Retrieval of Enterprise Models from PowerPoint: Solving Semantical Heterogeneities

Achim Reiz[0000-0003-1446-9670] and Kurt Sandkuhl[0000-0002-7431-8412]

Rostock University, 18051 Rostock, Germany
[achim.reiz, kurt.sandkuhl]@uni-rostock.de

Abstract. Grass-root enterprise modeling aims at enabling all stakeholders in an organization to create models or model-like content without the need to follow and learn strict modeling languages, tools or guidelines. While this would help to spread the use of Enterprise Modeling (EM), it would also require “light-weight” modeling tools or the use of widely available office tools for modeling. However, this.has its downsides regarding the technical quality of the models and adherence to meta-models. Due to the lack of formal notion, technical and semantical heterogeneities can occur.
In a previous paper at PoEM 2018, we presented a model retrieval algorithm from .pptx documents based on an ADO.xx Meta-Model and discussed possible heterogeneities for analyzing these unstructured models. This paper first briefly recapitulates the retrieval algorithm, and then proposes an algorithm for solving the semantic ambiguities “One diagram distributed over multiple slides” and “Multiple diagrams on one slide”. This includes a brief description of the mechanics of the algorithm as well as an example based on a prepared slide-set. In the end, we demonstrate practical limitations and give an outlook on possible solutions as well as further research.

Keywords: Enterprise Modelling, Grass-Root Modelling, Document Retrieval, PowerPoint, ADO.xx

1 Introduction

The discipline of Enterprise Modelling (EM), the formalization of a structure or behavior of an enterprise using a well-defined modeling language [1], is becoming more and more relevant for enterprises to achieve quality attributes like agility, adaptability and interoperability [2]. Historically, enterprise models were created by distinctive modeling departments in consultation with domain experts of the affected departments [3, p. 201]. But lately, research suggested that this does not unleash the full potential of EM: Due to the small teams, only a fragment of the information can be captured and made available throughout the enterprise. [4, p. 226].

Facing this challenge, the idea of Grass-root modeling arises. Grass-root modeling not only accepts but encourages the creation of models by everyone within the company. Informal drawings, like PowerPoints, often contain invaluable knowledge and even comply with the criteria for being models. [4, p. 226] But the use of PowerPoint comes with significant downsides as well. To overcome the downsides, we proposed the first step towards an automated document analysis in an earlier paper [5]. In this

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2499/paper1.pdf

preliminary work, the technical challenges of getting data out of PowerPoint were outlined and an algorithm for examining PowerPoints towards an ADOxx Meta Model was proposed. Also, we discussed several technical and semantical ambiguities of PowerPoint drawings.

This paper continues this research. After a brief overview of the different approaches to modeling in section 2, two examples of semantical heterogeneities are further discussed. It includes the implications of these modeling inaccuracies on the PowerPoint retrieval algorithm: The distribution of one diagram over multiple slides as well as the opposite: Having multiple diagrams on one slide. Section 4 shows the practical limitations of the current approach, followed by a conclusion and an outlook on further research.

2 Tool Support for Modelling

Modeling does not necessarily need to be complex and formalized. Drawing tools like PowerPoint enable a broad mass of people to create their own models – even in disciplines and granularities where models have not been present so far. But this opportunity comes with downsides as well. The following chapter gives an overview of the advantages and disadvantages of formalized and unformalized modeling approaches.

2.1 Modeling in PowerPoint

PowerPoint, the SlideShare Program invented in 1984 is a milestone in communication preceded in importance only by paper, the blackboard, the whiteboard, the overhead, and the slide projector. [6, p. 121] The SlideShare program offers a lot of advantages: It is widespread through organizations of every kind and is generally the accepted discussion format: Drawings can be shared without additional intermediate steps, through most of digital channels like chats, mails, wikis, etc. [7, p. 1778, 8]. Especially in the innovation stage, people can visualize ideas with the full flexibility and freedom of expression. As almost everybody uses PowerPoint, this tool supports Bottom-Up innovation practices and creates a better understanding of digital innovation itself [9, p. 220].

It is therefore not a surprise that not a formalized modeling tool, but PowerPoint is the most used tool for drawing models for almost every model type like use case, activity, architecture diagrams to business process models, etc. Even though a lot of employees have profound knowledge of formal modeling languages, they often refuse to use these kinds of tools in practice and prefer the usage of PowerPoint. [8, 9, p. 220]. Although formalisms are not well known, people often even tend to develop EM without an explicit intent to model. They still draw diagrams that fit the criteria of a model: An abstraction, a reduced view for a purpose, pragmatic towards a defined stakeholder. [4, p. 226]

In defiance to the mentioned upsides of modeling with PowerPoint, it tends to have significant downsides as well: Even though internal conventions regarding the design and modeling of slides and diagrams might exist, especially new employees might not be familiar with these modeling rules [8]. As a result, the goal of a model – to create a common understanding – might not be reached, enterprise architectures are covered with informal drawings of clouds and arrows that need the specific context to be understood [10, p. 206]. It is also difficult to spread the knowledge generated in PowerPoint across the Enterprise. On the one hand, it comes from the design of PowerPoint itself: while it is originally designed as a presentation tool, it is more and more used for documentation purposes. But these two goals partly contradict each other. [7, p. 1778] While the focus on a presentation is a onetime deliverable, a good documentation tool has to manage the ownership as well as new versions and updates on information. Both functions are not part of PowerPoint: Once the slides are designed, they might not be kept up to date. The models decay. On the other hand, the goal of the employees modeling in PowerPoint or in Enterprise Modelling Tools often differs: While an important attribute of Enterprise Modelling is the holistic view on the whole organization, PowerPoint models are mostly created and used by single departments (sometimes even without an explicit modeling intention), the models – even though they are correct and provide value for the enterprise – are just showing an isolated view without considering all dependencies within the company. A lot of independent local models might exist, the scope or accuracy might differ a lot. [4, p. 227] But as long as they are just created, used and stored locally, they cannot unleash their full capabilities as an Enterprise Model.

2.2 Enterprise Modelling Tools

As described above, PowerPoint diagrams have significant downsides regarding information governance, knowledge management and the creation of holistic, comprehensive models. Structured enterprise modeling tools can offer a solution to the described issues. While there is no designing governance in PowerPoint implemented, Enterprise Modelling Tools provide exactly that: A tool that contains guidelines and a formal foundation which ensures that the diagrams do not suffer from ambiguities and are fitted for an automated analysis [10, p. 206, 11, pp. 145-146]. Due to the fact that PowerPoint does not enforce modeling languages or meta-models, it can model every kind of notation. In opposite, enterprise modeling tools are fitted for a particular application domain and support the necessary concepts and functionality to model the reality to a chosen standard [12]. They also often integrate a knowledge management system in the form of a shared repository for all models.

There are a lot of different tools available. Most of them focus on specialized application domains: ARIS, for example, is developed for the creation of business process models [13]. Archi is a modeling tool for ArchiMate, a standard for EA modeling by The Open Group [14]. The Eclipse Modelling Framework (EMF) is focused - but not limited – on the creation of programming code based on modeled abstractions [15].

The wide range of different tools for specialized purposes leads also to a difficulty Vernadat described as the “Tower of Babel-Problem”. As every tool and every related language needs dedicated training, it might be impossible to know every formal notation used in the enterprise. To understand a new modeling tool, it is often necessary to learn new vocabulary, interface, and paradigms. Also, the various tools do not share a common data basis – the exchange and interconnection between the different software are not possible. This leads to the problem of missing trained staff, especially in larger and more complex projects where more people have to be involved [1, 16].

In comparison to most modeling tools that specialize on one modeling language, ADOxx has meta-modeling capabilities: instead of being specialized for one modeling language or approach, it provides the underlying constructs for meta-modeling, i.e. developing a tool for any modeling language. With the ADOxx development Toolkit, an Administrator can create a meta-model. This meta-model contains all elements like concepts and the corresponding relations that can be included later in the diagrams. It is also possible to add additional model functionality or validation by programming routines in ADOscript, the proprietary internal script language. Because the ADOxx library is uniform for every kind of notation, XML-exports of these libraries are used as an input for the developed algorithm to analyze the PowerPoint slides.

3 Heterogenities in PowerPoint Models

Even though PowerPoint files (.pptx) are based on an XML, the underlying data structure is not easily accessible for further analysis. Due to the drawing nature of the software, two diagrams with the exact same look can have different XML-representations. Reasons for that can be found in grouped shapes, dangling connectors or the use of Microsoft SmartArt. The possible challenges though are not limited to technical difficulties. On a semantical level, ambiguities can occur as well. Examples are underspecification, the use of the same shapes for different concepts, fused semantics, the insertion of multiple concepts into one shape or spreading one diagram over multiple slides.

In the previously released work [5], the possible heterogeneities and their implications on the retrieval algorithm were already discussed in detail. After a brief refreshment on the retrieval algorithm in section 3.1, this section examines possible solutions for two semantical heterogeneities: The spreading of one diagram over multiple slides as well as the opposite: Having more than one diagram on one slide.

3.1 Overview of the document retrieval algorithm

To identify diagrams, the retrieval algorithm opens the presentation and crawls through every slide for possible diagram candidates. A diagram candidate has at least two shapes that are connected with each other. If such is found, all shapes, as well as relations that are found in the slide, are stored as one data object in the internal data representation for further analysis. In perspective of the chosen scenarios, this behavior can cause some problems:

One Diagram in multiple Slide. If one diagram is distributed over two slides ore more, the algorithm threats every slide as a single, isolated entity. the connection between those diagrams cannot be retrieved. Instead of one single, coherent data fragment, the algorithm stores multiple – one for every slide. As a result, large distributed diagrams cannot be understood as the semantical information is lost.

Multiple Diagrams in one Slide. In the case that two different diagrams are drawn in one slide, they both are stored in one diagram data frame. From an algorithm perspective, the distinction between diagram parts that belong to each other but have no connection and two different diagrams with different meanings is not possible. Even though no shape or relational information is lost, the storage of these two diagrams into one data frame is semantically not correct.

3.2 Solving Heterogeneity "One Diagram on Multiple Slides"

A diagram that is stored in multiple slides will not be identified as one comprehensive diagram but as multiple with a different semantic meaning. This paper proposes an algorithm for resolving this heterogeneity. Therefore, it is assumed that two shapes describe the same concept if the shape has the same form and the same textual content.

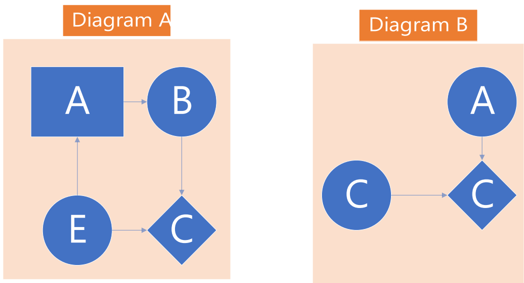

Fig. 1. Initial State. Due to the distribution of the diagrams over two slides, the algorithm stores them into two independent data fragments

After the initial run of the retrieval algorithm, the diagrams shown in Fig. 1 are stored in two data slots. After all data is extracted out of the PowerPoint, the software crawls through every slide and searches whether a shape with the same text and form exists in another diagram fragment as well. If that is the case, the shapes of both slides are combined into the first diagram fragment. In Fig. 1, Diagram A and Diagram B contain the rhombus-shaped "C". This is the trigger for the algorithm to merge the two objects into one. The first step for the algorithm is the storage of Diagram B into Diagram A.

After the two diagrams now share a common data fragment and therefore also a common meaning, the next step is the removing of redundant items: Due to the combination of slides, the first slide now contains two identical shapes. In the given example, the rhombus with the textual content "C" conditioned the merging event. In the next step, the two rhombuses are getting combined: All relations between the merged "C"-rhombus are getting connected to the old "C" rhombus and the new one will be deleted. The final diagram object shown below now contains all cohesive shapes stored with the right relations in one diagram object.

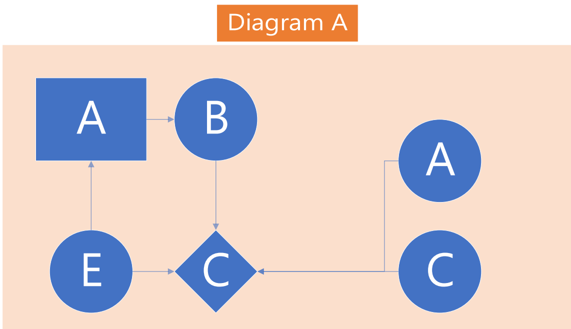

Fig. 2. The newly combined diagram with the corresponding relations.

If the slide deck is bigger, one iteration might not be enough. Imagine there are e.g. 4 slides (This example is independent of the examples shown in Fig. 1 - Fig. 2), with slide 1 containing (A, B, D), slide 2 containing (F, G, U), slide 3 containing (B, D, Z), slide 4 containing (F, G, D). In the first iteration of the algorithm, starting at slide 1, shape A, then B, then D get compared with the rest of the slides, then slide 2 F with slides 3 and 4, etc. After one iteration, the algorithm produced the following result: slide 1 (A, B, D, B, D, Z, F, G, D), slide 2 (F, G, U). The algorithm works recursively: As long as the count of diagrams gets smaller, the method will call itself. After the second iteration, slide 1 will contain (A, B, D, B, D, Z, F, G, D, F, G, U). As the last step, the program clears doubled shapes and the result will be: (A, B, D, Z, F, G, U).

Fig. 3 shows an example of a real input and output scenario of the software. Both slides contain an ER-diagram with the item set “Movie”. They most likely contain a common concept and can therefore be interconnected with each other. The algorithm identified the similarities, took the diagram name of the first slide and combined both diagram objects into one.

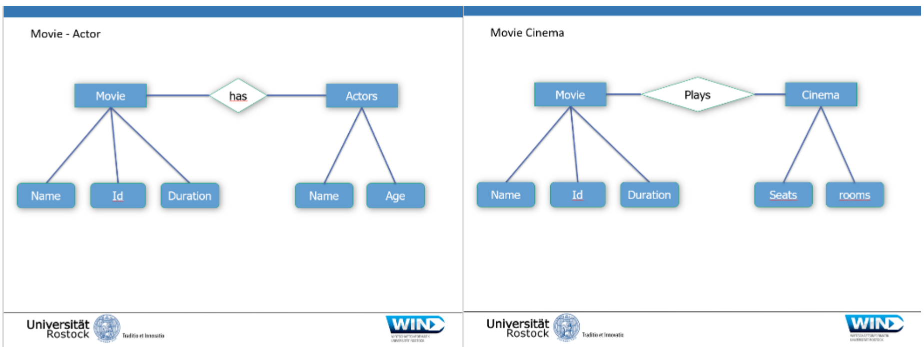

Fig. 3. Testing the algorithm with PowerPoint data

Table 1 prints out the new diagram object with name, shape types and the textual content of the shapes. There are no more doubled items in the shape object. Due to the limited capacity of the table, relations are not printed but are also newly connected to the "Movie" shape. As we can see in the column "Diagramname", the first slide is the one where the merging had taken place.

Table 1. Result: Solving "One Diagram on Multiple Slides"

Diagramname	ShapeType	ShapeText
Movie - Actor	RECT	Movie
Movie - Actor	ROUND_RECT	Name
Movie - Actor	ROUND_RECT	Id
Movie - Actor	ROUND_RECT	Duration
Movie - Actor	FLOW_CHART_DECISION	Plays
Movie - Actor	RECT	Cinema
Movie - Actor	ROUND_RECT	Seats
Movie - Actor	ROUND_RECT	Rooms
Movie - Actor	FLOW_CHART_DECISION	Has
Movie - Actor	RECT	Actors
Movie - Actor	ROUND_RECT	Age

3.3 Solving Heterogeneity "Multiple Diagrams on one Slide"

Besides the heterogeneity "One Diagram on Multiple Slides", it might also be possible to store multiple models in one PowerPoint slide. As the retrieval algorithm captures every slide as a single data object, these diagrams are stored in one data fragment. Besides the fact that this kind of storage is semantically incorrect, this can lead to further problems with the evaluation of the diagram, especially if both of the fragments contain different model notations (e.g. BPMN and UML).

For preventing these kinds of ambiguities, another cleaning algorithm additionally to the one described above was implemented. The software crawls through every slide and identifies groups of shapes that are not connected with each other (see example in Fig. 4). If such slide is found, the software splits the data fragment and creates an individual diagram object for every model.

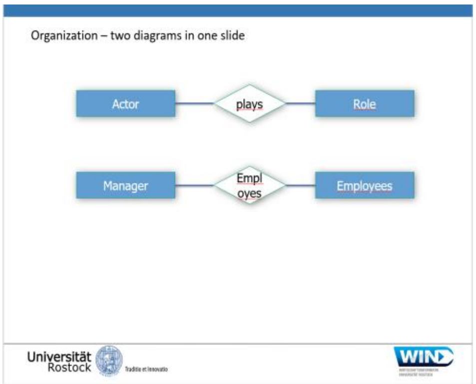

Fig. 4. Example for the Heterogeneity: Multiple Diagrams in one Slide

The algorithm detected that "Manager-Employes-Employees" and "Actor-Plays-Role" are not connected with each other nor share similar shapes. It therefore assumes independent concepts that have to be split up. As a result, the software creates an additional data fragment for the second model. It contains the title of the slide and combines it with an indexing number to create a unique model name. In the table below, the column "Diagramname" shows the split data objects.

Table 2. Result: Solving "Multiple Diagrams in one slide"

Diagramname	**ShapeType**	**Shape-Text**
Organization – two diagrams in one Slide	RECT	Actor
Organization – two diagrams in one Slide	FLOW_CHART_DECISION	Plays
Organization – two diagrams in one Slide	RECT	Role
Organization – two diagrams in one Slide – 2	RECT	Manager
Organization – two diagrams in one Slide – 2	FLOW_CHART_DECISION	Employes
Organization – two diagrams in one Slide – 2	RECT	Employ-ees

4 Practical Limitations

Multiple Diagrams on one Slide. The principle of the algorithm itself is very simple and robust. Problems arise if technical heterogeneities like dangling connectors or unlinked labels occur. As the algorithm searches for clusters of connected shapes that are not in relation to each other, an unlinked label like the example in Fig. 5 gets in the focus of the algorithm as well. As there is no connection between "Movie" and "has", the algorithm assumes two independent diagrams.

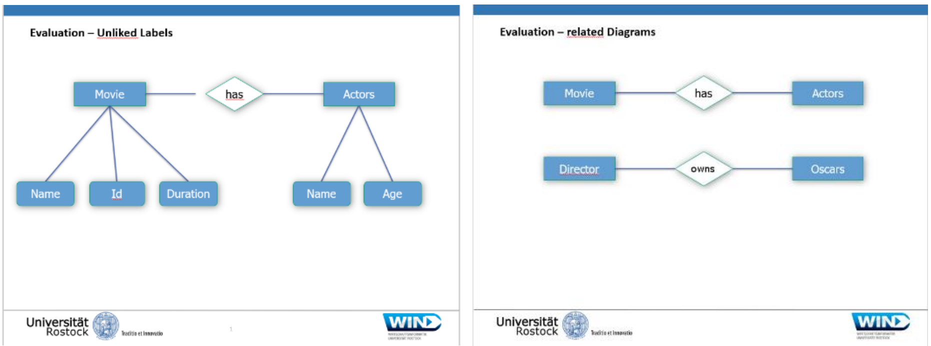

Fig. 5. One Diagram in Multiple Slides – defective Analysis

The often contained implicit information is a great challenge for the algorithm. While in the first example the connector is missing due to inaccurate modeling, in the second example there is no connection at all, but the domain of modeling is that close to each other that a connection between them can be assumed. It is arguable that in this kind of diagram, there is no need to split the slide into two diagram objects.

Table 3. Multiple Diagrams on one slide – Output

Diagramname	**ShapeType**	**ShapeText**
Evaluation – Unliked Labels	FLOW_CHART_DECISION	has
Evaluation – Unliked Labels	RECT	Actors
Evaluation – Unliked Labels	ROUND_RECT	Name
Evaluation – Unliked Labels	ROUND_RECT	Age
Evaluation – related Diagrams	FLOW_CHART_DECISION	has
Evaluation – related Diagrams	RECT	Actors
Evaluation – related Diagrams	RECT	Movie
Evaluation – Unliked Labels - 2	ROUND_RECT	Id
Evaluation – Unliked Labels - 2	ROUND_RECT	Name
Evaluation – Unliked Labels - 2	ROUND_RECT	Duration
Evaluation – Unliked Labels - 2	RECT	Movie
Evaluation – related Diagrams - 2	FLOW_CHART_DECISION	owns
Evaluation – related Diagrams - 2	RECT	Director
Evaluation – related Diagrams - 2	RECT	Oscars

A possible solution for dangling connectors is the consequent resolving of such technical ambiguities prior to the restructuring of the data. If all connectors are properly connected, the algorithms distinction between diagrams is more precise. The semantical affiliation of diagrams though is automatically solvable just to a certain degree. If there are no logical connections between diagrams, not even common names, the algorithm cannot decide towards a shared data fragment but rather store them separately.

One Diagram in Multiple Slides. Like in the heterogeneity "Multiple Diagrams in one Slide", the principle behind solving "Multiple Slides in one Diagram" is not complex as well. As long as unique concepts are described with unique names, the algorithm reliably detects the relation between models. A challenge are generic names for relational elements. In the example below, there are two independent diagrams displayed, but both with the connector "has". As the PowerPoint does not distinguish between a relation type and a shape, in this stage, the algorithm does not detect a difference between "Movie" and "has" and will find that the "has" element is similar in both slides.

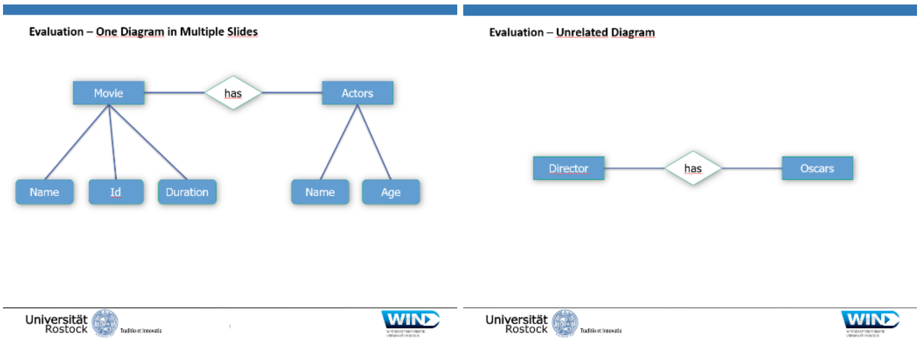

Figure 6: One Diagram in Multiple Slides – defective Analysis

This leads to a shared diagram object shown in Table 4Table 1. While both models ought to be independent of each other, the common object "has" bounds them together. The relation (which are not shown here due to the limited space of a table) are also reconnected from both "has"-object towards just one existing "has" item. In this example, the combination of both diagrams can interfere with the intended meaning of the modeler.

Table 4. Heterogenities - One Diagram in Multiple Slides

Evaluation – One Diagram in Multiple Slides	**ShapeType**	**has**
Evaluation – One Diagram in Multiple Slides	RECT	Oscars
Evaluation – One Diagram in Multiple Slides	RECT	Director
Evaluation – One Diagram in Multiple Slides	RECT	Actors
Evaluation – One Diagram in Multiple Slides	RECT	Movie
Evaluation – One Diagram in Multiple Slides	ROUND_RECT	Name
Evaluation – One Diagram in Multiple Slides	ROUND_RECT	Age
Evaluation – One Diagram in Multiple Slides	ROUND_RECT	Id
Evaluation – One Diagram in Multiple Slides	ROUND_RECT	Duration

There are a few thinkable solutions to the problem of the distinction between general terms and specific object descriptions. One solution could be a shared dictionary containing words that are marked as generic. Unfortunately, the idea of a thesaurus contradicts the idea of programming the software as general as possible. If the input source is just the Meta-Model, it is questionable that a dictionary could be created that can contain all relevant terms for all kinds of diagrams.

Another approach is to use more information from the Meta Model. ADOxx stores not only the possible directions of relations but associate them with a name as well. If a shape is a candidate for matching, it could be checked whether the shape content is similar to a stored relation. This, unfortunately, is not a valid solution for all kinds of Meta-Models. The example of the ER-Diagram is a good example: The relation “has” in the form of a Rhombus is an entity-object itself, not a relation, even though it represents one. Especially in larger PowerPoints, it might be possible to check for shapes that occur suspicious often. With an applicable threshold value, a shape can be identified as a general term and therefore be excluded for further merging analysis.

5 Conclusion and Future Work

Even though drawing tools like PowerPoint enable a bottom-up modeling culture, it comes with significant downsides regarding the reusability for the whole organization, as well as the possibilities for automatic analysis due to the occurrence of technical and semantical heterogeneities. Based on a previous publication on the PoEM 2018 where the general retrieval method was presented, we proposed two algorithms to resolve the issues: “multiple diagrams on one slide” and “one diagram on multiple slides”.

The results look promising. With the assumption that two or more groups of shapes that are not connected with each other can be split and two or more shapes that have the same form and content can be connected, the algorithm is able to reorder diagram-containing slides into separate, coherent models. Nevertheless, the algorithm cannot access implied knowledge that is hidden in the diagrams. Without this implicit knowledge, and with the dependency just on explicitly stated facts, it is not possible to distinguish between same looking concepts that have the same semantical meaning and those who do not.

Further research therefore has to test these assumptions with a broader set of data in a less controlled environment. In the future, the addition of titles or even machine learning technologies to as decision support is a possible research field as well.

References

[1] F. Vernadat, “UEML: Towards a unified enterprise modelling language,” *International Journal of Production Research*, vol. 40, no. 17, pp. 4309–4321, 2002.

[2] J. Zdravkovic, J. Stirna, M. Kirikova, D. Karagiannis, and R. Winter, “Advanced Enterprise Modeling,” *Bus Inf Syst Eng*, vol. 57, no. 1, pp. 1–2, 2015.

[3] H. Florez, M. Sanchez, and J. Villalobos, “iArchiMate: A Tool for Managing Imperfection in Enterprise Models,” in *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, Ulm, Germany, 2014, pp. 201–210.

[4] K. Sandkuhl *et al.*, “Enterprise Modelling for the Masses – From Elitist Discipline to Common Practice,” in *Lecture Notes in Business Information Processing*, vol. 267, *The practice of enterprise modeling: 9th IFIP WG 8.1. Working Conference, PoEM 2016, Skövde, Sweden, November 8-10, 2016 : proceedings*, J. Horkoff, M. A. Jeusfeld, and A. Persson, Eds., Cham: Springer, 2016.

[5] A. Reiz, K. Sandkuhl, A. Smirnov, and N. Shilov, “Grass-Root Enterprise Modeling: Issues and Potentials of Retrieving Models from Powerpoint,” in *Lecture Notes in Business Information Processing, The Practice of Enterprise Modeling*, R. A. Buchmann, D. Karagiannis, and M. Kirikova, Eds., Cham: Springer International Publishing, 2018, pp. 55–70.

[6] A. G. Gross and J. E. Harmon, “The Structure of PowerPoint Presentations: The Art of Grasping Things Whole,” *IEEE Trans. Profess. Commun.*, vol. 52, no. 2, pp. 121–137, 2009.

[7] D. Schoeneborn, “The Pervasive Power of PowerPoint: How a Genre of Professional Communication Permeates Organizational Communication,” *Organization Studies*, vol. 34, no. 12, pp. 1777–1801, 2013.

[8] R. Ciriello, A. Richter, and G. Schwabe, *PowerPoint Use and Misuse in Digital Innovation*. AIS Electronic Library: University of Münster, Münster, Germany, 2015.

[9] A. Karlsen, “Enterprise Modeling Practice in ICT-Enabled Process Change,” in *Lecture Notes in Business Information Processing*, vol. 92, *The Practice of Enterprise Modeling: 4th IFIP WG 8.1 Working Conference, PoEM 2011, Oslo, Norway, November 2-3, 2011, Proceedings*, P. Johannesson, J. Krogstie, and A. Opdahl, Eds., Heidelberg: Springer, 2011, pp. 208–222.

[10] M. M. Lankhorst, “Enterprise architecture modelling—the issue of integration,” *Advanced Engineering Informatics*, vol. 18, no. 4, pp. 205–216, 2004.

[11] Debdoot Mukherjee, Pankaj Dhoolia, Saurabh Sinha, Aubrey J. Rembert, and and Mangala Gowri Nanda, “From Informal Process Diagrams to Formal Process Models: 8th international conference, BPM 2010, Hoboken, NJ, USA, September 13 - 16, 2010 ; proceedings,” (eng), vol. 6336, pp. 145–161, http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10417272, 2010.

[12] N. Efendioglu and R. Woitsch, Eds., *Modelling Method Design: An Adoxx Realisation*.

[13] Software AG, *Welcome to ARIS Community!* [Online] Available: http://ariscommunity.com/. Accessed on: Apr. 13 2018.

[14] P. Beauvoir, *Archi: The Free ArchiMate Modelling Tool.* [Online] Available: https://archimatetool.com/. Accessed on: Apr. 13 2018.

[15] Eclipse Foundation, *Eclipse Modeling Framework (EMF).* [Online] Available: https://www.eclipse.org/modeling/emf/. Accessed on: Aug. 13 2018.

[16] S. Forster, “Investigating the Collaborative Process of Process Modeling,” *Doctoral Consortium of the 25th International Conference on Advanced Information Systems Engineerin*, no. 1001, 2013.