Sixty Years - and More - of Data Modelling

Hannu JAAKKOLA[a,1] and Bernhard THALHEIM[b,2]

[a] *Tampere University, P.O.Box 300, FI-28101 Pori, Finland*
[b] *Christian-Albrechts-University Kiel, Computer Science Institute, 24098 Kiel, Germany*

Abstract. Data (conceptual, data, information, knowledge) modelling is still the work of an artisan, i.e. an art in the best case, made by humans, because of the need for human intelligence. Data modelling is an essential part of Information System (IS) design, specifying how data is implemented as part of an IS. The principles of data modelling follow the evolution of IS development paradigms, and these in turn follow the progress of technological changes in computing. Although technology has changed a lot during the decades of commercial use of computers – since the early 1950s to now, close to 70 years – data modelling is still based on the same basic principles as decades ago. Or is it really so? Finding the answer to this question was the main motivation to start writing this paper. Since the future is more interesting than the past, we set our research problem to be "What are the challenges for data modelling in the future?". The reason for this is that we see some significant changes in the future in the data modelling sector which we wanted to examine. However, the future is a continuum of the past. The future cannot be fully understood without understanding the past. Humans also tend to forget the details of the past. Even the most remarkable innovations from the past have become part of the new normal. Consequently, at the beginning of our paper we look shortly at the progress of data modelling during the era of commercial computing. Our focus is on the recent past and we look at the technological changes that have been of key importance in data modelling in the role of triggers and enablers. To find the answer to our research question, we retrieved some recent studies handling the future of data modelling and analyse the challenges found in these sources. The paper is concluded by some future paradigms. In general, the big changes seem to be the growing importance of Artificial Intelligence (AI) and machine learning (ML) as its fuel. AI not only conducts algorithmic rule-based routines, it has learning capability, which makes it more intelligent and adaptable, and able to compete with human intelligence, even in data management tasks.

Keywords. data modelling, information modelling, information system modelling, information system design, data base

1. Introduction

Data Modelling defines the principles of how to create data models (physical, logical, conceptual, etc.) related to an information system (IS). It covers defining and determining the data needs of the target organization, and the goals of using the data. It is an essential part of IS design and creates the foundation of the whole development life cycle. Since the 1950s – the early phase of the commercial use of computers – the paradigms and technologies of IS development have changed radically. The same goes for the principles of data modelling.

---

[1] hannu.jaakkola@iki.fi

[2] thalheim@is.informatik.uni-kiel.de

A variety of terms - conceptual modelling, data modelling, information modelling, etc. – are used to describe the same basic element of IS design from different points of view and abstraction levels. In our paper we have adopted the term "*data modelling*" to cover the use of modelling techniques in different parts of the IS development lifecycle; thus it is an umbrella term covering conceptual modelling as well as data-oriented aspects in IS design at all levels of abstraction. The development of the model reflects the principles of the development paradigm. The final manifestation of the conceptual model (after several modifications during the development life cycle) can be seen in the data structure (database, software data structure) and functionality of the application.

The aim of this paper is to consider the changes in data (conceptual, information, …) modelling over the decades, with the main aim to look at the current situation and the future. We look backwards to a paper from the late 1990s, which the second author of this paper co-wrote (Chen, Thalheim & Wong in 1999) [3]. If we examine the era of databases and the conceptual modelling related to it, that paper was written in the *growth era* (technology was maturing towards the growing use of relational databases) of database management systems (DBMS). It is reviewed briefly in Section 2 of this paper to provide an opportunity to compare "how the future was seen in 1999", "what really happened," and how we see the future today.

What has been the progress of data modelling over the decades? Foote's paper in 2017 [7] answers this question; it is slightly modified in the following according to the authors' subjective interpretations. It separates four (overlapping) phases covering the era of systematic data modelling from the 1960s to 2020. These are briefly introduced in the following paragraphs.

The development of DataBase Management Systems - DBMS (from roughly the 1960s to 1999). In this period a variety of implementation architectures existed: hierarchical, inverted list, network, and at the end of the period the idea of object-oriented database management systems - OODBMS. The last mentioned emerged in parallel with the coming of the object-oriented programming paradigm and Smalltalk. The first commercial database system, the integrated data store (IDS), was introduced in 1964 by Charles Bachman. Most of the "OODBMSs" were implemented on top of an existing architecture without having the pure properties of an object model; these extensions were also typical in the following era (RDBMS).

The birth of Relational DBMS – RDBMS (started at the end of the 1980s/early 90s). Edgard F. Codd introduced his data model based on relational calculus in 1969/1970[3]. At the beginning, it was the focus of a wide researcher community, and the first experimental implementations occurred quite quickly. Although the first "commercial" implementations (Multics) was available in the late 1970s, RDBMSs were not adopted for real commercial use until the 1990s. In addition to the logical structure of databases it brought a user-friendly way to define database contents and to implement queries in it the Structured English QUery Language (SEQUEL), which was later abbreviated to the more commonly known SQL. This also gave rise to the commonly used name for this

---

[3] The starting point of relational model and RDBMS is based on the following two articles:

Codd, E.F (1969). Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks, Research Report, IBM.

Codd, E.F (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM. Classics. 13 (6): 377–87. doi:10.1145/362384.362685. Archived from the original on 2007-06-12.

era: SQL Databases. In IS industry practices this era is continuing, but to an increasing extent it is mixed with approaches providing additional functionalities for data handling.

Specialized DBMSs and their use in data analysis (started in 1990s and continuing in its varying forms). Foote refers to this era as the support for online analytical processing (OLAP) but in practice it covers a variety of forms to apply managed data for business intelligence applications, data mining and a variety of (big) data analytics. OLAP itself is an approach that allows multi-dimensional handling of data. To allow powerful processing the data must be organized to support this (multidimensional) method of data handling. In practice these are different applications of relational and NoSQL? databases. Currently the frontier technologies support effective handling of mixed structured and non-structured data in (big) data analytics. One of the frontier technologies today is *Apache Hadoop*[4], which consists of a data storage part (distributed file system (HDFS)) and a processing part (MapReduce programming model) supporting parallel data processing. We see this as a continuum of the progress that began with more simple "intelligent" data analysis, despite it being considered as the start of a new era of (specialized) data processing.

NoSQL (started in late 2000). The progress of the NoSQL data model was accelerated by the growing need for big data analytics. NoSQL provides a mechanism for handling data that no longer follows the structure of relational databases. The source of such data may be in large masses of documents and in the data streams (pipelines) of real-time WWW applications. In this kind of context, the role of data modelling faces new challenges; traditionally the data model is based on a structure – with components and the relations between them presented in a selected framework. In NoSQL data modelling, with such data stores, the model is embedded in the code, instead of traditional data models.

From this, we continue in the following sections of this paper by providing a look at the current situation and the future. The purpose and the role of data modelling have varied over the eras presented above. There is a clear link between the *models*, the *IS development paradigms* and the *tools used* in different life cycle phases. The evolution of programming languages has had a strong influence on shaping the practices of using these languages, as well as on the tools used to build the IS models. An additional aspect worth noticing is the capabilities of computing environments. Lack of processing power, computer memory capacity, the structure and capacity of mass memory are typical reasons for delays in adopting new technologies. What is the reason for RDBMS becoming a commercial success some 20 years after its theoretical foundations were introduced? According to Jaakkola et al. in 2017 [15], computer performance (memory, processing), mass memory capacity and networking (technology, speed) are triggers for adopting new paradigms and technologies. Progress in storing and handling data, as well as in data modelling, has had the same change triggers.

The aim of this paper is to consider the changes in data modelling over the decades, with the focus on the future. The history and path of the progress were dealt with above. We continue in Section 2 by mirroring the current situation with a view from 1999. Section 3 reports our findings related to the future of data modelling based on a "literature mapping" study. In Section 4 two future "paradigms" are discussed. Section 5 concludes the paper.

The *research problem* of this paper can be crystallized as follows:

What are the challenges for data modelling in the future?

---

[4] Apache Hadoop – see https://en.wikipedia.org/wiki/Apache_Hadoop.

2. How the future was seen in 1999

2.1. View of the future from the past

The work of Chen, Thalheim & Wong in 1999) [3] summarizes and abstracts the papers of a 1997 workshop published in a post-proceedings form. The papers in this volume were written by several researchers, mainly from the conceptual modelling community and especially by contributors to ER conferences. The paper we reconsider summarizes the entire post-proceedings, which were the basis for subsequent compilations by various authors, e.g. Embley & Thalheim in 2011 [6] and Thalheim in 2000 [25]. Here, however, we will use the summary from 1999 and then discuss the differences with the current state of the art.

The future and important directions are divided into the following in this paper:

- *Active modelling*: Modelling should become a continuous process. Once a model has been implemented within a system, the improvement of the system and the database code should be accompanied by a revision of the model. Models need continuous improvement due to changes in the applications, changes in the technology, integration of systems, extension to areas in which databases can also be successfully used, the quality of data in use, modernization of DBMS, and size of code and models.
- *Relationship between natural languages and conceptual modelling*: The semiotics of natural languages has great potential for improvement of modelling languages. At the same time, a sophisticated methodology similar to literal would improve modelling. Modelling languages should become as powerful as natural languages.
- *Conceptual framework for sharable information services*: Technology may enable the development of global information services that can be called in a variety of exchange facilities and protocols. The information explosion has to be met by the provision of small and effective services for the needs and demands of users.
- *Relationships between the real world and the software world*: From the viewpoint of normal users, computer systems are mainly software systems since hardware has become encapsulated and interfaced by software. Models of the software world and its activities allow us to handle, understand and evolve it properly according to the needs of the real world. These models should also support the cross-infusion of the software world and the information systems world.
- *Conceptual model as the basis for interoperability*: Systems should be transitioned into harmonized systems that run in a consistent and seamless manner instead of using a "let's-glue-pieces-together" approach. Models and conceptual models might be a top-down solution.
- *Conceptual modelling as the first step for application engineering*: Systems are continuously evolving. New users need their own interfaces which should be based on proper understanding of the entire metadata. Modelling should become a continuous activity which includes management of change, extension of scope, and utilization as well as integration into existing infrastructures.

- *Global communication*: While global data sharing is going to take place, sharing knowledge and high-quality information remains a key concern. Aspects of information and knowledge management have to be included.
- *Human knowledge integration*: Human knowledge can be understood from three perspectives: normative, situational and existential. Its integration is the ultimate goal for information technology development. Such systems must support inference, justification, question proliferation and answer generation according to the profile of human beings.
- *Pragmatics of conceptual modelling*: Modelling languages should be extended in such a way that they allow the development of a conceptual data model at a high-quality level. At the same time, they should correspond to improved object-relational database management systems.

The list of trends remained incomplete at that time. It would never be possible to provide an exhaustive list. However, it is a survey of most of the major trends as seen in 1999.

2.2. Continuing - Some trends since 1999

The architecture of information systems became separated into a frontend system and a backend system. The frontend system can use several data model paradigms. The backend system is typically built as an object-relational system. The systems around 1999 followed a global-as-design approach with a global data model for the backend and derivable views defined on this global model for the backend. With the sophistication of the frontend, this decision has been revised, especially for web architectures and distributed databases. Data models then became *model suites* (by Thalheim in 2008) [26]. This model suite approach also allows the co-designing of data models with business process models.

The experience gained with the maturation of classical applications has been summarized, generalized and abstracted in reference models. These reference models can be used in new projects as a starting point. Universal applications use such generalized data models for the generation of a specific model

Information systems became the kernel of data-intensive applications with the advent of web infrastructure. These systems are distributed and integrate a number of viewpoints. They use a variety of data models and are continuously revised and modernized. The two main branches of programming (in the small and in the large) have been extended to programming in the web by partners that only partially follow a common strategy. Typical programming in the web is not governed by any guru. Projects co-evolve without tight coordination, integration or harmonization.

Data modelling has become an issue for almost all disciplines of science and engineering. While in the past databases were developed by specialists, laymen have now become systems developers. At the same time development and research have become more interdisciplinary. Collaborations that use common data are not yet supported by the co-development of the corresponding databases.

Databases have become huge in some cases. At the same time, data models have become *more complex*. Big databases are highly distributed and run in parallel. Data replication has enabled controlled redundancy.

2.3. Analysis and the state of the art compared to 1999

Let us now structure our trend observations and experience gained in many projects into three groups: directions that have resulted in almost completed research, directions that are still research tasks but achievable, and directions that proved to be unachievable:

(1) Successful research[5]:
- The linguistic foundation of modelling has become a common framework in data modelling.
- Modelling methodologies have reached maturity at ISO 33001 level 2 or 3.
- Distributed and federated database systems have become well-supported by modelling.
- Ontologies can be used for harmonization of viewpoints for business users.
- Service architectures and provision have become the standard.
- Question-answer forms combined with input-output forms provide an initial solution to the global communication problem.

(2) Issues for continuing research:
- Data modelling is still an art and has not become a culture. Each research and development group follow their own approach. The integration and harmonization of the variety of approaches hinder data integration in such collaboration projects.
- Model transformation and active modelling has not yet become state-of-the-art.
- Co-evolution of database systems and models is nowadays tackled by models-as-programs.
- Continuous modelling in dependence on changes is still a major lacuna.

(3) Tasks that have not been continued:
- Modelling languages will not achieve the expressivity of natural languages.
- Pragmatics of modelling did not receive common treatment in dependence of users.
- Research on object-oriented database systems became the source for object-relational system technology.
- Interoperability of data-intensive systems and applications can only be provided for greenfield application development and cannot be supported in migration and evolution scenarios.
- Knowledge and information management inside a database system is not supported.

We will return to these topics to some extent in the following sections.

3. How the future is seen in 2020

3.1. Challenges to data modelling in its current form

We know of more than two dozen data modelling languages. For instance, the handbook of conceptual modelling by Embley & Thalheim in 2011 [6] discusses almost a dozen of them. Half a dozen of these languages have found their way into practice with the most

---

[5] Obtained from a large reference list for this body of knowledge and research.

prominent UML class diagrams and various extensions of entity-relationship modelling languages. Data modelling based on such languages faces several challenges:

- The challenge of neglected foundations: Most languages are purely syntactically defined. Some of them exist only as graphical languages. UML class diagrams have several interpretations. Data modelling languages mainly provide notions with one lexical meaning of a word in a given culture and organization, e.g. the term "Person_Name" has many different meanings. Part of the foundation is thus to enhance a data model with concepts from a conception space. The syntax orientation of constructs limits semantics to local semantics for each construct. Full semantics of a data model are not derivable.

- The challenge of model usage beyond the description purpose: So far, data modelling has eroded to descriptive modelling. Data models are generally considered to be inspiration models for database system developers. The model will not be used as a prescription for development since many essential elements that are needed for development can be interpreted in various ways or are missing. The model will become an historic document for system documentation that cannot be and will not be maintained or modernized with respect to changes.

- The challenge of non-agreement in the community of practice: The (conceptual) data modelling communities have not agreed on a common notion of the (conceptual) data model in 45 years of its investigation in conferences, international bodies and projects. There are more than threescore different notions (Thalheim in 2018) [27]. Three ER conferences (2017, 2018 and 2019) incorporated workshops for the central issue of developing a commonly accepted notion of the conceptual model.

- The challenge of language simplicity: The entity-relationship model in its current widely used form uses rather simple language constructs. The main sentence constructor follows the SPO paradigm, i.e. two or more kinds of object are associated by a relationship type. English language has 25 sentence patterns and is thus far richer. The ER language is not cognitively complete. It allows only representation of three out of six cognition dimensions. Advanced extensions of the ER modelling language struggle with the sixth dimension.

- The challenge of the logical separation of syntax and semantics: Natural languages use wording that have a syntax, semantics and pragmatics. A word field combines all three semiotic dimensions. Computer science uses rigid layering and omission: syntax first, semantics later on the basis of the syntax, and pragmatics is not of interest. This separation leads to flat models, i.e. models whose constructs do not have an inner structure and an inner semantics, e.g. "Person_Name" with its specific nature.

- The challenge of concentration on one abstraction level for data: Data often come from various sources with their quality, i.e. micro-data, cleansed meso-data, aggregated macro-data. OLAP cube applications suggest that it might be essential to drill down from more abstract data to less abstract or to roll up from fine granular data to less granular. Moving though data abstraction layers requires understanding of the meaning of the data and of the attributes at their abstraction layers (Molnar & Thalheim in 2007) [22].

- The challenge of structure-behaviour separation: A common understanding of a data model is that a data model defines the "data structures, data operations, and so forth, that together make up the abstract machine with which the user interacts" (Date in 2005) [5]. Data models thus concentrate on the data structure, constraints defined on

those structures and, to a certain extent, operations which can be compiled from higher-order functions in order to support computing on these structures. Structures and constraints may however be dynamic (Thalheim in 2000) [25]. Furthermore, performance support requires auxiliary structures, application-governed management of the data and structuring. The theory of normal form which is today outdated has been developed for data restructuring according to performance and behaviour.

- The challenge of M0-M1-M2 layering: Natural languages allow definitions and construction of complex items inside the same language. They are somehow "flat". OMG proposes to separate things at the M0 layer from their representation at the M1 layer (termed the model layer), to separate representation from the language layer (M2, called the "meta"model layer), and to separate the language layer from the foundations layer (M3, called the "meta""meta" layer). This leads to a difficult treatment of IsA hierarchies together with class hierarchies (`a student is a person', `the class of student is a sub-set of the class of persons'). It is often inappropriate, e.g. the difference of various variants Is-A and of Is-Instance associations is not clear. Multilevel data modelling (Frank in 2014) [8] crosses these layers.

- The challenge of global-as-design: Data modelling assumes that a global schema is specified at the conceptual and implementation layers (which can be separated into a logical and a physical layer). The external layer concentrates on defining views which represent the viewpoints of business users. The three-layer architecture was appropriate for mainframe systems. Local viewpoints have to be represented by views. Co-design of data and business processes becomes very difficult since the business processes are based on a local-as-design paradigm. It results essentially in an impedance mismatch between the two extreme modelling strategies.

Despite these challenges there are many other lacunas, unexplored scientific issues and hidden culture problems.

3.2. First steps from the past to the present

At the end of section 1 we defined the research problem of this paper: *What are the challenges for data modelling in the future*". We also concluded that data modelling principles cannot be distinguished from the *general trends* in IS development, which in turn influences the IS development environment, including modelling and implementation tools. We also emphasized that progress in tools and environments depends significantly on progress in the triggering technologies. What are the leading technologies defining the trends in IS development? Some of these are listed here, referring to a variety of sources: virtualization and cloud, XaaS (something as a service), opportunity to use parallel (cluster based) processing, growing importance of system interoperability and commonly used open interfacing solutions (APIs, e.g. REST), integrating (big) data analytics to legacy ISs, the transition from plan-driven development towards agile development processes, distributed development and DevOps, the growing importance of application development for affordable devices (mobile terminals) using development platforms for them, the scalability of IS to be used in varying platforms, the use of the internet as a delivery mechanism, new kinds of revenue logics (transfer from licenses to pay-as-you-go), replacement of the customer by a faceless user (loose, if any, supplier-customer relationship), the importance of openly available components and (sub)solutions, growing value of the (external) data, "every product is a software product" – the era of IoT and embedded software. This progress

was examined from the point of view of the software business in a paper by Yrjönkoski et al. in 2019 [29].

Going back to history, the basic principles of conceptual modelling (e.g. ER models) were developed in the period of very traditional IS development based on the waterfall type of life cycle, a strong client-customer interface and the use of IS in a customer-managed operating environment. The progress in modelling has gradually followed the progress in the characteristics of ISs and their development process by adapting to the changes. One remarkable step towards the real needs of the ICT sector was the unification of modelling practices (languages) by introducing the unified modelling language (UML)[6] in the middle of the 1990s. Closely related to this, an iterative software development process – the rational unified process (RUP)[7] - was introduced to replace the traditional waterfall model. An important step in the adoption of UML was introduced by Kruchten in his 4+1 view software architecture model [8]. It integrates five views of IS models (based on UML) and highlights the principle of *multiple (consistent) models* in the modelling of IS during its development life cycle. Where traditional ER modelling focused on the *static structure* of the IS (concepts and their relations) leaving modelling of its dynamics (behaviour) to be done separately, UML highlighted and supported the need to model IS's dynamics as an integrated part of the whole. UML replaced the scattered set of models with an integrated set of modelling languages and transferred IS development from the era of structured development to the object-oriented (OO) era.

When analysing ER conference proceedings since 2010, we observe an increasing number of papers that solve some of the modern application problems by developing enhancements for the current technology. The data models proposed are essentially extensions of classical data models. Typical data modelling examples are proposals oriented to application challenges such as the following: (big) data analytics, data warehouses and data marts, data exchange among distributed systems, heterogeneous data ensembles, weakly structured massive data, evolving and migrating systems, clients such as search clients, web-based systems, support of the entire development cycle starting with requirements, graph-based computation, support for advanced systems such as geographic information systems, XML representation, challenging applications such as genomics and life sciences, enterprise data integration, service systems, research collaboration, support of model-driven approaches, privacy, not-only-SQL (NoSQL) systems, and the integration and collaboration of systems. At the same time, ontologies have become widely accepted as a modelling technique.

---

[6] OMG (1997). UML Specification version 1.1 (OMG document ad/97-08-11)". Retrieved from https://www.omg.org/cgi-bin/doc?ad/97-08-11 on January 29th, 2020.

[7] Wikipedia (2020). Rational Unified Process. Retrieved from https://en.wikipedia.org/wiki/Rational_Unified_Process on January 29, 2020.

[8] Kruchten, Philippe (1995). ViewModel of Software Architecture. IEEE Software 12 (6), November 1995, pp. 42-50. Openly available and retrieved from https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf on January 29th, 2020.

3.3. The current trends – being prepared for future needs

We approached this topic by conducting a simple literature study, implemented as an Internet search, applying (freely) the principles of the mapping study approach by Pettersen in 2008 [23]. From the results we selected a dozen articles for further investigation. The study gave evidence of our expectations: this topic has not been dealt with widely by researchers in scientific writings, but more by well-known analyst companies (Gartner, Forbes, McKinsey etc.) in their studies. In turn, these have been analysed in numerous articles, which were finally selected as the sources of our summarizing analyses. This was done in a critical manner augmented with interpretations / concluding remarks / subjective experience-based opinions.

The article by Ghosh in 2019 [13] names *Artificial Intelligence* (AI) and *Machine Learning* (ML) as the key factors of the current dynamic in data modelling. She lists some remarkable changes that are discussed in the following paragraphs - interpreted and integrated by the authors.

Data Science. Significant changes will occur in the division of jobs and duties. Ghosh refers to the study of Gartner in 2019 [12], which predicts 40 per cent *automation* in data science tasks (augmented analytics, with augmented data analytics as part of it). *Model analytics* decreases the opportunity for human error and increases the quality of models, whereas *automatic model generation* decreases human work. Data analysis tasks and traditional practices in master data management (MDM), metadata management and data governance can be automated by AI/ML. More power in (business) data modelling will be offered to *citizen data scientists;* analytics will be driven by automated ML models. *Two types of data models* (views of the same data) are needed – one for professionals and one for citizen users on a plug-and-play basis to execute quick solution type tasks.

Non-structured data and pipeline-driven data integration. The growing importance of non-structured data is a reality. New database technologies - NoSQL/non-relational databases, data lakes – are becoming part of ISs. In applications, instead of on the traditional ETL (extract, transform, load) basis, data is transferred in a pipeline-driven manner as a continuous stream from a variety of sources (IoT, social network data streams, etc.), which is not common in traditional ISs.

Traditional DB technologies are not disappearing – instead there is a hybrid mix of technologies. In spite of the rise of non-relational databases, the hybrid mix of database technologies maintains the importance of traditional data modelling - relational systems are not disappearing and will have a more important role than ever as part of the data infrastructure. Although "algorithmic intelligence, self-describing data formats and standardized models" have taken some of the labour out of data modelling, the newer database technologies have initiated a whole new set of challenges for data modellers. These *hybrid systems* are challenging for data modellers, who must be familiar with the beneficial use of opportunities that AI and ML provide for their traditional tasks.

Robotic Process Automation (RPA): The rise of RPA gives an increasing role to *business process modelling*. Combined with the cloud trend - SaaS (software as a service) solutions and ML (MLaaS – machine learning as a service) -, this creates challenges in data governance issues, which may remain *external* to the organization (externalization).

Transition? from problem specific to problem area specific solutions. In data modelling there is a transition from problem specific to problem area specific instruments. This is analogous to the progress of ISs, which are no longer single systems but complicated *systems of systems*; see the conference Paper of Boehm in 2006 [1]. At

the conceptual level we can also create an analogy as the transition towards *concept systems* instead of concepts per se – a kind of hierarchy. Interoperability-related requirements will of necessity move towards *commonly used standardized data structures and models*. Modern systems are to an increasing extent built with the guidance of existing frameworks – *platforms and ecosystems*, which leads to the *higher abstraction* of the models. The role of the ecosystem owner may create links and dependences that are not fully under the control of an individual organization. The performance expectations of modelling are expanding to new areas – to the ability to manage complex IS network structures, their interfaces and interactions, for example. Data governance is becoming more problematic – there are an increasing number of *unknown* data governance issues, due to cloud-based (XaaS) solutions. Cloud dominance affects data structures and needs support from a complicated data management ecosystem.

In summary, it is easy to see the growth of modelling complexity, the transfer of data (modelling) related tasks from professionals to end users and AI to support human work. At a basic level, data modelling remains as it has always been, but in practice a lot of new challenges are appearing.

The paper by Knight in 2020 [21] emphasizes that in the future (and now): "Modeling will need to do more, faster, with less". *Faster* means speed, *more* refers to the growth of the amount and complexity of data, and *less* indicates the need to lower the costs. Information must be processed faster, the volumes handled are greater, having their source in a *multitude of systems*, and a certain dynamic (readiness to react quickly to changing needs) is expected. AI will have a growing role in *automating business* tasks, even in the data management sector. The work of data modellers will transition to solving a decreasing amount of *complex business problems* in a short timeframe. The aim of companies is to optimize the data modelling activities to be able to manage them at lower cost; this includes the adoption of AI and the ML within it. Knight emphasizes four important trends in data modelling in 2020: just-in-time data modelling, better automation and machine learning, more widespread use of data modelling, and focused data modelling. These are briefly explained in the following paragraphs.

Just-in-time data modelling. This refers to changes in the IS development process. In agile development, even *agile data modelling* is expected. Because the agile philosophy is based on minimal (sufficient) design, even modelling activities are based on the *on-the-fly* principle. Just-in-time data modelling deals well with varying *mixed (hybrid) data structure*s – relational, NoSQL, dimensional[9] (data warehouse), master data[10], graph database[11], etc. The basic work of a data modeller remains the same – documenting relational (and other) structures. The work will happen to an increasing extent in the *cloud* because of its easy scalability, availability and low cost. This will also encourage use of *larger (and more complicated) applications*. Thanks to accelerating data transmission, data modelling is becoming "portable" – higher volumes of data from a wider range of systems and easy access to data models from anywhere by the relevant stakeholders.

Better automation and machine learning. Knight follows the same theme as Ghosh (discussed above) a year earlier. A lot of tasks will be automated based on the AI and ML built in the tools and development environments. The tasks following predefined

---

[9] https://en.wikipedia.org/wiki/Dimension_(data_warehouse)
[10] https://en.wikipedia.org/wiki/Master_data
[11] https://en.wikipedia.org/wiki/Graph_database

processes will be replaced / supported by RPA-type solutions, which transfer human routine work to machines. Therefore, the duties of data scientists / data modellers will be changed to more complex (non-routine, non-algorithmic) tasks.

More widespread use of data modelling. This confirms the trend already discussed above. Put simply – the role of *citizen data scientists* is growing because of the higher intelligence of the plug-and-play type (flexible) analysis and modelling tools.

Focused data modelling. Data modelling will become more focused; this is because less comprehensive general-purpose data models are needed for use with hybrids of DB architectures and data streams. The scope of the data models will *cover larger* amounts of enterprise knowledge and technical details.   According to the IDC study and prediction[12], *unstructured data* will account for 80% of enterprise knowledge by 2025. The focus of data management will move to a *higher level of abstraction*, the management of knowledge. The models will be used by an *expanding audience of stakeholders* and must be customized to cover the needs of specific purposes. Data models must be organized by *multi-level concerns* covering business level, solution level (logical level concerns) and implementation, in some cases providing additional viewpoints.

Related to the other papers selected for our analysis, the findings seem to repeat the topics handled above. This is evidence of penetration (as a stop rule in the literature study) in the coverage of the topics handled. Therefore we will satisfy ourselves with considering these papers and articles briefly in the following paragraphs, by highlighting some new findings.

Frisendal (in 2020) [10]. This paper points out the importance of semantic *graph-based modelling* techniques and graph databases as an important category of NoSQL databases. Semantic graphs allow the modelling of semantic relationships between concepts. The ISO/IEC JTC1 SC32 WG3 database languages committee (SQL committee) has decided to add support for property graph queries within SQL. Graph databases represent data and its relations as a set of nodes; this "*semantic network*" supports data analysis type tasks. Frisendal also refers to traditional "*semantic web*" technologies (RDF, OWL etc.), which support the knowledge graph idea and have been adopted by some vendors.

Cagle (in 2018) [2]. This article focuses on large-scale enterprise models with large vocabularies to manage. He provides RDF[13] (resource description framework; semantic web) as a tool for such enterprise-level modelling.

Roussopoulos & Karagiannis (in 2009) [24]. This paper provides an interesting historical perspective on the progress of conceptual modelling. The last section of the paper focuses on the future. Conceptual modelling is seen as a continuous process. The web environment is seen as a constantly changing environment. The user must have the opportunity to *adapt the conceptual schema* and to track his interactions with the Web and SOA (service oriented architecture). This approach is called "*schema-during (SD)"*, in which the process of conceptual modelling is on a continuum with the operations in the database. Maintenance of the model needs a *user-friendly click-language* tool for *end-user conceptual modelling*. This approach is convergent with the growing

---

[12] https://www.datanami.com/2019/08/23/enterprise-search-in-2020-and-beyond-5-trends-to-watch/; See also https://solutionsreview.com/data-management/80-percent-of-your-data-will-be-unstructured-in-five-years/.

[13] https://www.w3.org/RDF/

importance of citizen data analysts – more responsibility is being transferred from professionals to end users.

Harper (in 2019) [14]*.* This article emphasizes the change from relational database towards *heterogeneous data sources*, which are purpose-built for real-time use (IoT, data from a variety of sources based on pipelining). *Adaptable data modelling* practices accelerate the modelling process, especially when including diverse sources. It is based on the use of tools supporting *codeless models* (model-driven techniques) and *visual model building*, which can be used for *collaborative data management*, especially when the data comes from a wide variety of different sources.

Wells (in 2016) [28]*.* This article provides an interesting view of data pipelines, which lead to *reverse (backward) data modelling*. The traditional model is built on a top-down basis – conceptual modelling -> logical design -> physical model - to get the structure to store the data. In pipeline-based data sources, the data already exists and is stored without the opportunity to change its physical structure. The logical model is then deduced in a *reverse modelling* process by starting with fields, then trying to deduce the items that they describe and finally the relationships between those items. ER models are not able to handle many constructs that are typical of *NoSQL databases*. Wells lists the following: many-to-many relationships, multi-valued attributes, embedded arrays, associations implemented without foreign key relationships.

Guess (in 2020) [11]*.* This article handles the role of *low-code programming*. Although this phenomenon is a little beyond the scope of the debate on data modelling, it confirms the transfer of work from professionals to end users, as in the "citizen data analyst" and "no-code" data modelling discussed above. Low-code development is supported by a *platform* that provides an environment for the programmer to use when developing an application using a graphical user interface. The platform typically supports application area-oriented development and may need additional coding for special situations. Typical applications suitable for low-code development cover *databases*, *automation of business processes*, user interfaces and *web applications*.

David (in 2017) [4]. This paper largely follows the same approach as the one we followed at the beginning of this paper. It lists milestones that have had / will have high importance in IS development, including aspects related to data modelling. All the aspects included in the paper have already been covered in the discussion above.

Kiyoki et al. handle in their paper [20] semantic modelling. It is based on the *Mathematical Model of Meaning* (MMM), in which an orthogonal multidimensional semantic space is created, and used for semantic associative search. Retrieval candidates and queries are modelled - mapped - onto the semantic space, and the semantic associative search is performed by calculating the correlation of the retrieval semantic space. In [20] the method is applied in the multi-spectrum images used for environmental analysis, covering spatial, temporal and semantic axis in the analysis and modelling.

We have analysed a set of articles / papers dealing with data modelling, and a slightly wider scope of IS modelling. The concluding message is that the future of modelling work – already visible in the present – faces a lot of challenges caused by the breakthrough of new types of data sources. On the one hand automation helps the work, but on the other hand the remaining work needs new updated skills. The role of the end user will grow in IS development and in the use of data sources. The essence of data is changing towards built-in semantics and characteristics of knowledge.

3.4. Summarizing the findings – discussion

We apply the categorization of technological changes introduced by Freeman & Perez in 1988 [9] in the classification of the changes in data modelling; the changes are divided into four categories (discussed in detail by Jaakkola et al. in 2017 [16; 17]):

- *Incremental changes* appear continuously in existing products and services (continuing the existing trend); these accelerate the existing change in the existing trend.
- *Radical changes* appear when new research findings are applied in products to transfer their properties or performance to a new step or cycle; at a certain moment there is an upward shift (break in the trend) caused by the innovation.
- *Changes in technological systems* are caused by combinations of several incremental and radical innovations in societal and organizational systems; these changes provide the means for increasing competitiveness and fast growth of productivity for early adopters.
- *Changes in paradigms* are revolutionary and lead to pervasive effects throughout the target system under discussion. They indicate permanent changes in societal systems, daily ways of operating and business models that replace the old ones.

Although the ideas of Freeman & Perez were developed to analyse technological changes in society, the principle can be applied in the case of a single technology, i.e. data modelling (as one of the technologies in IS design). We have included one additional category in our analysis below - *triggering and enabling technologies.* These are changes that do not have a direct connection to data modelling itself, but act as an enabler to make the changes necessary or possible.

In technology analysis it is worth noting that the situation does not remain stable. New change-driven innovative technologies tend to become the *new normal* over the course of time. This has happened for instance to UML and the object-oriented approach. These were revolutionary (radical, system change, paradigm change) at the time of their appearance, but no longer cause any significant changes.

In sub-section 3.1 we discussed the challenges related to the practices of data modelling and the languages used. The source of the challenges lies in the expression power of the modelling tools (languages), which typically are *semi-formal* by nature. They have high (but limited) expression power, but lack precision, which leaves space for interpretation. The beneficial aspect is high (visual, graphical) readability from the point of view of a heterogenous set of interest groups. The challenges have solutions, which are *incremental*, and are at most *radical* in character.

Sub-section 3.2 focused on the near-past phenomena that mainly have characteristics of *triggering and enabling technologies* and *radical* changes. ISs are transitioning quickly and powerfully towards cloud-based implementation and service architectures (XaaS, SOA). This has caused remarkable changes in data modelling. Modelling tasks are "*outsourced*" to the responsibility of the cloud data management and to an increasing extent an individual organization has to adopt the dominance of *"external" data models* and adapt their own solutions within these. We listed the following trends, which belong to the category of changes in technological systems:

- the growing importance of system interoperability and interfaces (API),
- transfer from plan-driven to agile development processes,

- the use of distributed collaborative (concurrent) development environments (Devops),
- transition towards platform and ecosystem dominance, and
- the growing value of external (non-modellable, non-structured) data produced by IoT and a variety of other external sources.

We are also losing the power of direct supplier-customer communication, which is being replaced by the development paradigm "faceless supplier to faceless client" caused by the growing demand for service-based systems, the use of web-based and ecosystem-dominated applications, and openly available components. This makes requirements engineering and elicitation of requirements of an IS more demanding.

Sub-section 3.3 focused on the "future" of data modelling. The future is partially already here as a continuum of the current movement towards the more mature solution models and technologies of the future. These are long-term changes in data modelling. Consequently, we have included these in the *system changes* category. Almost all the papers we analysed underlined *AI and ML* as the key change factors in data modelling. Undoubtedly, these have characteristics of triggering / enabling technologies. We do not want to separate these from their application context: on their own they are not important but, when connected to a variety of application contexts, they are. AI supports the work of data analysts, which is partially *automated*. Part of the work is also *transferred* to the stakeholders in their other roles – as *citizen data scientists*, *low code programmers*, *end users* etc. Data modelling must support the heterogenous needs of all these groups – the transition to *multilevel concerns* covering a *variety of abstraction levels* is a reality. Adaptable schemas, adaptable data modelling and visual model building are needed to support the needs of end users. In addition, the agile development process needs an on-the-fly type of modelling. *IS complexity* is growing. Consequently, models are growing, and focus must be placed on a multitude of systems instead of only one. Data management solves complex business problems rather than less complex IS problems. In data modelling the role of *external data* (pipeline-based, having the source in IoT nodes, sensors and a variety of data streams) is growing. Challenges in managing mixed (hybrid) data formats in one application make data modelling complex. External pipeline data requires *reverse modelling* without the opportunity to affect its structure. *Platform* and *ecosystem* based IS development supports the transition from specific data models to less comprehensive, problem area specific and general-purpose data models that support interoperability between independent systems. *RPA* transfers the focus in modelling from IS level to business process modelling.

In section 4, we will build a view of the future by presenting ideas about paradigm changes in data modelling.

4. Future paradigms in data modelling

4.1. Data modelling will become database programming

Data models are widely used in most data-intensive applications. Data models could be reduced to a database systems construction. There are, however, also other scenarios for the usage of a data model: communication and negotiation, description and conceptualization, documentation, explanation and discovery for applications,

knowledge discovery and experience propagation, and explanation and discovery for systems.

For system construction, we can see the *first generation of data modelling*: Models are used as a mediator and blueprint for database realization, for the description of the data in an application, for negotiation etc. They allow reasoning on database systems, i.e. *model-based development* and reasoning. This generation can be characterized as *data-modelling-enabling-data-programming*.

Jaakkola and Thalheim in 2020 [18] have also envisioned a *second generation of data modelling*: Model development and the modelling infrastructure have reached a maturity that allows the derivation of database structuring and supporting programs (or their essential parts) from data models or data model suites. Models will thus be the source code for executable programs. We characterize this generation as *data-modelling-as-programming*.

The next generation can be characterized as *data-models-as-database-programs*. Data models can be directly translated into database dictionaries and database programs, i.e. database programming can be performed almost entirely through modelling. The third generation of data modelling is then the *true fifth generation database programming,* which essentially frees the database developer from writing database programs and host programs with third or fourth generation languages. Everybody, including database laymen, who can specify data models will become a database programmer. Data models will then be model suites.

4.2. Data modelling must change along with application changes and technology evolution

Modern database applications are becoming more voluminous, have higher velocity, use a variety of models, are often hampered by data quality and veracity, are becoming more viable, have to cope with variations, use highly heterogeneous data, are challenged by viability, and must provide tangible added value. An appropriate technology has still not been found. To some extent, big data machines can solve some of the problems. It seems, however, that conventional technology must also be revised. Already very large data sets frighten us and result in the avoidance of complex operations beyond $n \cdot \log(n)$ complexity. Big data should be supported by technology with operations far below data size (i.e. n) complexity. Proper data modelling approaches that could meet these challenges are under investigation. So far, modelling research has not led to any convincing approaches.

The classical database systems have been based on OLTP-web_frontend, distributed database, OLTP-OLAP, OLTP-data_warehouse-data_mart, or the micro_data-macro_data-analysis_data layered database architecture. They result in an increase in data size and computational problems. We already know solutions for modelling through *model suites*, i.e. a collection of coherent and sufficiently tightly associated models where each of the data models reflects some of the aspects in these architectures. Kiyoki & Thalheim in 2013 [19] revised the layered architecture for a real big data application to a highly *distributed data network* with some data collaboration styles and patterns, e.g. publish-subscribe or master-slave. Each node in this network could use its data model. In the given case, each node used an internal data model, an import data model and an export data model where the last two were essentially views of the first model.

The application world will result in a number of rapid changes to data modelling as well as the capability of modern systems. All these changes will initiate revision modernization of data modelling. Let us consider some of them:

- *Flexible model transformation without information loss*: Models represent a specific application viewpoint that coexists with a larger variety of viewpoints in other application fields. At the same time, partners essentially operate using the same data. Therefore, we need data models that can be transferred to each other without loss of data and structure information. The notion of infomorphism is a starting point.

- *Large data models*: Already legacy (or rather heritage) applications use models that have been developed by generations of developers and have become "mannered and stilted cathedrals". With the wide spread of database technology, data models become larger and larger. Their management is becoming an obstacle for applications.

- *Weakly structured data*: Data might be weakly structured and can be enhanced at a later stage with refined structuring. Data models must be sufficiently robust for this kind of evolution or complete modernization.

- *Data models reflecting provenance and quality*: Data is seldom 100% correct. This observation was already true in 1999. Nowadays, such data is widely used. We use data that is partially corrected and cleansed together with data that needs improvement. Data might also be changed especially for biased usage with decreasing quality. Engineering has learned how to cope with errors inside its systems without changing the system. Such kind of inner quality correcting data models? allow to become robust.

- *Toughening the data model*: A data model might be a starting point that is continuously improved during exploitation. The model has a higher quality within the quality portfolio of the application. The classic 'code-that-smells' repair approaches will be replaced by the co-evolution of models and their data.

- *Adaptable and self-adapting data models*: Data structures and applications continuously evolve. This evolution can be mastered by a real artisan. It is far better to integrate the evolution strategy and the adaptation tactics into the data model.

- *Heritage extraction models*: Many applications have journeyed through data models that have evolved over time. This journey is badly documented. It is also hampered by very special implementation issues. We will fall into the Excel macro trap if we are unable to recover the models in such a form that the current status of data is captured.

- *Just-in-time data models*: Data models will be developed wherever data occurs in an application. The initial data model will often be proprietary and not harmonized with existing ones. A typical case is large research clusters (e.g. within the German Excellence Programme) without controlled data integration.

- *Data models reflecting storage are separated from data models supporting computation*: It is not necessary to keep all the data together with the data processing facilities. Instead, we might use a universal data store with all the data together with local data stores that neatly correspond to the local application. The binding mechanism could be a sophisticated export/import view management.

- *Special data models for special tasks*: Data mining and analysis also target the detection of patterns within the data. These patterns are the kernel of explanation models. The data used is empirical data and is thus often of problematic quality. We might use data model pipelines that reflect this mining and analysis framework.

Similarly, many applications need their specific domain-oriented modelling language which is well embedded into the application domain. Enterprise data models must thus be different from social data models.

- *Layman data models*: Data models are often developed by users that do not have sufficient experience and knowledge of modelling. Models reflect their disciplinary, education and business context without explicitly stating it. Models are local. Their integration is only possible if the full context is known.
- *Harmonization of data modelling languages with computation features*: Big data challenges current technology and will force us to redevelop current database technology. We envision that only those operations that perform well in a given architecture and in a given application world are going to be realized in systems. Similar to the past when we co-developed algorithms with their own data structure, we need data modelling languages that support high-performance computation for challenging data volumes.

This list is at its best only a starting point. It should be enhanced by the trends discussed above in order to meet the challenges discussed above. One solution path to better data models is *standardization*. We start with high-quality models as components of the models. Model composition follows approved composition approaches. Mechanical engineering and industrialization have widely used this approach. Another solution is development through *generic models*. These models are supported by sophisticated refinement strategies and tactics for their improvement. Generic models are also widely used in engineering for production moulds or forms.

5. Conclusions

At the beginning of this paper we set our research problem: "What are the challenges for data modelling in the future?". We did not dissect it in the form of research questions but followed a systematic path that started with an analysis of the progress path of data modelling, covering the decades from the early phases of commercial computing until the present day. After that we made a "forecast of the past" by benchmarking a future forecast from 1999 – over twenty years ago. It provided us with an opportunity to compare the state of the art of today to the expected progress. Understanding the future is not possible without understanding the past – everything is a kind of continuum from the past to today and on into the future. The future analysis of the paper is based on the "literature" mapping study. We collected expert opinions from several sources about future visions in data modelling. These findings were then classified and analysed. No clear future path can be defined, but a set of dominant factors can be gathered – the growing complexity of models, increasing tool intelligence, growing importance of external (non-modellable) data, heterogeneity of stakeholders in modelling and technology as a change driver – just to mention some of the findings. The paper concludes with two future paradigms.

References

[1]   Boehm, B. (2006). A view of 20th and 21st Century Software Engineering. Paper presented at the Proceedings of the 28th International Conference on Software Engineering (ICSE), Shanghai, China.

[2]   Cagle, Kurt (2018). Why Data Modelling is Important (and Why It's Not). Retrieved from https://www.forbes.com/sites/cognitiveworld/2018/11/05/why-data-modeling-is-important-and-why-its-not/#1e9cc98e50cb on January 31st, 2020.

[3]   Chen, P. P., Thalheim, B., & Wong, L. Y. (1999). Future Directions of Conceptual Modeling. In G. Goos, J. Hartmanis, J. van Leeuwen, P. P. Chen, J. Akoka, H. Kangassalu, & B. Thalheim (Eds.), Conceptual Modeling: Current Issues and Future Directions (pp. 287-301). Berlin, Heidelberg: Springer Berlin Heidelberg. Available also from https://www.researchgate.net/publication/2847577_Future_Directions_of_Conceptual_Modeling on January 31st, 2020.

[4]   David, Amos (2017). Current trend in data modeling and information systems. International Conference on Applied Information and Communication Technology. Lead City University, Ibadan, Nigeria. Retrieved from https://www.researchgate.net/publication/327120249_Current_trend_in_data_modeling_and_information_systems on January 31st , 2020.

[5]   Date C.J. (2005). Database in depth: Relational theory for practitioners. O'Reilly, Sebastopol.

[6]   Embley D. and B. Thalheim, editors (2011). The Handbook of Conceptual Modeling: Its Usage and Its Challenges. Springer.

[7]   Foote, Keith D. (2017). A Brief History of Data Modeling. Retrieved from https://www.dataversity.net/brief-history-data-modeling/ on January 29th, 2020.

[8]   Frank U. (2014). Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design. Business & Information Systems Engineering, 6(6):319–337, 2014

[9]   Freeman C., Perez C. (1988). Structural Crises of Adjustment, Business Cycles and Investment Behavior. In Dodi G., Freeman C., Nelson R., Silverberg G. and L. Soete L. (Eds), Technical Change and Economic Theory. Pinter Publishers, London.

[10]   Frisendal, Thomas (2020), Ten 2020 Visions for Data Modelers. Retrieved from https://www.dataversity.net/ten-2020-visions-for-data-modelers/ on January 31st, 2020.

[11]   Guess A.R. (2020). Low-Code 20/20: A Clear Vision for Solving the Software Crisis. Retrieved from

[12]   Gartner (2019). To 10 Strategic Technology Trends for 2019. Retrieved from https://datavizblog.com/2018/11/18/gartner-top-10-strategic-technology-trends-for-2019/ on January 29th, 2020.

[13]   Ghosh, Paramita. (2019). Data Modeling Trends in 2019. Retrieved from https://www.dataversity.net/data-modeling-trends-in-2019/ on January 29th, 2020.

[14]   Harper, Jelani (2019). 2019 Trends in Data Modeling: Real-Time Integration for Cognitive Computing. Retrieved from https://aibusiness.com/2019-trends-data-modeling/ on January 31st, 2020.

[15]   Jaakkola, H., Henno, J., & Mäkelä, J. (2017). Technology and the Reincarnation Cycles of Software. In Z. Budimac (Ed.), SQAMIA 2017 - Proceedings of the Sixth Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications. Belgrade, Serbia, September 11-13, 2017. (Vol. Vol-1938, pp. 5:1-10). Belgrade, Serbia: CEUR Workshop Proceedings.

[16]   Jaakkola, H., Henno, J., Mäkelä, J., & Thalheim, B. (2017a). Today is the Future of Yesterday, What is the Future of Today? In P. Biljanović (Ed.), MIPRO 2017 - Proceedings of the 40th Jubilee International Convention. May 22-26, 2017, Opatija, Croatia. (pp. 741-749). Opatija, Croatia: Mipro and IEEE.

[17]   Jaakkola, H., Henno, J., Thalheim, B., & Mäkelä, J. (2017b). The educators' telescope to the future of technology. In P. Biljanović (Ed.), MIPRO 2017 - Proceedings of the 40th Jubilee International Convention. May 22-26, 2017, Opatija, Croatia. (pp. 766-771). Opatija, Croatia: Mipro and IEEE.

[18]   Jaakkola H. and Thalheim, B. (2020). Model-based fifth generation programming. In Information Modelling and Knowledge Bases Vol. XXXI, Frontiers in Artificial Intelligence and Applications, 312, pp. 377–396. IOS Press.

[19]   Kiyoki Y. and Thalheim, B. (2013). Analysis-driven data collection, integration and preparation for visualisation. In Information Modelling and Knowledge Bases, volume XXIV, pp. 142–160. IOS Press.

[20]   Kiyoki, Y., Chen, X., Sasaki, S. and Koppåipat, C., Multi-Dimensional Semantic Computing with Spatial-Temporal and Semantic Axes for Multi-spectrum Images in Environment Analysis. In Welzer, T., Jaakkola, H., Thalheim, B., Kiyoki, Y., Yoshida, N. (Eds.), Information Modelling and Knowledge Bases XXVII. IOS Press, 2016, pp. 14-31.

[21]   Knight, Michelle (2020). Data Modeling Trends in 2020: A Year of Optimization. Retrieved from https://www.dataversity.net/data-modeling-trends-in-2020-a-year-of-optimization/ on January 31, 2020.

[22]   Molnar A. and B. Thalheim (2007). Conceptual development of OLAP applications. In Business Intelligence: Methods and Applications, pp. 27 – 38. Klöden-Verlag, 2007.

[23]   Pettersson F., Ivarsson M., Gorschek T., Öhman P. (2008). A practitioner's guide to lightweight software process assessment and improvement planning. Journal of Systems and Software 81, 6 (June 2008), 972–995. DOI:https://doi.org/10.1016/j.jss.2007.08.032.

[24]   Roussopoulos N., Karagiannis D. (2009). Conceptual Modeling: Past, Present and the Continuum of the Future. In: Borgida A.T., Chaudhri V.K., Giorgini P., Yu E.S. (eds) Conceptual Modeling: Foundations and Applications. Lecture Notes in Computer Science, vol 5600. Springer, Berlin, Heidelberg

[25]   Thalheim B. (2000) Entity-relationship modeling – Foundations of database technology. Springer, Berlin.

[26] Thalheim, B. (2008). Model suites. In H. Jaakkola, editor, Selected Topics on Distributed Disaster Management: Towards Collaborative Knowledge Clusters., pp. 108 – 128. Tampere University Press, Pori unit.

[27] Thalheim B. (2018) Conceptual model notions - a matter of controversy; conceptual modelling and its lacunas. EMISA International Journal on Conceptual Modeling, February, pp. 9–27.

[28] Wells, Dave (2016). Big Changes in the World of Data Modeling. Retrieved from https://tdan.com/big-changes-in-the-world-of-data-modeling/19578 on January 31st, 2020.

[29] Yrjönkoski, K., Jaakkola, H., Mikkonen, T., Systa, K., & Henno, J. (2019). SQAMIA: Software business: a short history and trends for the future. In Z. Budimac & B. Koteska (Eds.), 8th Workshop on Software Quality Analysis, Monitoring, Improvement and Applications, SQAMIA 2019 Proceedings (Vol. Vol-2508, pp. 18:11-18:18): CEUR Workshop Proceedings.