulty arises at an important step in OLAP cube design, namely at the design of aggregations. Unfortunately, only preliminary research results are available so far to solve this problem.
- The third problem is related to the change of ontologies (evolution, versioning). At the moment, this is still in its infancy.
- The next challenge is the ETL process. Integrating unstructured data into the ETL processes of the data warehouse is not easy, as it may contain many elements that are not known by the processes that handle structured data such as non-relational operators, machine learning calculations, and complex data types.
- The integration of the semantic layer with ontologies can raise performance issues that need to be taken into account when optimising the resulting design.
- An exploratory OLAP system must be able to integrate data sources dynamically. This can lead to high computational costs that may even compromise feasibility.
- Exploratory OLAP means automatic access not only to the schema but also, to some extent, to the data. It implies the need to be able to interpret and reason about data at the instance level.
- Finally, the last difficult question is the suitability of semantic web technology to support managerial decision making. The main problem here is the integration of unstructured data and the independence of the individual data sources.

1.3 Research questions

In my research, I aimed to answer the following questions:

- How to make the traditional OLAP and the data warehouse semantic, i.e. how to design the integration of the semantic layer into the data warehouse?
- How can the designed model be put into practice with a prototype?
- What could be an effective validation method for the exploratory OLAP model?

2 RESEARCH METHODS

This chapter describes the methodology used in the research. It includes the design science used in information systems design and software development, data collection and analysis, ontology development and evaluation methodology, and the development methodology used in prototyping.

2.1 Design science

Design science is a set of summarising and analytical techniques and perspectives that can be effectively applied to IT research. The aim is to understand a problem from an information systems perspective. It is usually done through two basic activities:

- Acquiring new knowledge through the creation of an innovative artefact
- Analysis of the feedback received during the use of the artefact

The artefact is the exploratory OLAP prototype and the ticketing ontology in my research. The design and analysis of an artefact are always done in a specific context. The context in our case is the ticketing system.

A design science project is always iterative, with two main activities: design and testing. The design activity can be divided into three parts. These are: investigating the problem, designing a solution to the problem and validating the solution. The iteration of these three sub-activities is called a design cycle.

The design cycle is part of a larger cycle, where the output of the design cycle (the validated problem handling) is delivered to users who use and evaluate it. This larger cycle is called the engineering cycle.

The cycle consists of the following steps:

- Examining the problem. What needs to be improved? (e.g. how to increase the efficiency of text data management in data warehouses)
- Designing a procedure to deal with the problem. Design of one or more artefacts to handle the problem (e.g., exploratory OLAP prototype design)
- Validate the problem handling procedure. Do these plans solve the problem?
- Implement the problem handling procedure. Handle the problem with one of the artefacts (e.g., implementation of an exploratory OLAP prototype)

- Evaluation of the implemented process. How successful is the problem addressed (e.g., how well does the prototype meet the requirements?) This step often marks the start of a new iteration.

This cycle is repeated until the desired result is achieved.

2.2 Data collection and analysis

The first data collection method is completing a systematic literature review in the field (ticketing systems). It helps in identifying the research challenges, developing a conceptual framework for exploratory OLAP, and creating a prototype. Besides, it allows to identify unstructured domain-specific data sources and create an initial ontology.

The following method is to examine the data collected. It includes the analysis of data quality, which mainly involves checking the following:

- Missing values (e.g., fields not filled in)
- Outliers (e.g., a comment field that is much longer than the average)
- Duplications
- Incorrect values (such as words not in the dictionary, abbreviations).

Depending on data quality and examination of the data, it may be necessary to clean the data to prevent inappropriate data entry into the system.

2.3 Ontology development

Although many ontology development methodologies are known today, few of them are widely accepted and sufficiently mature. In my research, I have selected the ontology development methodology to be used, taking into account its advantages and disadvantages and its suitability for the specific task.

I mainly considered the following methodologies:

- TOVE (Toronto Virtual Enterprise): a project. Its aim was to model enterprise operations and describe enterprise integration. The result can be seen as a second-generation expert system. The ontology methodology developed from this project was later used for other purposes (e.g. supply chains).
- CommonKADS: a commonly used methodology for creating knowledge-based systems. Unlike other methodologies, this approach is closely related to the UML notation methodology used in object-oriented programming.

- SENSUS: an ontology-based methodology based on the extension of WordNet (the lexical database of the English language).
- On-To-Knowledge: a methodology developed in the framework of an EU project. The aim was to create a knowledge base based on a large number of heterogeneous unstructured or semi-structured documents. The documents were taken from intranets of large corporations and the web.

After reviewing the literature on the development of the ticketing ontology, I chose a simplified version of the On-To-Knowledge methodology, as it focuses on concept discovery. The discovered concepts will later help generate multidimensional identifiers, dimensions, facts and measures.

The steps of the methodology are shown in the following figure. It is also possible to step back between steps if necessary.

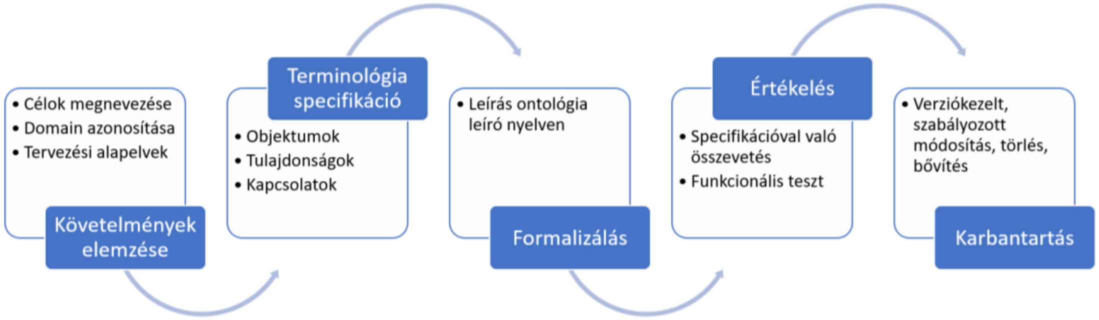


Figure 1: Steps of ontology development based on the On-To-Knowledge methodology (own editing)

2.4 Ontology evaluation

For the ontology evaluation, I chose a criterion-based evaluation method. The evaluation, i.e. the examination of the fulfilment of the specific criteria, involves the following questions:

- Is the resulting ontology model relevant to the domain (in our case, the ticketing system)?
- What is the quality of the resulting ontology behind the conceptual model domain from an independent perspective, i.e. how effective is the knowledge representation?
- How technically appropriate is the resulting ontology, considering the latest technologies and tools?
- How useful is the resulting ontology for a given practical task?
- Can the ontology be used in other applications?

These criteria were checked after the second version of the ticketing ontology was created.

2.5 System development

Prototyping can be seen as a system development task, including software development, so it is advisable to do it according to an appropriately selected methodology.

These methodologies have in common that the same activities (sub-processes) are carried out during development but in a different way and approach. The sub-processes of software development are:

- Specification. During this process, customers and software developers precisely define the tasks to be performed by the software to be produced and the constraints on its operation.
- Development. This sub-process involves designing and building the software.
- Validation. Checking that the software is designed to meet the user's needs.
- Evolution. Update software to meet changing user or market needs.

There are several software development methodologies, and I chose the so-called rapid prototyping method for prototyping. It focuses on rapid, iterative prototyping, thus providing tangible results in the early stages of software development. I used the ideas of Prasad and Abello to create prototypes. The steps to create a prototype are shown in the following figure:

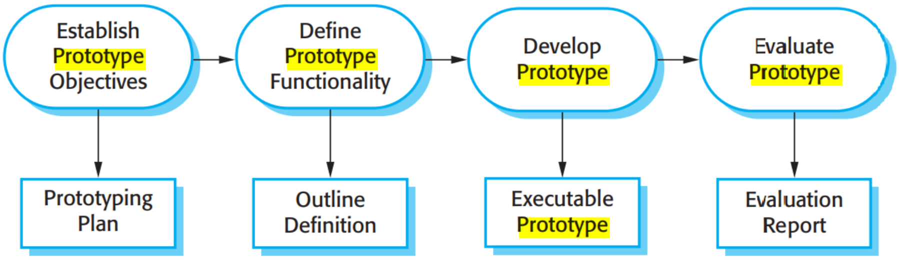


Figure 2: Steps to prototyping

3 RESULTS OF THE RESEARCH

In this section, I present the results of the research. Among these, I would like to highlight the creation of a ticketing ontology and exploratory OLAP prototypes, for which I could not find any examples in the literature. My version of the ticketing ontology does not cover the whole ticketing domain. It focuses mainly on incident reporting.

There are also very few working versions of exploratory OLAP prototypes in practice. The prototypes I have created in this research use existing models (Prasad, Abello), but I simplified them to the extent necessary.

3.1 Investigation of exploratory OLAP models

In the literature, I found three relevant models, which can be attributed to Prasad, Abello and Ibragimov.

In Prasad's model, text extracted from data sources containing unstructured data is first subjected to text analysis procedures, and then the results of the analysis are placed in appropriately designed tables in the database/data warehouse. The fact table and dimension tables have a star layout and are suitable for creating OLAP cubes and reports.

Abello et al.'s exploratory OLAP concept uses semantic web technologies. The key part of the model is ontology-based knowledge representation. Ontologies enable the identification of facts. Furthermore, possible dimensional concepts can be identified for each fact, stored in a well-ordered hierarchical form using functional dependencies.

As a data source, Ibragimov used Linked Open Data (LOD) stored in RDF format. The system he proposes consists of four modules:

- Global Conceptual Schema - stores information about the data frame
- Semantic Query Processor - generates SPARQL queries from MDX queries
- Distributed Query Processor - queries endpoints, collects data
- Source Discovery Schema Builder - keeps in touch with users during schema creation.

A comparison of the three exploratory OLAP models is shown in the following table:

Viewpoint	Abello	Ibragimov	Prasad
Data sources (unstructured)	**Any**	**Linked Open Data**	**Any**
Tools, technologies	**ontologies**	**RDF, MDX, SPARQL**	**text analysis, XML**
Biggest challenge	**ontology --> MD**	**MDX→ SPARQL**	**text analysis**
Does a prototype plan exist?	**not**	**yes**	**yes**
Output location	**nerd**	**nerd**	**data warehouse**
Form of output	**OLAP schema**	**OLAP schema**	**Star schema**

Table 1: Comparison of exploratory OLAP models

3.2 Ticketing ontology

I created two versions of the ticketing ontologies. The first version is a two-level taxonomy to facilitate semantic search in incidents. In the first step, identifying and categorising the incident topic is the most important task. I solve this task using the Latent Dirichlet Allocation (LDA) model. It is an unsupervised machine learning model that assigns topics to text documents. In addition, the model also shows the extent (percentage) to which topics are present in each document. The result is illustrated in the following figure:

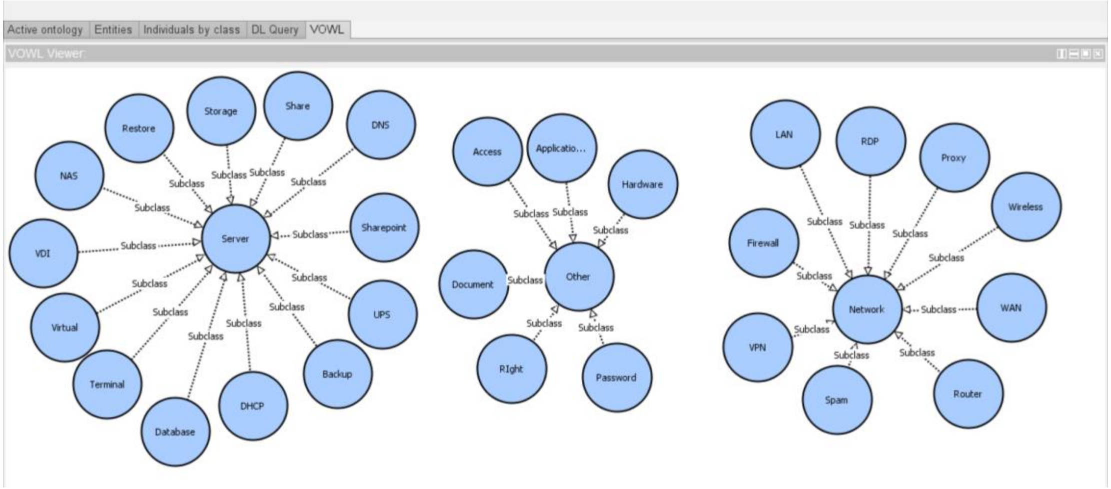


Figure 3: Initial system of ticketing ontologies (detail)

The requirements were extended in the second version of the ontology (needed for the second prototype). The most important new requirement is that the ontology should not only be able to categorise incidents. Instead, as far as possible, it should describe as well as possible the functioning of the ticketing system, including the handling of error reports (incidents), based on the following points:

- The customer reports the problem to the helpdesk, which opens a ticket.
- The text of the bug report consists of a short formulation (subject) and a more detailed description. It also has other characteristics. For example, the time when the ticket was created or the urgency (priority) of the bug.
- The ticket will be assigned to a competent person (assignee, analyst) who will solve the problem. The actions taken during the resolution will be recorded (logging).
- During the solution, certain characteristics of the ticket (e.g., its status) may change several times.
- The aim is not to describe the entire life of the ticket; only the initial and final state of the ticket is relevant.
- It should also be possible to show some metrics (metric, KPI), e.g., the average time to solve an error.

In the second version of the ticketing ontology, it is important to include other relationships besides the class-class relationship and class properties. However, creating concrete instances is not necessary for exploratory OLAP prototyping.

The method used to define the concepts ("reverse engineering") is to integrate the concepts extracted from the data sources and the dictionary of the given field (ticketing incidents). The ontology thus obtained is a so-called user-centric domain ontology, i.e. it contains only the knowledge about the domain that is relevant for the data sources under investigation. The list of concepts (keywords) from the textual data sources was already completed for the previous prototype and was therefore available. To select the terms relevant to the ontology, a Python script was used to perform a semantic comparison between the keywords extracted from the textual data sources and the words in the domain-specific glossary.

I then filtered the results to identify the words in the domain glossary for which there is a high similarity index (mainly above 0.75) of keywords from the data sources. From the resulting words, I manually selected and defined the ontology's concepts (classes) and properties. Finally, I created the relationships based on the requirements. The result is shown in the following figure:

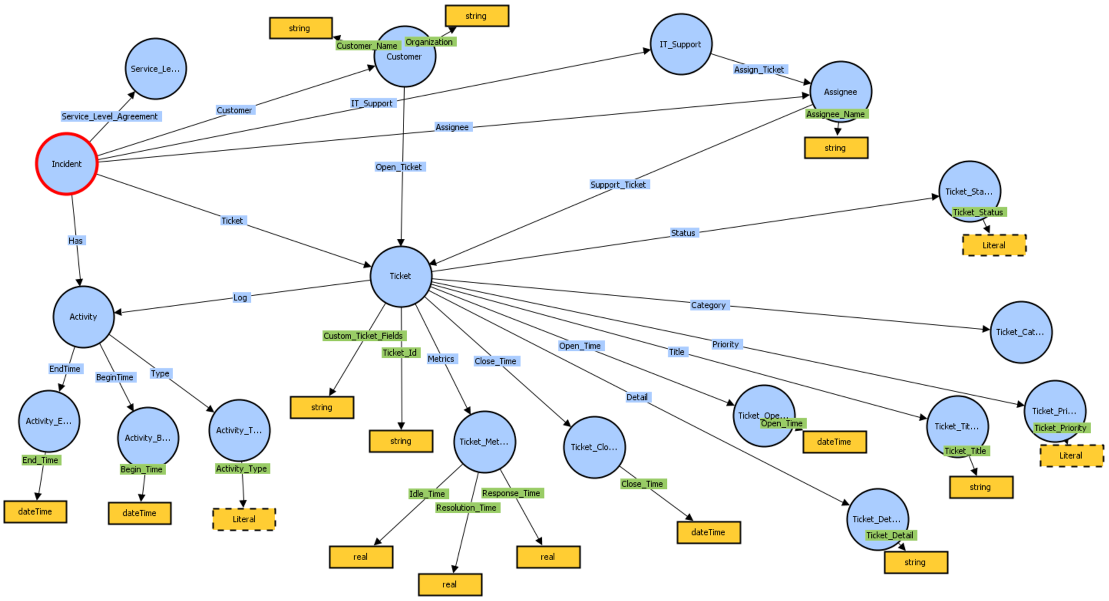


Figure 4: The second version of the ticketing ontology

3.3 Exploratory OLAP – first prototype

The primary goal of the prototype is to process text data from the ticketing system to produce the star schema proposed by Prasad.

The prototype is also expected to implement the following functions:

- Word processing operations (keyword search and tagging)
- The ability to learn keywords
- Creating an OLAP scheme using the star schema
- Create a dashboard to visualise the results.

Semi- or fully automated operation is not yet the goal. The prototype has been created in four steps based on a conceptual model based on, but different from, the Prasad approach:

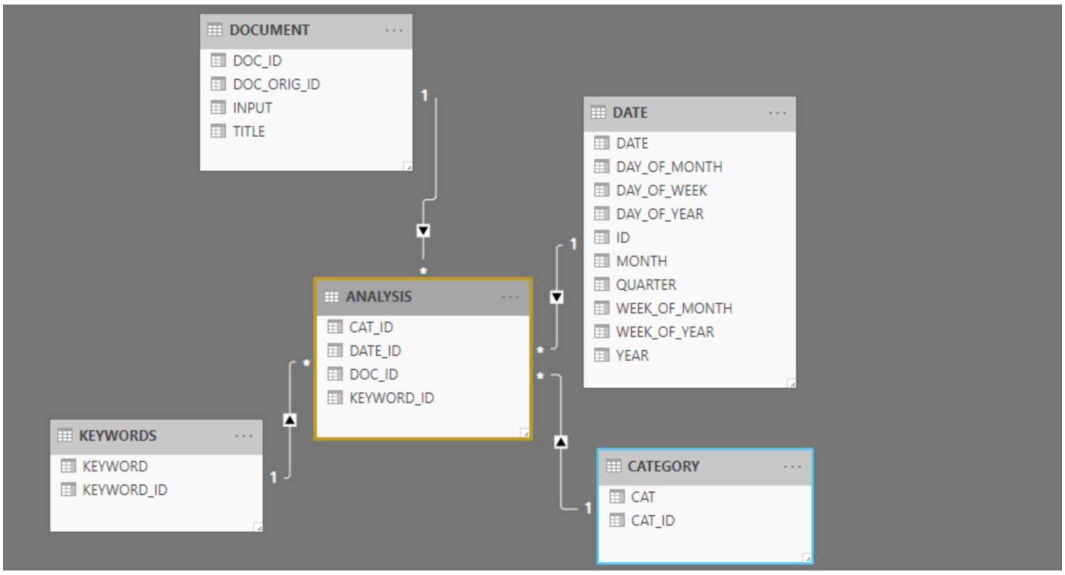
- Preparing data
- Carrying out text analysis
- Create a star schema
- Creating and testing OLAP cubes.

The following software was used in the implementation:

- Service Desk (ticketing) program for exporting source data
- Python 3.7 (Jupyter, Anaconda): for creating text processing routines

- MS SQL Server 2017 Express, SQL Server Management Studio and SQL Server Import and Export wizard to create a star schema
- Power BI Desktop to build and test the OLAP framework.

The prototype produced the following schema from the source data:



DOCUMENT
DOC_ID
DOC_ORIG_ID
INPUT
TITLE

DATE
DATE
DAY_OF_MONTH
DAY_OF_WEEK
DAY_OF_YEAR
ID
MONTH
QUARTER
WEEK_OF_MONTH
WEEK_OF_YEAR
YEAR

ANALYSIS
CAT_ID
DATE_ID
DOC_ID
KEYWORD_ID

KEYWORDS
KEYWORD
KEYWORD_ID

CATEGORY
CAT
CAT_ID

Figure 5: Schematic of the first exploratory OLAP prototype

The prototype is relatively simple to implement, but its applicability is limited by the predefined final schema.

3.4 Exploratory OLAP – second prototype

The development of the second type of exploratory OLAP is based on the conceptual framework shown in Figure 6.

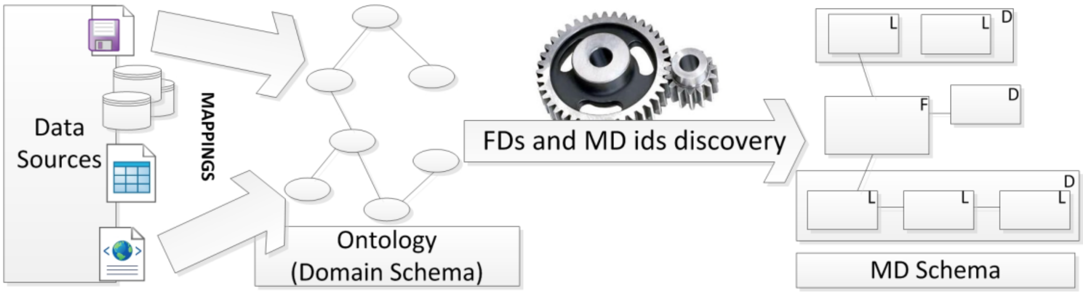


Figure 6: Ontologies in domain modelling

The first step in the development was data preparation. The source data relevant to the problem are:

- Error ticket data (such as ID, opening time, priority)
- Error ticket category data
- Details of the activities linked to error tickets (such as start date of activity, type of activity)
- Details of the IT staff carrying out the activities
- Details of the customer creating the error ticket
- SLA data

I have put the data, originally in Excel/CSV format, in a relational schema. The schema also contains the textual data (subject and detailed description of error flags).

Data from data sources are mapped to a reference ontology (see. 3.2). Mappings to the ontology are used to identify Functional Dependencies (FD) and Multidimensional Identifiers (MD ids). The MD schema is then generated by identifying the facts and dimensions. The latter is illustrated in the following figure:

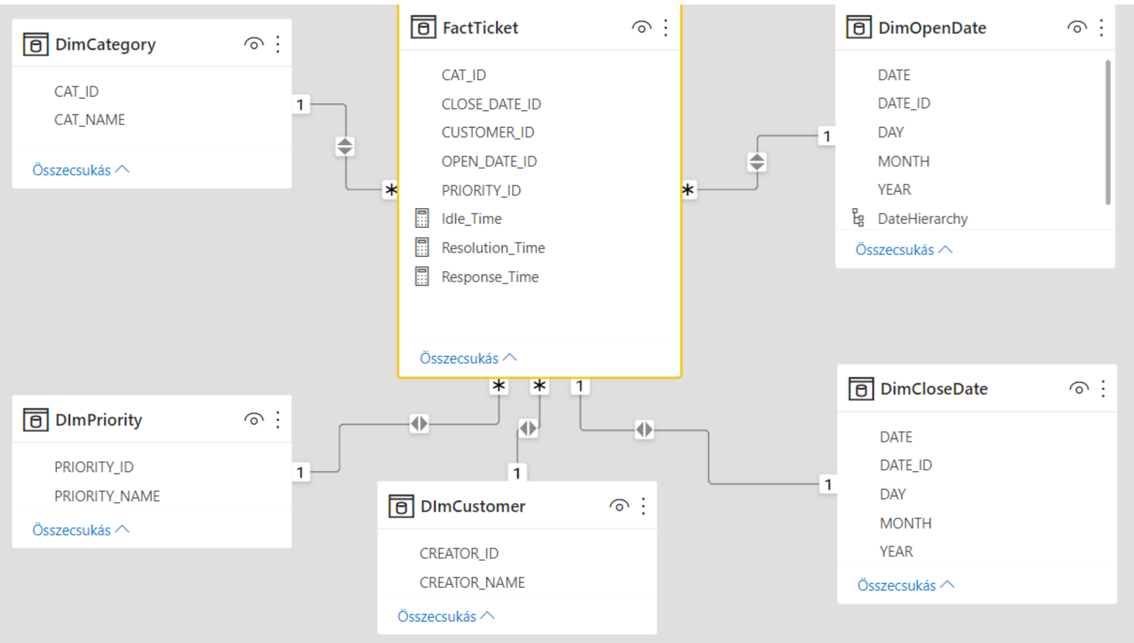


Figure 7: Schematic of the exploratory OLAP prototype 2

The resulting prototype can be compared to a "zeroth" dimensional model. In other words, it can provide a foundation on which to build an OLAP schema much more easily than starting the process from scratch.

3.5 Summary of the research results

The following table illustrates the research objectives and the results achieved.

Target	Result
Exploratory OLAP model testing.	I compared the models and then examined their feasibility and potential applications. Prasad's and Abello's models were feasible with the necessary modifications and simplifications. However, Ibragimov's model uses different tools from those preferred in research, and therefore its feasibility was not investigated. In practice, Abello's model is the most applicable of the three exploratory OLAP models. It allows the creation of an initial dimensional model starting from raw data.
Exploratory OLAP prototyping	I also made two prototypes using the ideas of Prasad and Abello. Each one uses data from a ticketing system.
Development of a ticketing domain ontology	Two versions of the ontology have been produced. The first is a two-level taxonomy well suited for the first exploratory OLAP prototype. The second prototype required further ontology development with new concepts, properties and relations.

Table 2: Summary of research objectives and results

3.6 The importance of research

My research can help:

- Ontology developers who need to create a new domain or application-specific ontology from scratch.
- Database (including ETL - Extract, Transform, Load) developers who are responsible for integrating textual data sources into the data warehouse.
- Data analysts who want to extract useful information from unstructured data sources
- For managers who want to analyse structured and unstructured data together.

The field of exploratory OLAP systems is still new and immature, so researching it is a particular challenge. However, the growing number of studies and articles related to exploratory OLAP demonstrates the importance of this topic. As far as I know, this is the first research on this topic in Hungary.

4 MAIN REFERENCES

Abelló, A. (2015). In *Using Semantic Web Technologies for Exploratory OLAP: A Survey.* IEEE. Forrás: https://core.ac.uk/download/pdf/41822169.pdf

Abello, A., & Romero, O. (2010). A framework for multidimensional design of data warehoses from ontologies. Data Knowl. Eng. Vol 69, no. 11.

Bánné Varga, G. (2012). In *Az adattárház készítés technológiája* (old.: 19-20, 23, 3.fejezet). Typotext.

Biemann, C. (2005). Ontology Learning from Text: A Survey of Methods. LDV Forum. Forrás: jlcl.org

Cuzzocrea, A., Bellatreche, L., & Song, I.-Y. (2015). In *Data Warehousing and OLAP over Big Data: Current Challenges and Future Research Directions.* International Journal of Business Process Integration and Management.

Gomez, A., Corcho, O., & Fernandez-Lopez, M. (2002). Methodologies, tools and languages for building ontologies. Data & Knowledge Engineering 46 (2003).

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge acquisition, Vol. 5 No. 2 .

Ibragimov, D., Hose, K., Pedersen, T. B., & Zimány, E. (2015). In *Towards Exploratory OLAP Over Linked Open Data – A Case Study.* International Workshop on Business Intelligence for the Real-Time Enterprise. Forrás: https://link.springer.com/chapter/10.1007/978-3-662-46839-5_8

Iqbal, R., Murad, A., & Mustapha, A. (2013). An Analysis of Ontology Engineering Methodologies: A Literature Review. Faculty of Computer Science and Information Technology, Universiti Putra Malaysia.

Klein, A. (2017. July). *Hard Drive Cost Per Gigabyte.* Forrás: https://www.backblaze.com/blog/hard-drive-cost-per-gigabyte/#:~:text=From%20January%202015%20to%20January,the%20cost%20of%20providing%20storage.

Kő, A., & Gillani, S. (2019). A Research Review and Taxonomy Development for Decision Support and Business Analytics Using Semantic Text Mining. *International Journal of Information Technology & Decision Making*.

Liang, X. (2018. február). Forrás: https://towardsdatascience.com/textrank-for-keyword-extraction-by-python-c0bae21bcec0

Liu, H., & Wang, P. (2014). Assessing Text Semantic Similarity Using Ontology. *Journal of Software, 9*(2), 490-496.

Luqi, F. K. (2002). An Introduction to Rapid System Prototyping. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 28*(9), 817-820.

Nebot, V., Berlanga, R., & Pérez, J. M. (2009). *Multidumensional Integrated Ontologies: A Framework for Designing Semantic Data Warehouses.* doi:10.1007/978-3-642-03098-7_1

Neumayr, B., Anderlik, S., & Schrefl, M. (2012). Towards ontologybased OLAP: Datalog-based reasoning over multidimensional. Proc. 15th Int. Workshop Data Warehousing OLAP.

Pâslaru-Bontaş, E. (2007). A Contextual Approach to Ontology Reuse: Methodology, Methods and Tools for the Semantic Web. REFUBIUM - FREIE UNIVERSITÄT BERLIN.

Phoebe Wong, R. B. (2019. October). *Everything a Data Scientist Should Know About Data Management.* Forrás: KDnuggets: https://www.kdnuggets.com/2019/10/data-scientist-data-management.html

Prasad, K. S. (2010). In *Text Analytics to Data Warehousing.* Kalli Srinivasa Nageswara Prasad: Text AInternational Journal on Computer Science and Engineering,. Forrás: https://www.researchgate.net/publication/49941856_Text_Analytics_to_Data_Warehousing

Revert, F. (2018. 12 17). *An overview of topics extraction in Python with LDA*. Forrás: https://towardsdatascience.com/the-complete-guide-for-topics-extraction-in-python-a6aaa6cedbbc

Romero, O., & Abello, A. (2012). Ontology driven search of compound IDs. Knowl. Inform. Syst., vol. 32.

Skoutas, D., & Simitsis, A. (2007). Ontology-based Conceptual Design of ETL Processes for both Structured and Semi-structured Data. *International Journal on Semantic Web and Information Systems, 3*(4), 1-24.

Sommerville, I. (2011). Software Engineering Ninth Edition. Pearson.

Vaishnavi, V., Kuechler, B., & Petter, S. (2004). DESIGN SCIENCE RESEARCH IN INFORMATION SYSTEMS. Eds.

Wieringa, R. J. (2014). In *Design Science Methodology for Information Systems and Software Engineering.* Springer.

5 LIST OF PUBLICATIONS

May 2022

JOURNAL ARTICLE

Géza Molnár [2021]: Ticketing Data Warehouse System Development: Challenges and Experiences In SEFBIS JOURNAL NO.14/2021 pp. 14-24

Géza Molnár [2022]: Decision-Making Through a Self-Service Business Intelligence Solution
In JOURNAL OF APPLIED MULTIMEDIA

CONFERENCE BULLETIN

Géza Molnár [2020]: An Implementation of Exploratory OLAP System Based on Prasad's Approach
In: AIS 2020 PROCEEDINGS, ISBN 978-963-449-209-2, pp. 89-92.

CONFERENCE ABSTRACT

Molnár Géza [2019]: application possibilities of semantic web technologies in exploratory OLAP
In OGIK 2019

Molnár Géza [2019]: The road from data to reports, or developing a management information system using Microsoft software - case studyIn
MAFIOK 2019, pp. 43

Molnár Géza [2018]: development of a management information system for an IT company - case studyIn
OGIK 2018, pp. 56

OTHER (BOOK EXTRACTS, BOOK CHAPTERS)