A Hypergraph Based Formal Description Technique for Enterprise Architecture Representation

Dóra Őri Ph.D.
Department of Information Systems
Corvinus University of Budapest
Budapest, Hungary
dora.ori@uni-corvinus.hu

Bálint Molnár Ph.D.
Department of Information Systems
Eötvös Loránd University
Budapest, Hungary
molnarba@inf.elte.hu

***Abstract*—This paper provides a description about a hypergraph-based formalism for representing enterprise architecture, including formalized constraints for enterprise architecture definition. The paper describes a solid concept using the TOGAF Content Metamodel and the hypergraph theory. Hypergraphs provide a flexible mathematical structure to describe complex relationships in an enterprise architecture, mirroring the dependencies among components, and exploring integrity and consistency issues. The approach outlined in this paper offers extended analytical potential for discrepancy checking in complex enterprise architecture structures utilizing the tool set of hypergraph theory. The proposed approach can be used for extending EAM-based analysis of business information systems.**

Keywords—Enterprise Architecture; TOGAF; Artefact; Formal Modelling; Hypergraph

I. INTRODUCTION

The main goal of information strategy/information technology (IT) planning, as part of the broader concept of IT governance is the alignment and harmonization of information systems (IS) and business plans. In the present era of digitalization, growing organizational complexity, emerging disruptive technologies and rapid technological changes make strategic planning of IS reasonably challenging. Although traditional planning approaches [19] are still in use, enterprise architecture management (EAM) approaches are the major facilitators of planning dynamisms nowadays [9]. Information strategy planning is a complex and especially important activity of organizations, an exercise through which an organization utilizes its technological resources [8]. Information strategy is a major tool to integrate IT opportunities and concerns into business planning, harmonizing business and IT domains. Several methods can be used for the harmonization of IT initiatives with business goals (e.g. co-engineering initiatives), 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. New IT-based innovative models increase the pressure for organizational changes, in addition, legal constraints, compliance requirements and ethical issues make planning activities more complicated. Strategic harmonization of business and IT domains is more relevant than ever before, and EAM can be a major facilitator of strategic alignment by detecting, analysing and preventing misalignment problems while achieving competitiveness [29]. EAM provides appropriate tools and methods to reduce organizational complexity [26], promote agility and control uncertainties [7]. EAM helps 1) the alignment of the organisation with strategic goals, 2) the control of interdependencies in business and IT, and 3) it enables organisations to agility and fast reaction. Strategic EAM facilitates strategy formulation too, by analyzing the as-is (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 and by monitoring architecture evolution [13], [1].

The complexity of IT and IS infrastructures amplifies the need of enterprise architecture modeling within organizations. There are several approaches of Enterprise Modeling [27], [31] that provides opportunities for semi-formal modeling by 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 complex relationships between the building blocks of architecture, the representation by a formal approach requires a flexible descriptive method in which there are no restrictions on enhancing and extending the representation with new types 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. In this paper a hypergraph-based model will be introduced to provide a representation for an enterprise architecture structure.

The paper presents the formal foundations of hypergraph-based representation of EA. It also introduces the constraints for describing an EA. In a practical context, the paper aims to approach the topic of EAM-based analysis of misalignment problems by proposing the extension of an existing, EAM-based method. This EAM-oriented model will be extended with the hypergraph-based approach. A motivating example will illustrate how the existing model can be transformed into the hypergraph-based representation.

doi: 10.1109/IIAI-AAI.2018.00164. Copyright © 2018, IEEE

To discuss the contribution of the proposed hypergraph-based model, a detailed description will be given on how the extended model promotes co-engineering of business needs and IT services.

The rest of the paper is structured as follows: Section 2 summarizes the theoretical context. Section 3 introduces our hypergraph-based approach, including the mathematical background, a motivating example, architecture constraints and our proposition for the hypergraph-based model. Section 4 discusses our contribution to co-engineering principles. At the end of the paper conclusions are drawn.

II. Theoretical Background

The theoretical foundation of the paper consists of 3 parts. Firstly, alignment and misalignment assessment approaches will be summarized. Secondly, enterprise architecture analysis methods will be presented. Thirdly, the topic of co-engineering is introduced.

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 [10]. In the literature there are several examples of alignment assessment methods [5]. There are also a few misalignment models mentioned in the literature [4], [6]. These models 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. [4]. Symptoms can be found e.g. in EA models. However, other kinds of sources can also be used in the analysis (see [21]).

Enterprise architecture (EA) is the construction of an enterprise, described by its entities and their relationships. EA is an organizing logic for business processes and IT infrastructure in order to review, maintain and control the whole operation of an enterprise [12], [31]. Enterprise architecture management is a management philosophy concerned with corporate change. EAM provides several benefits by improving IT efficiency (reducing redundancy, ensuring homogeneity, etc.); by enabling IT effectiveness (ensuring goals, strategy and means conformity, etc.); by improving IT reliability (reducing risk) [16]. An enterprise architecture framework is a collection of descriptions and methods to create and manage EA. The most recognized frameworks are the Zachman Framework and the TOGAF framework [27], [31]. 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 [27]. Enterprise architecture analysis types are methods that are capable of assessing EA models, e.g. evaluating dependencies, isolated objects, complexity or heterogeneity. [28] introduced general process models for EA analysis. Several authors (e.g. [30]) proposed EA analysis collections. Tools for supporting the process of EA analysis are expounded by e.g. [23]. Automated collection of EA models is supported by e.g. [11].

Service Agility is an important requirement against the dynamic changes in the economic environment. Adaptive Complex Enterprise (ACE) frameworks supply business and IT related skills to advance business processes and the underlying complex IT architecture [22]. ACE provides a holistic, conceptual perspective on the complex interactions of services in the organizations. Co-engineering serves as an analytical aid for analyzing and improving ACE performance. In co-engineering Business, IT, Operations and Strategy (BioS) performance are improved at the same time. As a result, co-engineering initiatives provide a complex roadmap for achieving strategic goals and covering several business aspects of the organization (e.g. value creation, risk management). Co-engineering principles describe the practical goals of simultaneously engineered BioS performance. Co-engineering can be used for several purposes, e.g. strategic improvement of non-routine services, promoting business need and IT services convergence, using co-engineering for achieving lean-related objectives in IT services, relating IT to service innovation [24], [22]. In our research co-engineering theory will be used for better relating business needs and IT services.

III. A Hypergraph-Based Approach

The following section describes the mathematical background, provides a motivating example and presents the architecture formalism for hypergraph description, including our concept for hypergraph-based architecture description and the evaluation of the formalism using the running example.

A. Mathematical Background

In the following section the mathematical background of hypergraph theory will be introduced. 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 [3].

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,

doi: 10.1109/IIAI-AAI.2018.00164. Copyright © 2018, IEEE

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 [15]. A hypergraph is a generalization of an ordinary graph where edges, called hyperedges, can connect any number of nodes. Formally, let $G(V,E)$ denote a hypergraph, where V denotes a finite set of nodes v, E denotes the set of hyperedges e. Each hyperedge $e \in E$ is a subset of V.

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. The tail and the head of hyperarc E can be denoted by $T(E)$ and $H(E)$, respectively, or alternatively a hyperarc $\vec{e}_i \in \vec{H} = (V; \vec{E} = \{\vec{e}_i | i \in I\})$ can be perceived as an ordered pair $\vec{e}_i = \left(\overrightarrow{e_i^+} = (e_i^+; i); \overrightarrow{e_i^-} = (e_i^-; i)\right)$, where $e_i^+ \subseteq V$ is the set of vertices of $\overrightarrow{e_i^+}$ and $e_i^- \subseteq V$ is the set of vertices $\overrightarrow{e_i^-}$. The elements of $\overrightarrow{e_i^+}$ (hyperedges and/or vertices) are called *tail* of $\vec{e}_i$, while elements of $\overrightarrow{e_i^-}$ are called *head* [3].

Definition 4. Limb is either the head or the tail of a hyperarc and designated by $\vec{e}_i^{\mp}$.

Definition 5. Architecture Describing Hypergraph is a generalized hypergraph that can be extended by some functions and operations. It can be designated as a tuple $\langle V, A, E, E_U, E_D, Attr \rangle$:

V is the set of *vertices*;

A is the set of arcs, i.e. directed edges, an arc is an ordered pair $\langle i,j \rangle$, where $i,j \in V$ (in the sense of traditional graph theory);

E is the set of hyperedges;

$\boldsymbol{E}_U$ is the set of the *undirected* hyperedges, because of the properties of generalized hypergraphs, a hyperedge *e* is either $e_i \neq \emptyset, e \subseteq V$, (*basic hyperedge*), or a *bag* of hyperedges;

$\boldsymbol{E}_U$ is divided up at a meta-level into partitions:

$\boldsymbol{E}_C$ consists of the *configuration hyperedges*. Each $h_i \in \boldsymbol{E}_C$ is a simple hyperedge, i.e. containing only vertices, not complex structures and other hyperedges. All $h_i \in \boldsymbol{E}_C$ can be labeled unambiguously. The configuration hyperedges manifests the structure of "things", the vertices within a hyperedge are the properties of the specific "thing". The properties can be perceived as variables or attributes (depending on the context) that can be valuated thereby they linked to an individual value (vertex in *D*) or a set of values, e.g. to a grouping hyperedge.

$\boldsymbol{E}_E$ is composed of the e*xtensional hyperedges*. The extensional hyperedges can represent collections of data, the instances of *generic artifacts*. For example, the collections of data can be built up by tuples of data items, the instances of artifacts can be composed of certain bags of free variables that are contained in the particular artifacts' object/element structure. In these examples, the distinct elements, the vertices of these hyperedges can be considered as constituents of extensional hyperedges.

$\boldsymbol{E}_I$ comprises the *intensional hyperedges*. The intensional hyperedges show the logical and rule-based interrelationships among the vertices (models within the architecture).

$\boldsymbol{E}_G$ is made up of *grouping hyperedges* that embody various structuring principles on components, as e.g. view, viewpoint and perspectives etc. in architecture describing approaches; they symbolize interrelationships between certain models and pieces or parts of documents as e.g. business activity models, business process models, catalogues of architecture building blocks and responsibilities of roles within an organization unit. The hyperedge $h \in \boldsymbol{E}_G$ can be utilized for sorting out the vertices (representing either artifacts, viewpoints, or models) into organizational-related, artifact-related and activity related relationships.

$\boldsymbol{E}_D$ is a set of *hyperarcs*, i.e. *directed hyperedges;* the hyperarc $\vec{e}_i \in \boldsymbol{E}_D$ can be as it follows (see **Definition 2**.):

$\vec{e}_i = \langle v_j, \vec{h} \rangle = \left(\vec{e}_i^+ = \left(e_i^+ = \langle v_j, i \rangle\right); \vec{e}_i^- = (\langle i, e_i^- = h \rangle)\right)$

where $v_j \in V$, and $\vec{h} \in \boldsymbol{E}_G$;

$\vec{e}_i = \langle v_j, \vec{h} \rangle = \left(\vec{e}_i^+ = \left(e_i^+ = \langle v_j, i \rangle\right); \vec{e}_i^- = (\langle i, e_i^- = h \rangle)\right)$,

$v_j \in V$, and $\vec{h} \in \boldsymbol{E}_C$;

$\vec{e}_i = \langle v_j, \vec{h} \rangle = \left(\vec{e}_i^+ = \left(e_i^+ = \langle v_j, i \rangle\right); \vec{e}_i^- = (\langle i, e_i^- = h \rangle)\right)$

where $v_j \in V$, and $\vec{h} \in \boldsymbol{E}_E$; there does not exist two hyperarcs $\vec{e}_i = \langle v_j, \vec{h} \rangle$ and $\vec{e}_k = \langle v_l, \vec{h'} \rangle$ that either $\vec{h}, \vec{h'} \in E_C$ *or* $\vec{h}, \vec{h'} \in E_E$, i.e. every vertex $v_j \in V$ is linked, at most, to one configuration hyperedge ($\boldsymbol{E}_C$) and at most to one extensional hyperedge ($\boldsymbol{E}_E$). These conditions can be interpreted the following way: a vertex may belong to a configuration structure (either artifact or model), or it may belong to an extension that represents the instantiation of either an artifact or a model.

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

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

$source_E: E \longrightarrow V$, $target_E: E \longrightarrow V$

these functions return the source and target vertices of an edge *E;*

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

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

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

D can be grasped 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.

doi: 10.1109/IIAI-AAI.2018.00164. Copyright © 2018, IEEE

B. Motivating Example

This part of the paper presents a motivating example for utilizing the hypergraph-based approach to describe an enterprise architecture and analyze strategic misalignment. While organizations address alignment achievement, they are continually suffering from misalignments. These difficulties (the misalignments) encumber the achievement of alignment. 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. The problem of revealing the typical symptoms of misalignment will be addressed in order to assess the state of alignment in an organization. The running example 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. Road control initiative is a pilot project for setting up the EA practice in the authority. The project was set off to outline the process of road control with EA methods over 2 set of changes. The as-is state presents the actual state of road control activities. To-be phases deal with the changes in process execution, supportive applications and underlying technological infrastructure.

C. Architecture Constraints for Hypergraph Representation

An enterprise architecture can be described using the TOGAF content metamodel. Constraints for building an enterprise architecture include the following: 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. An enterprise architecture consists of metamodel entities. A metamodel entity may connect to other metamodel entities. A metamodel entity connects to an architecture layer. 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. An enterprise architecture consists of viewpoints. A viewpoint contains artifacts. A viewpoint may connect to other viewpoints.

D. Our Concept for Hypergraph-Based Architecture Description

Using the above described approach for enterprise architecture formalization, we now present our concept for a hypergraph-based architecture description [14]. The Enterprise Architecture at a given organization can be perceived as an overarching artifact that contains all potential artifacts and represents the architecture in a comprehensive way. This overarching conceptual artifact consists of generic artifact types that comprise artifact types. The hierarchical relationships can be described by configuration hyperedges (EC); the instances of a member of a hierarchy that is made of either generic artifact types or artifact types can be perceived as extensions and can be represented by extensional hyperedges. Thereby, this way of representation makes possible for a recursive definition of artifact types and grouping them into a generic artifact type.

A *generic artifact type* ***GAT*** is a *hierarchy of artifact types* ***ATH***. The elements of ***ATH*** can belong to a configuration hyperedge $e_{C_i} \in \boldsymbol{E_C}$ as vertices. The generalized hypergraphs allow that the vertices may appear as complex structures, as hyperedges. Therefore, a node can be a hyperedge that itself a configuration hyperedge that contains a hierarchy of document types. Thereby, the representation makes possible for a recursive definition of document types and gathering them into a generic document type.

In our concept an EA builds on the TOGAF metamodel in content and structure. In the following we first conceptualize the elements and the relationships of the metamodel, then we move to defining artifacts.

Definition 6. The elements of a Metamodel Entity in an Enterprise Architecture are represented in hypergraph by hyperedge m_i where $m_i \in \boldsymbol{E_{mm\,ent}} \subseteq \boldsymbol{E}$ the set of hyperedges. The elements of m_i are the vertices that represent metamodel entities that will describe an artifact. In the Metamodel Entity hyperedge metamodel relationships enmesh the set of metamodel entities by using directed hyperedges.

According to our model, an artifact is composed of a set of metamodel entities and the related overarching metamodel relationships.

Definition 7. The elements of an Artifact in an Enterprise Architecture are represented in hypergraph by hyperedge a_i where $a_i \in \boldsymbol{E_{art}} \subseteq \boldsymbol{E}$ the set of hyperedges. The elements of a_i are the vertices that represent a set of metamodel entities that describe the artifact in the TOGAF Metamodel. Overarching metamodel relationships that enmesh the set of metamodel entities are also connected to an artifact hyperedge by using directed hyperedges. After defining the concept of artifacts, we now turn to build the different grouping options for dimensioning, structuring and categorizing the artifacts, using viewpoints, architecture layers and artifact types.

We start the description of grouping options with defining Viewpoints, which serve as a container of artifacts to create a specific perspective of the architecture for a specific stakeholder group.

Definition 8. The elements of a Viewpoint in an Enterprise Architecture are represented in hypergraph by hyperedge h_i where $h_i \in \boldsymbol{E_{viewp}} \subseteq \boldsymbol{E}$ the set of hyperedges. The elements of h_i are the vertices that represent artifacts that each of them describes a single model.

Artifacts can be parts of several Viewpoints at the same time. After defining Viewpoints, we now turn to the dimensioning option of EA by describing the concept of Architecture Layer in hypergraph-based manner.

Definition 9. The elements of an Architecture Layer in an Enterprise Architecture are represented in hypergraph by hyperedge l_i where $l_i \in \boldsymbol{E_{layer}} \subseteq \boldsymbol{E}$ the set of hyperedges.

doi: 10.1109/IIAI-AAI.2018.00164. Copyright © 2018, IEEE

The elements of l_i are the vertices that represent artifacts that each of them describes a single model.

Like viewpoints, architecture layers provide similar multi-containing possibilities. To complete the definition of grouping options, we now specify Artifact Type, which provides categorization for the different kind of artifacts.

Definition 10. The elements of an Artifact Type in an Enterprise Architecture are represented in hypergraph by hyperedge at_i where $at_i \in \boldsymbol{E}_{art_type} \subseteq \boldsymbol{E}$ the set of hyperedges. The elements of at_i are the vertices that represent artifacts. An artifact has a single artifact type: at_c stands for *catalog* artifact type, at_m stands for *matrix* artifact type, while at_d stands for *diagram* artifact type.

Unlike viewpoints and architecture layers, the sets artifact types are disjoint, i.e. every artifact has only one artifact type: catalog or matrix or diagram. Finally, we define relationships between the components of an EA. In the proposed concept has-a and is-a relationships cover the whole architecture.

Definition 11. We refer to relationships as a hyperedge $r_i \in \boldsymbol{E}_{rel} \subseteq \boldsymbol{E}$ that contains directed hyperedges r_{is-a} and r_{has-a}. These hyperedges contain hyperedges (the components that we will connect to each other) on both the head and the tail parts. The relationship is composed by defining edges between the head and the tail.

E. Hypergraph-Based Misalignment Detection

In this section we use the motivating example to utilize the proposed hypergraph-based approach for analysing strategic and structural misalignment problems in an EA structure. Preliminary reviews provided us with assumed malfunctioning areas. This step was followed by a categorisation of perceived misalignment symptoms. Non-analysable symptoms were excluded from further analysis. The remaining analysable symptoms were assessed in the case. An exemplary list of misalignment symptoms under analysis consist of the following items: 1) Undefined organizational mission, strategy and goals; 2) Lack of relation between process goals and organizational goals; 3) Application functionality does not support at least one business process task; 4) Critical business process does not depend on scalable, available applications. The sample symptoms stem from previous literature on misalignment symptoms [4], [20], [25].

In a former approach [17], [18] an XML-based analysis tool was created, which detected the symptoms of misalignment with rule assessment techniques. Traditional alignment perspectives were connected with typical misalignment symptoms. Next, relevant artefacts were provided with the misalignment symptoms, i.e. the models which may contain the symptom in question. Subsequently, 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. 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.

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 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 EA can be implemented. One of the possible approaches when processes, workflows, data and documents are handled together under the notion of "case" or "case management" [2] as an overarching concept, even this approach can be dealt with proposed models giving ways to automatized and algorithmic analysis.

IV. Contribution to Co-Engineering Principles

In this section we describe how the proposed approach would contribute to co-engineering of EA, information systems and business processes in order 1) to co-engineer business need and IT services, 2) to achieve higher alignment between business strategy and IT/IS architecture and 3) to reduce discrepancy level between business and IT/IS areas. The following proposition is primarily based on the co-engineering introduction by [22]. The proposed approach aims the solution-driven improvement of IT services to achieve business efficiency, innovation and agility. The model provides an approach that helps to mirror the understanding of requirements, strategic goals, IT service lifecycle and the needs for underlying technological infrastructure. It aims to support IT decision-making by promoting alignment between business and IT. It also helps to represent deeper levels of complex systems. The model assists pattern generation too: It supports the characterization of negative operational patterns while achieving and sustaining business and IT alignment. The graph-based model represents a general architecture for a goal-based complex system. In the main, our formal, graph-based model would provide an accurate approach to approximate the strategic goals by IT solutions.

V. Conclusions

The approach presented in this paper provides a formal way to use the mathematical analytic methods for exploring misalignment and discrepancies between Business Strategy and Structure and the related IT strategies, Information Systems, Information Architecture. Business Processes and Workflows are inherent components of EAs. The requirements towards integrity, coherence, and consistency with the other elements of EA are crucial. Co-engineering of

doi: 10.1109/IIAI-AAI.2018.00164. Copyright © 2018, IEEE

business needs and IT services is needed, i.e. the Information Architecture that contains the representation of data in entity format and documents should fit to the Business Processes (Business Architecture). The proposed hypergraph-based modeling and representing approach provides the opportunity to check and control the discrepancies in complex EAs. Topics deferred to future examination include a formalized description about the hypergraph-based approach, including the introduction of hyperedge sections, labeled hyperedges, dual hypergraphs and more detailed formalized constraints for defining an enterprise architecture.

ACKNOWLEDGMENT

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.

COPYRIGHT NOTICE

Copyright © 2018, IEEE
D. Ori and B. Molnár, "A Hypergraph Based Formal Description Technique for Enterprise Architecture Representation," *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*, Yonago, Japan, 2018, pp. 799-804, doi: https://doi.org/10.1109/IIAI-AAI.2018.00164.
https://ieeexplore.ieee.org/document/8693350

REFERENCES

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

[2] Bouafia, K., Molnár, B., Khebizi, A.: A Functional approach for transformation to abstract specifications for Web services from BPEL Programs Characteristics of the approach, winter school, at Pecs – Hungary (2017).

[3] Bretto, A.: Hypergraph theory: An introduction. Berlin: Springer. 2013, pp. 111-116. (2013).

[4] Carvalho, G., Sousa, P.: 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. (2008).

[5] Chan Y.E., Reich, B.H.: State of the Art. IT alignment: what have we learned? Journal of Information Technology, 22(4), 297-315 (2007).

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

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

[8] Earl, M.J.: Management Strategies for Information Technology, London: Prentice-Hall (1989).

[9] Hanschke, I.: Strategic IT management: a toolkit for enterprise architecture management. Springer Science & Business Media (2009).

[10] Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging information technology for transforming organizations. IBM Systems Journal, 32(1), 4-16, (1993).

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

[12] Kossak, F., Illibauer, C., Geist, V., Natschläger, C., Ziebermayr, T., Freudenthaler, B., Schewe, K., Hagenberg, D.: Business process modelling method, Springer (2016).

[13] Lankhorst, M.: Enterprise Architecture at Work. Modelling, Communication and Analysis, Springer, Heidelberg (2013).

[14] Molnár, B., Benczúr, A.: 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 (2013).

[15] Molnár, B., Béleczki, A., Benczúr, A.: Information systems modelling based on graph-theoretic background. Journal of Information and Telecommunication, 10.1080/24751839.2017.1375223, 1-23 Taylor & Francis (2017).

[16] Niemann, K.D.: From enterprise architecture to IT governance (Vol. 1), Springer Fachmedien. (2006).

[17] Őri, D.: On exposing strategic and structural mismatches between business and information systems: Misalignment symptom detection based on enterprise architecture model analysis, PhD Thesis (2017).

[18] Őri, D., Szabó, Z.: Pattern-Based Analysis of Business-IT Mismatches in EA Models: Insights from a Case Study, 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).

[19] Peppard, J., Ward, J.: Strategic Planning for Information Systems, 4th Ed., Chichester, John Wiley and Sons Ltd. (2016).

[20] Pereira, C.M., Sousa, P.: Enterprise Architecture: Business and IT Alignment, ACM Symposium on Applied Computing, pp. 1344-1345, ACM, New York (2005).

[21] Purao S., Desouza, K.: 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. (2010).

[22] Ramanathan, J.: Co-Engineering Applications and Adaptive Business Technologies in Practice: Enterprise Service Ontologies, Models, and Frameworks: Enterprise Service Ontologies, Models, and Frameworks. IGI Global (2009).

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

[24] Rebovich Jr, G., White, B.E.: Enterprise systems engineering: advances in the theory and practice. CRC Press (2016).

[25] Sousa, P., Pereira, C.M., Marques, J.A.: Enterprise Architecture Alignment Heuristics. Microsoft Architects Journal 4, 34-39 (2005).

[26] Strnadl, C.F.: Aligning business and it: The process-driven architecture model. Information systems management 23(4), 67-77 (2006).

[27] TOG: The Open Group: TOGAF Version 9. The Open Group Architecture Framework (TOGAF). http://theopengroup.org/ (2015).

[28] Ullberg, J., Franke, U., Buschle, M., Johnson, P.: 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. (2010).

[29] Versteeg, G., Bouwman, H.: Business architecture: A new paradigm to relate business strategy to ICT. Information systems frontiers 8(2), 91-102 (2006).

[30] Wagter, R., Proper, H.A., Witte, D.: A Practice-Based Framework for Enterprise Coherence, In: PRET 2012. LNBIP 120, Springer, Heidelberg, pp. 77-95 (2012).

[31] Zachman, J.A.: A framework for information systems architecture. IBM Systems Journal 26(3), 276–292 (1987).