Formal Modelling Approach of Enterprise Architecture

Hypergraph based Representation of Business Information Systems

Dóra Őri[1], Bálint Molnár[2] and Zoltán Szabó[1]

[1]*Department of Information Systems, Corvinus University of Budapest. Budapest, Hungary*

[2]*Department of Information Systems, ELTE, Eötvös Loránd University, Budapest, Hungary*

Keywords: Enterprise Architecture, Business Information System, Strategic Alignment, Formal Modelling, Hypergraph.

Abstract: The complexity of strategic alignment is an overwhelming issue in the digital age for organizations. Enterprise Architecture Management (EAM) is a major tool that facilitate the alignment efforts, providing several methods for planning and analysis. There are several methodologies that need a formal and systematic approach. The artefacts describing an Enterprise Architecture can be perceived as documents that can be represented in hypergraphs. The graph-based approach lays the groundwork for formal analysis that can assist to identify discrepancies, gaps, security, integrity and consistency issues. The paper depicts a high-level model for artefacts representing Enterprise Architecture in a hypergraph formalism. This approach can be a promising solution for EAM-based analysis of information systems and their organizational context.

1 INTRODUCTION

Information strategy planning is a complex and especially important activity of organizations, an exercise through which an organization utilizes its technological resources (Earl, 1989). Information strategy is a major tool to integrate information technology (opportunities and concerns) into business planning, harmonizing the business and IT domains. The overall goal of IT planning, as part of the broader concept of IT governance is the alignment of information systems (IS) and business plans. In the era of digitalization, the growing organizational complexity, emerging disruptive technologies, and rapid technological changes make strategic planning of IS extremely challenging. Classic planning approaches (Peppard and Ward, 2016) are still used, but recently the enterprise architecture management (EAM) based approaches are the major facilitators of the planning initiatives (Hanschke, 2009).

Several well-known methods can be used for the harmonization of IT initiatives with business goals, but complexity and uncertainty of the technical and business domains has become an overwhelming issue with the traditional approaches of strategic IT planning. The planning cycle, integrated with implementation and monitoring activities requires the support of EAM. Competition is technology-dependent, new IT-based innovative models increase the pressure for organizational changes. In addition, legal constraints, compliance requirements and ethical issues make planning more and more difficult. Planning has no sense without implementation – but the burden of legacy systems, organizational inertia (products, processes, sales channels, partners, regulation, etc.), interdependencies of IT services make the development projects (and maintenance too) cumbersome. Strategic harmonization of business and IT domains is more relevant than ever before, and EAM can be a major facilitator of strategic alignment by discovering, analysing and avoiding misalignment problems and achieving competitiveness (Versteeg and Bouwman, 2006). EAM provides tools and methods to reduce organizational complexity (Strnadl, 2006), promotes agility, and controls uncertainties (Choi et al., 2013). EAM helps the alignment of the organisation with strategic goals, the control of interdependencies in business and IT, and it enables organisations to agility and fast reaction. Strategic EAM facilitates strategy formulation too, by analysing the current (business and IT) situation, by assessing strategic options, by formulating strategic initiatives, by developing an architectural vision, by planning roadmap migration activities, by evaluating project portfolio, by monitoring architecture evolution (Lankhorst, 2013; Ahlemann et al., 2012).

Őri, D., Molnár, B. and Szabó, Z.
Formal Modelling Approach of Enterprise Architecture.
DOI: 10.5220/0006804607270735
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018)*, pages 727-735
ISBN: 978-989-758-298-1

The complexity of infrastructures of IT and Information Systems (IS) coerces the exercise of modeling of Enterprise Architecture within companies. There are several approaches of Enterprise Modeling (Zachman 1987; Bent et al., 2008) that provide opportunities for semi-formal modeling through exploiting visual representations and specifications of various pre- and post-conditions. The central concept of Enterprise Architecture Modeling is the artefact that is the outcome of some modeling, designing and analysis activity. The artefacts can be considered as documents that describe architectures through complex relationships among the elements of artefacts. As the documents allow for depicting multifaceted relations among components that reflect the intricate relationships among the building blocks of architecture, the representation by a formal approach requires a flexible descripting method in which there are no restrictions on enhancing and extending the representation with new type of relationships, concepts, hierarchies, and networks. The hypergraph theory provides a very elastic mathematical structure that has the capability, on the one hand, to mirror the multifarious dependencies among constituents, and on the other hand, to exploit the graph structure for analysis utilizing the tool set of mathematics. This paper is intended to discuss the topic of EAM-based analysis of misalignment problems, introducing an existing method (Őri, 2017). The EAM-oriented analysis model is extended with a hypergraph-based approach, and the foundations of the extended conceptual framework is described. Our research objective was to provide a solid foundation for the extended, more comprehensive EAM-based analysing framework, and prepare the future implementation.

The rest of the paper is structured as follows: Section 2 summarizes the theoretical background to the subject. The hypergraph-based approach is introduced in Section 3. Section 4 presents an illustrative example for the proposed formalism in form of a case study. At the end of the paper conclusions are drawn.

2 THEORETICAL CONTEXT

The theoretical foundation of the paper consists of 2 parts. Firstly, alignment and misalignment assessment approaches will be summarized. Secondly, enterprise architecture analysis methods will be presented.

There are four dominant alignment perspectives, so-called cross-domain relationships in the Strategic Alignment Model (SAM): 1) Strategy Execution, 2) Technology Transformation, 3) Competitive Potential and 4) Service Level (Henderson and Venkatraman, 1993). Process models of alignment accent the process-like nature of alignment (vs. end state). Several process models of alignment deal with the evolution of alignment (i.e. how alignment has changed over time). In the literature there are several examples of alignment assessment methods. Many attempts have been made in order to classify and analyze alignment evaluation techniques. In general, alignment can be measured by different approaches, including e.g. typologies and taxonomies, fit models, mathematical calculations, survey items, qualitative assessments and psychological measures (Chan and Reich, 2007). In recent years a growing body of literature has examined alignment evaluation methods. Most of the introduced approaches for alignment measurement build on strategic and/or functional level assessment and include top-down construction approach (Chan and Reich, 2007).

There are a few misalignment models mentioned in the literature. The most famous ones are the BISMAM model (Business and Information Systems MisAlignment Model) by Carvalho and Sousa (2008) and the BITAM method (Business IT Alignment Method) by Chen et al. (2005). The former provides different classification schemes for the indicators of misalignment. One of them, misalignment symptoms are considered as evidences of inefficiencies, difficulties or inabilities that encumber alignment achievement. Misalignment symptom detection deals with the identification of such indicators. Several misalignment symptom collections have been proposed in recent literature on misalignment. These collections contain different types of misalignment symptoms, e.g. Carvalho and Sousa (2008). Symptoms can be found e.g. in EA models. However, other kinds of sources can also be used in the analysis (see Purao and Desouza, 2010).

Enterprise architecture (EA) is the construction of an enterprise, described by its entities and their relationships. EA is an organising logic for business processes and IT infrastructure in order to review, maintain and control the whole operation of an enterprise (Zachman, 1987, Kossak 2016). Enterprise architecture management is a management philosophy concerned with corporate change. EAM provides several benefits by improving IT efficiency (reducing redundancy, ensuring homogeneity, integration, consistency, reusability); by enabling IT effectiveness (ensuring goals, strategy and means conformity, results orientation, schedule orientation); by improving IT reliability (reducing risk) (Niemann, 2006). An enterprise architecture framework is a collection of descriptions and methods to create and manage enterprise architecture. The most recognised frameworks are the Zachman Framework (for rather theoretical purposes) (Zachman, 1987) and the TOGAF framework (for rather practical usage) (TOG, 2015).

TOGAF (The Open Group Architecture Framework) is a commonly used architecture framework. It is a holistic approach which describes a metamodel for enterprise architecture. TOGAF provides 4 architecture layers: 1) Business Architecture, 2) Data Architecture, 3) Application Architecture and 4) Technology Architecture. TOGAF metamodel is a reference model which sets up the formal structure of an EA model as well as provides implementation guidance on core building blocks (metamodel entity) and their relationships. TOGAF describes different viewpoints for enterprise architecture. TOGAF provides a minimum set of necessary EA models, called artefacts (TOG, 2015).

Architecture principles and patterns are used for framing the architecture content (Pessi et al., 2011). Enterprise architecture analysis types are methods that are capable of assessing EA models, e.g. evaluating dependencies, isolated objects, complexity or heterogeneity. A number of research efforts have focused on proposing models for EA analysis, i.e. EA-based analysis types that are capable of assessing EA models. Ullberg et al. (2010) introduced general process models for EA analysis. Several authors [e.g. Wagter et al. (2012)] proposed EA analysis collections. According to the object being investigated (e.g. Dependency, Coverage, Interfaces, Heterogeneity, Complexity and Conformity) different analysis procedures are introduced. Tools for supporting the process of EA analysis are expounded by e.g. Ramos et al. (2015). Automated collection of EA models is supported by e.g. Holm et al. (2014). Őri (2017) provided a detailed comparison of possible formal approaches for implementation.

3 A HYPERGRAPH-BASED APPROACH

The hypergraph structure provides a generic model for representing very complex relationships between "things" as component of Information Systems, Description of Enterprise Architecture. The adequately formulated conditions and properties make possible to leverage the set of graph algorithms to discover gaps, discrepancies, misalignment between the realized Enterprise and Information Architecture and the ideal typic one required by the Enterprise IT strategy (Molnár, 2017).

The representations of artefacts for architecture materialize as documents in XML and / or JSON document format typically. The hypergraphs, especially the generalized hypergraphs provide a flexible structure to describe complex relationships that can be explored among models during analysis and design of IS (Bretto, 2013).

Definition 1. The concept of the directed hypergraphs is an ordered pair of vertices and hyperarcs that are directed hyperedges, i.e. each hyperarc is an ordered pair that contains a tail and a head.

Definition 2. The generalized or extended hypergraph. The notion of hypergraph may be extended so that the hyperedges can be represented – in certain cases – as vertices, i.e. a hyperedge e may consist of both vertices and hyperedges as well. The hyperedges that are contained within the hyperedge e should be different from e.

The hypergraphs as a tool for describing Information Systems from various viewpoints yields a formal method to analyze the system, and to check the conformance, compliance, and consistency of the set of models (Molnár, 2017). An hypergraph is structured as:

- V is the set of vertices
- A is the set of arcs, i.e. directed edges, an arc is an ordered pair j,i , where $V_{j,i}$
- E is the set of hyperedges
- E_D is a set of hyperarcs

A hypergraph is a generalization of an ordinary graph where edges, called hyperedges, can connect any number of nodes. Formally, let $G\ (V, E, w)$ denote a hypergraph, where V denotes a finite set of nodes v, E denotes the set of hyperedges e, w is a weight function defined as: $w : E \Rightarrow R$. Each hyperedge $e \in\ E$ is a subset of V and is assigned a positive weight $w(e)$.

Definition 3. A directed hyperedge or hyperarc is an ordered pair, $E = (X, Y)$, of (possibly empty) disjoint subsets of vertices; X is the tail of E while Y is its head. In the following, the tail and the head of hyperarc E will be denoted by $T(E)$ and $H(E)$, respectively.

Definition 4. *Architecture Describing Hypergraph* is a generalized hypergraph that can be extended by some functions and operations:

$label_{node}: V \to L_{node}$; where L is a set of labels, it is a vertex labeling function;

$label_{edge}: E \to L_{edge}$; where L is a set of labels, it is an edge labeling function;

$source_E : E \to V;$

$target_E : E \to V;$ these functions return the source and target vertices of an edge *E;*

$attr : Attr \to V ;$ attribute assignment function;

$source_{Attr} : Attr \to V;$ The vertex that owns the attribute is returned;

$target_{Attr} : Attr \to D;$ The *data values* of attributes are yielded; *D* represents the set of data;

D can be grasped (efficiency of the representation is left out of the investigation) again as vertices within the hypergraph and it can be interpreted as *variables*.

Over *D* as a set of variables, set of operations (*OP*) can be defined that can be used to describe constraints and rules within formulas.

Table 1 describes our concept for representing an enterprise architecture in a hypergraph structure (Molnár 2013). *Figure 1* presents our high-level description about the overall construction and the components that will be transformed into hypergraph structure. The description builds on the ISO/IEC/IEEE 42010 software and systems engineering international standard for architecture description. *Figure 1* can be interpreted as follows: An enterprise architecture can be described using the TOGAF content metamodel. 1) An enterprise architecture contains *architecture layers* (has-a relationship). Business Architecture, Data Architecture, Application Architecture and Technology Architecture are architecture layers (is-a relationship). An architecture layer contains metamodel entities, artifacts, viewpoints and other architecture layers (has-a relationship). 2) An enterprise architecture contains *metamodel entities* (has-a relationship). Actual entities are metamodel entites (is-a relationship). Metamodel entities has relationships (has-a relationship). 3) An enterprise architecture contains *relations* between metamodel entities (has-a relationship). Actual relationships are relationships (is-a relationship). A relationship contains metamodel entities (has-a relationship). 4) An enterprise architecture contains *artifacts* (has-a relationship). Actual models are artifacts (is-a relationship). An artifact contains metamodel entities, artifact type and other artifacts (has-a relationship). 5) An enterprise architecture contains *artifact types* (has-a relationship). Catalogs, matrices and diagrams are artifact types (is-a relationship). 6) An enterprise architecture contains *viewpoints* (has-a relationship). Actual viewpoints are viewpoints (is-a relationship). A viewpoint contains artifacts, metamodel entities, artifact types and other viewpoints (has-a relationship).

Constraints for building a hypergraph-based representation of enterprise architecture include the following: 1) An *enterprise architecture* consists of *architecture layers*. An *architecture layer* may connect to other *architecture layers*. An *architecture layer* contains *artifacts*. An *architecture layer* contains *metamodel entities*. An *architecture layer* contains *metamodel relations*. An *architecture layer* contains *viewpoints*. 2) An *enterprise architecture* consists of *metamodel entities*. A *metamodel entity* may connect to other *metamodel entities*. A *metamodel entity* connects to an *architecture layer*. 3) An *enterprise architecture* consists of *artifacts*. An *artifact* contains *metamodel entities*. An *artifact* contains *metamodel relations*. An *artifact* belongs to a disjoint *artifact type*: catalog/matrix/diagram. 4) An *enterprise architecture* consists of *viewpoints*. A *viewpoint* contains *artifacts*. A *viewpoint* may connect to other *viewpoints*.

Figure 2 can be interpreted as follows: An enterprise architecture consists of enterprise architecture models. Models belong to an architecture layer: Business Architecture Layer, Data Architecture Layer, Application Architecture Layer and Technology Architecture Layer. Artifacts of the enterprise architecture are handled as documents, and consist of the same groups as the enterprise architecture, namely the Business Architecture model base, the Data Architecture model base, the Application Architecture model base and the Technology Architecture model base. To provide transition between artifacts and enterprise architecture models, the layers and the model base classifications are linked. Finally, documents can be in different states: Free/ Finalized/ Ground/ Intensional. Document states can be connected to the artifacts.

4 CASE STUDY

This part of the paper presents a case study for utilizing the hypergraph-based approach to analyse strategic misalignment.

While organisations address alignment achievement, they are continually suffering from misalignments. These difficulties (the misalignments) encumber the achievement of alignment.

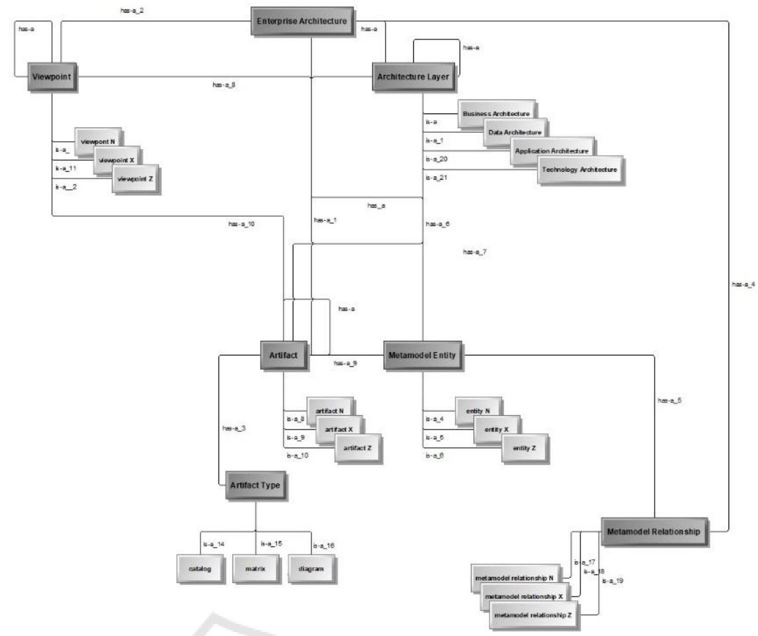

Figure 1: Conceptual Model for Enterprise Architecture Description.

Misalignment analysis (detecting, correcting and preventing misalignment) is an important step in achieving alignment since it helps to understand the nature and the barriers of alignment. Our conceptual analysis relates to the concept of strategic alignment, and aims to approach strategic alignment from the perspective of misalignment. The problem of revealing the typical symptoms of misalignment will be addressed in order to assess the state of alignment in an organisation.

4.1 Organisational Context

The empirical investigation focuses on a road management authority. The study was carried out in a fragment of the road management authority's EA model structure. It describes a road control initiative, showing the relevant EA models and artifacts to be modified during the progression of the project.

The road management authority is a non-profit government corporation that handles matters relating to road safety, road traffic management and transportation for around 32000 kilometers national public road network. The scope of activities spans from road operation and road maintenance over professional services to providing road information. In its actual form the authority was set up in 2006 as a successor of a previous road management government authority. The head quarter and three sites are located in Budapest, and the authority has approx. 170 branches around Hungary. In 2016 the authority employed around 8200 employees.

Road control initiative is a pilot project for setting up the EA practice in the authority. The initiative is part of an integrated road network development project which aims to transform the internal operation as well as to optimize processes in order to increase operational efficiency and transparency within the road management authority. As part of the above introduced integrated road network development project, the road control project is concerned with the implementation of a traveling warrant system. The goal of the project was manifold: to achieve real-time road control information forwarding, to deliver up-to-date information and control specifications onboard, to provide exact information retrieval about past activities and coordinates by place and by date, to provide electronic administration about road control, to provide an expandable and integral solution for road control support, to decrease paper administration related to road control tasks. The project was set up to eliminate the following problems related to the previous road control solution: 1) administration overload, 2) too many isolated information systems, 3) slow escalation of road control-related information, 4) non-automated read-in of road control-related data, 5) non-electronic retrieval of previous road control routes and coordinates.

The road control project was set off to outline the process of road control with EA methods over 2 set of changes. The as-is state (OV) presents the actual

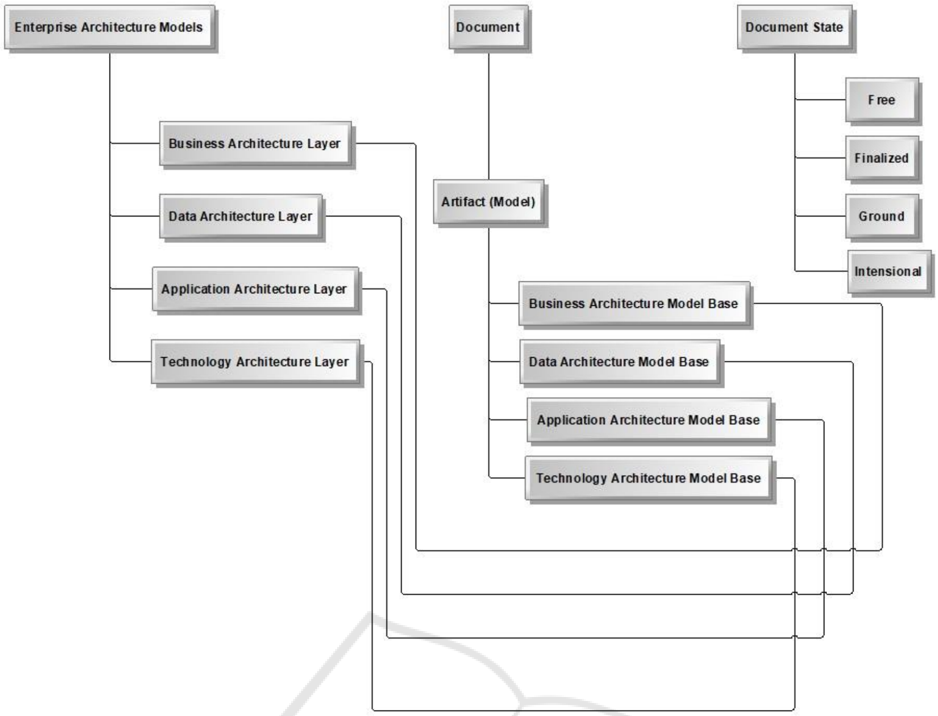

Figure 2: Enterprise Architecture Models Described as Documents.

state of road control activities. To-be No. 1.0 and To-be No. 2.0 phases deal with the changes in process execution, supportive applications and underlying technological infrastructure.

4.2 Hypergraph-based Analysis

Preliminary reviews on the case consisted of the list of influential areas to review and the analysis of assumed malfunctioning areas. It was followed by the categorisation of perceived misalignment symptoms. Non-analysable symptoms were excluded from further analysis. The remaining analysable symptoms were assessed in the case study.

In a former approach (Őri, 2017; Őri and Szabó, 2017) an XML-based analysis tool was created, which detected the symptoms of misalignment with rule assessment techniques. The applied research methodology used an alignment perspective-driven approach. In the first step, traditional alignment perspectives were connected with typical misalignment symptoms. In the second step, relevant artefacts were provided with the misalignment symptoms, i.e. the models which may contain the symptom in question. In the third step, suitable EA analysis types were suggested to the misalignment symptoms. These EA analysis types were able to detect the symptoms in the recommended containing artefacts. Misalignment symptoms, containing artifacts and recommended EA analysis types came from catalogs that were based on recent literature. *Figure 3* illustrates the components of the artifact-based methodology. Discussion on the results of the symptom detection was given by Őri (2017) and Őri and Szabó (2017). To translate the above introduced methodology into the hypergraph-based approach, we need the following concepts: 1) Alignment perspectives: This list contains the corresponding alignment perspective for symptom detection. 2) Misalignment symptom catalog: This list comprises the perceived misalignment symptoms. 3) Artifact catalog: This list encompasses the possible containing EA models. 4) EA analysis catalog: This list includes the possible EA analysis types to recommend. 5) Presence in the artifact: This concept describes the sign of the symptom in the EA models.

6) Occurrence on model entity level. This concept defines how the symptom is manifested on model entity level. 7) Occurrence in XML model export: This item describes how the symptom is manifested in the XML export of the EA model. 8) XML-based query.

The hypergraph-based description, representation of EA models provides extended opportunities for analysis based on a formalized, solid conceptual framework. While the original rule-based framework was appropriate to discover the misalignment

Table 1: Representation of Enterprise Architecture by Hypergraph.

Concept of Enterprise Architecture	**Representation of concept in the domain of hypergraph theory**
Enterprise Architecture	The outcome of a project dedicated to description and development of Enterprise Architecture. The building blocks of the architecture are realized in artifacts. The components of artifacts can be represented as hyperarcs, generalized hyperedges, and the atomic elements as vertices.
Information System (IS)	A result of a system-development exercise that created a set of design artifacts. The set of elements and a *relationships* among them can be represented as nodes and edges within the graph. We can map the model elements to a *hypergraph* that consists of nodes and hyperedges.
Node/vertex in a hypergraph	Each vertex corresponds to an element within the content metamodel, e.g. architecture layers, metamodel entities, entities of architecture layers (constituting a tree structure), etc. The documents may represent one of the aspects for the information flow both inwards and outwards.
Edge in a hypergraph	*Edge* is a specific *hyperedge* with cardinality equal to two. Edge denotes binary relationships between two nodes, as e.g. a descriptive relationship between two metamodel entities, an entity belongs to an architecture layer, an architecture layer connects to another architecture layer, etc.
Hyperedge	A hyperedge represents a relationship among a subset of nodes as e.g. entities belonging to a specific architecture layer, entities belonging to core content, entities belonging to extension content, etc.
System graph	A hypergraph that includes a disjoint node for modeling the environment of the system, plus all the nodes and hyperedges of the content metamodel.
Sub-system	A subset of nodes and their incident hyperedges. A node/vertex is *incident* to a hyperedge if the hyperedge contains the node/vertex. A sub-system may be composed of core metamodel entities, core content metamodel, extensions, etc.
Interconnecting sub-systems - hyperedges graph of the generalized hypergraph	A graph consisting of all the nodes in a sub-system and all hyperedges connecting together subsystems.

problems represented in the EA models, the results were limited by the quality of the models. By extending the approach using the hypergraph concept a more detailed and formalized description and representation of enterprise architecture can be implemented. This approach enables organizations to explore and analyse misalignment symptoms, but workflow patterns, implementation phases related issues, dynamic changes, broader business and management concepts can also be involved. One of the possible approaches when processes, workflows, data and documents are handled together under the notion of "case" or "case management" (Bouafia et al., 2017) as an overarching concept, even this approach can be dealt with proposed models giving ways to automatized and algorithmic analysis.

5 CONCLUSIONS

The outlined approach provides the opportunity to make use of formal and mathematical based analytic methods for discovering misalignment among IT strategies, existing Information Systems, Information Architecture.

The Business Processes and Workflows are immanent component of the Enterprise Architecture so that the integrity, coherence, and consistency with the other elements of Enterprise Architecture is critical. The Information Architecture that contains the representation of data in entity format and documents should fit to the Business Processes (Business Architecture). The hypergraph modeling and representing offers the chance to check and control the discrepancies in a complex enterprise architecture model base. The approach outlined in

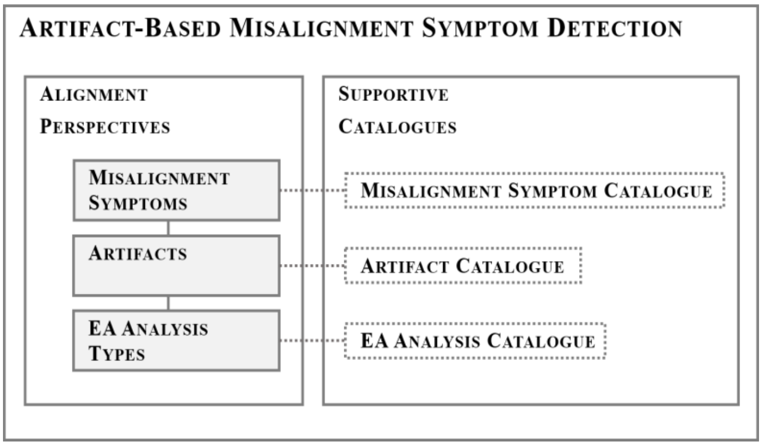

Figure 3: Artifact-Based Method for Misalignment Symptom Detection.

this paper can be mapped onto an appropriate hypergraph structure. The graph theoretical tool set permits the complex analysis of the represented model, e.g. exploiting the graph mining algorithms. There is an ongoing project for developing an adequate hypergraph database that fits to modelling both Information Systems and Enterprise Architecture (Iordanov, 2010; Béleczki, 2016). The next step of the research is to provide a more detailed description about the hypergraph-based formalism, including labeled hyperedges and formalized constraints for defining an enterprise architecture.

ACKNOWLEDGEMENTS

This work was partially supported by EFOP-3.6.3-VEKOP-16.

Supported by the ÚNKP-17-4 New National Excellence Program of the Ministry of Human Capacities.

REFERENCES

A. Ramos, J.P. Sáenz, M. Sánchez, and J. Villalobos, 2015. "On the Support of Automated Analysis Chains on Enterprise Models" Enterprise, Business-Process and Information Systems Modeling. Springer International Publishing, pp. 345-359.

Ahlemann, F. Stettiner, E., Messerschmidt, M., Legner, C., 2012. "Strategic enterprise architecture management: challenges, best practices, and future developments", Springer Science & Business Media.

Bent, H. V. D., Sante, T. V., Kerssens, D., & Kemmeren, J. 2008. TOGAF, the open group architecture framework. Zaltbommel: Van Haren Publishing. Retrieved from http://www.opengroup.org/togaf/

Béleczki, A; Molnár, B. 2016. "Modeling framework for designing and analyzing document-centric information systems based on HypergraphDB" CEUR Workshop Proceedings (ISSN: 1613-0073) 2046: pp. 17-22.

Iordanov, B. 2010. Hypergraphdb: a generalized graph database, in: Web-Age Information Management. pp. 25-36, Springer Berlin Heidelberg

Bouafia, Khawla; Molnár, Bálint; Khebizi, Ali,A 2013. Functional approach for transformation to abstract specifications for Web services from BPEL Programs Characteristics of the approach, winter school, at pecs – Hungary , February 2017.

Bretto, A, Hypergraph theory: An introduction. Berlin: Springer, pp. 111-116.

Choi, J., Nazareth, D. L., Jain, H. K. 2013. "The impact of SOA implementation on IT-business alignment: A system dynamics approach", ACM Transactions on Management Information Systems (TMIS), 4(1), 3.

Earl, M.J. 1989. Management Strategies for Information Technology, London: Prentice-Hall.

G. Carvalho and P. Sousa, 2008. "Business and Information Systems MisAlignment Model (BISMAM): An Holistic Model leveraged on Misalignment and Medical Sciences Approaches" Proceedings of the Third International Workshop on Business/IT Alignment and Interoperability (BUSITAL'08), CEUR, vol. 336, CEUR-WS, Aachen, pp. 104-119.

H. Holm, M. Buschle, R. Lagerström, M. Ekstedt, 2014. "Automatic data collection for enterprise architecture models" Software & Systems Modeling, 13(2) pp. 825-841.

H.M. Chen, R. Kazman, and A. Garg, 2005. "BITAM: An engineering-principled method for managing misalignments between business and IT architectures" Science of Computer Programming, 57(1), pp. 5-26,

Hanschke, I. 2009. Strategic IT management: a toolkit for enterprise architecture management. Springer Science & Business Media.

J. Ullberg, U. Franke, M. Buschle, and P. Johnson, 2010. "A Tool for Interoperability Analysis of Enterprise Architecture Models using Pi-OCL" Enterprise Interoperability IV. Making the Internet of the Future

for the Future of Enterprises, Springer, London, pp. 81-90.
J.C. Henderson and N. Venkatraman, 1993. "Strategic Alignment: Leveraging information technology for transforming organizations" IBM Systems Journal, 32(1), pp. 4-16.
K. D. Niemann, 2006. "From enterprise architecture to IT governance" (Vol. 1). Springer Fachmedien.
K. Pessi, T. Magoulas, and M.A. Hugoson, 2011. "Enterprise Architecture Principles and their impact on the Management of IT Investments" The Electronic Journal of Information Systems Evaluation, 14(1), ECIME 2010 Special Issue, pp. 53-62.
Kossak, F., Illibauer, C., Geist, V., Natschläger, C., Ziebermayr, T., Freudenthaler, B., Schewe, K. D. Hagenberg 2016. Business process modelling method (pp. 1-255). Springer.
Lankhorst, M. 2013. "Enterprise Architecture at Work. Modelling, Communication and Analysis", Springer, Heidelberg.
Molnár, B, Benczúr, A.. 2013. Facet of modeling web IS from a document-centric view. International Journal of Web Portals (IJWP), 5(4), 57-70. DOI: 10. 4018/ijwp. 2013100105
Molnár, B., Béleczki, A., & Benczúr, A. 2017. Information systems modelling based on graph-theoretic background. In: Journal of Information and Telecommunication, 10.1080/24751839.2017.1375223, pp. 1-23. Taylor & Francis.
Őri, D. 2017. On exposing strategic and structural mismatches between business and information systems: Misalignment symptom detection based on enterprise architecture model analysis, PhD Thesis, May 2017
Őri, D., Szabó, Z. 2017. Pattern-Based Analysis of Business-IT Mismatches in EA Models: Insights from a Case Study. In: Sylvain Hallé; Remco Dijkman; James Lapalme (Eds.) Proceedings of the 2017 IEEE 21st International Enterprise Distributed Object Computing Conference workshops and demonstrations (EDOCW 2017), pp. 92-99, 2017
Peppard, J., and Ward, J. 2016. Strategic Planning for Information Systems, 4th Ed., Chichester, John Wiley and Sons Ltd.
R. Wagter, H.A. Proper, and D. Witte, 2012. "A Practice-Based Framework for Enterprise Coherence" PRET 2012. LNBIP 120, Springer, Heidelberg, pp. 77-95.
S. Purao and K. Desouza, 2010. "An enterprise-wide intervention at IRS: A longitudinal analysis of stakeholder sentiments" International Workshop on Trends in Enterprise Architecture Research, Springer Berlin Heidelberg, pp. 30-43.
Strnadl, C. F. 2006. "Aligning business and it: The process-driven architecture model", Information systems management, 23(4), 67-77.
TOG: The Open Group, 2015. TOGAF Version 9. The Open Group Architecture Framework (TOGAF). http://theopengroup.org/.
Versteeg, G. and Bouwman, H. 2006. "Business architecture: A new paradigm to relate business strategy to ICT", Information systems frontiers, 8(2), pp. 91-102.
Y.E. Chan and B.H. Reich, 2007. "State of the Art. IT alignment: what have we learned?" Journal of Information Technology, 22(4), pp. 297-315.
Zachman, J. A. 1987. A framework for information systems architecture. IBM Systems Journal, 26(3), 276–292.