Modeling framework for designing and analyzing document-centric information systems based on HypergraphDB

András Béleczki
bearaai@inf.elte.hu

Bálint Molnár
molnarba@inf.elte.hu

ELTE Eötvös Loránd University
Budapest, Hungary

Abstract

Using Document-centric Information Systems (*IS*) in an Enterprise is very common nowadays: the *IS*s serves as the basis for storage of data, elements of business processes and they provide a flexible communication protocol between Web-services too (XML-like documents). Designing work-flows, object hierarchies and much more business-related entities can be done by various types of models (UML, BPMN, Petri-nets). The models and their relationships can be represented mostly through the combination of the Zachman and the TOGAF framework [1, 2] in our case. To assist the designing, analyzing, validating and optimizing steps during model creation, we suggest a generic modeling approach based on the generalized hypergraphs. This helps avoid inconsistencies between the models without using any extra transformations or cross-checks [3]. In our proposed designer tool we use the HypergraphDB which is a graph database providing every advantages of the generic hypergraphs [4, 5]. To extend the object-definition and rule formalism we use description logics beside the hypergraph formalism.

1 Introduction

Since Information Systems (*ISs*) become more and more complex in these days, designing, developing or even analyzing it become a lot harder than before. This complexity originates from the shifting paradigm that the structured data-entities appears in the format of documents. The use of various electronic document types is common in organizations. The basic XML provide a semi-structured formalism, however the XML documents may embed unstructured parts as well beside the meta-data that can be considered semi-structured ones. Moreover, the core of an *IS* may contain both structured and semi-structured data collections that are extended by unstructured elements. The documents in both the outer and inner environment of an *IS* are a reflection of the intended and realized data flows that embody the life cycles of data collections in relationships to the overarching organization structure and roles, to the related business processes and work-flows. Within this complex situation, we need a theoretically sound approach that support the modeling and design steps then cross-checking and

In: E. Vatai (ed.): Proceedings of the 11th Joint Conference on Mathematics and Computer Science, Eger, Hungary, 20th – 22nd of May, 2016, published at http://ceur-ws.org

verification of consistency the models that were placed in the architecture. The model refinement and extension are carried out by systematic design principles that are under the supervision of constraints that are deduced from the assumptions of consistency and integrity. The set of models ordered into architecture framework yields support for operational function in the production time of an *IS*.

In Section 2, we introduce these particular models through previous researches in literature, in Section 3 we define some notion for a better understanding about hypergraphs, in Section 4 we explain the chosen database-system for our model and Section 5 provides conclusions and possible future work.

2 Literature and Technical Review

The Web-based and Web Information Systems are the typical examples that make extensive use of the various document formats. The emphasis on Web technologies slowly diminished as the application of Web technology, definitely at user interfaces, became commonplace. A systematic design approach to construct web-based applications is discussed in. The method explained in [6] makes use of semi-structured and interactive documents represented by XML. Another paper presents an approach for a well-founded, concepts-based modeling process for a Web site. For designing of Web Information Systems, Rossi presents a design procedure [7]. There are many frameworks, which help to grasp the complexity of Information Systems, namely the Blokdijk's perception of Information Systems, Zachman ontology and TOGAF, all of them were created for information systems [1, 2, 8].

An Information System supports business processes (*Business Process Modeling, BPM*) within an enterprise and is tightly coupled to other IS usually. A fairly standard way to model business processes is either the application of Business Process Modeling (BPM) methods, or using Petri-nets. The Information Systems can also be perceived as a structure with underlying databases for structured, semi-structured (XML-based, *eXtensible Markup Language*) as well as unstructured documents . The documents play important role at the interface, at interaction level and at core activities of data processing. The integration level and the degree of reconciliation between Business Processes and organization can be analyzed on the base of ontologies and semantic approaches [9]; it provides an approach for validation and safeguarding the relationships between organization and processes within the architecture.

There were some previous papers and researches that tried to put the before-mentioned approaches into a unified framework by essentially semi-formal way [10, 11, 12, 13].

The Enterprise Architecture framework is provided by a mapping across Zachman ontology and TOGAF framework [1, 2]. The Blokdijk's collection of Information System Models yields a structuring guideline [8].

Since our proposed approach of the unified modeling is based on a generalized hypergraph theorem, it induces a need for a storage with this capabilities. From the technological point of view, there are a lot of graph-based database systems. The suggested HypergraphDB is an open-source project which is based on the knowledge management formalism known as directed hypergraphs.

3 Mathematical Background

Hypergraphs. There are several conceptual formalization that are mentioned in other papers [10, 13] which can be described by a set of relationships from individual models (like UML-based class-diagram, work-flows, etc.). Since these models are representing different facets of perception of IS, and they represent a complex system through a set of complex, heterogeneous relationships. This set of relationships can be described by directed hypergraphs; the directed hypergraph applies the same basic notions as the generalized hypergraphs with the extension of direction. In this set we can separate the elements in two sub-sets:

- hierarchical;
- network-like relationships.

The hypergraphs as mathematical structure seems to be suitable for representing the interrelationships among the models, views, viewpoints, perspectives, and the overarching documents and business processes [1, 5].

To gain insight into the hypergraphs we start with the basic definitions in order to apply for depicting the before-mentioned complex relationships.

Definition 1. A *hypergraph H* is a pair of *(V,E)* of a finite set of $V = \{v_1, v_2, ...v_n\}$ and a set E of nonempty subsets of V. The elements of V are called vertices or nodes, the elements of E are called edges [4].

Definition 2. *Generalized* or *extended* hypergraphs. 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 [4].

Considering a document model, a proper document type hierarchy can be interpreted as a ordered sub-set of the hyper-edges. In a document subpart hierarchy, a specific subpart of document may be denoted by a vertex within a particular hyper-edge that describes this document that contains the subpart, although that subpart as a vertex may include a document type hierarchy that can be depicted by a hyperedge.

Definition 3. A *directed hypergraph* is an ordered pair

$$\overrightarrow{H} = \left(V, \overrightarrow{E} = \{\overrightarrow{e_i} : i \in I\}\right); \tag{1}$$

where V is a finite set of vertices and $\overrightarrow{E}$ is a set of *hyperarcs* with a finite index-set I. Every *hyperarc* $\overrightarrow{e_i}$ can be interpreted as an ordered pair

$$\overrightarrow{e_i} = \left(\overrightarrow{e_i^+} = (e_i^+, i), \overrightarrow{e_i^-} = (i, e_i^-)\right); \tag{2}$$

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 $\overrightarrow{e_i^+}$ are called *tail* of $\overrightarrow{e_i}$, while elements of $\overrightarrow{e_i^-}$ are called *head* [4].

The potential implementations of hypergraphs in a hypergraph database allows for linking attributes to vertices, even more to hyperedges. The target domain, namely documents and model of Information Systems within organizations, contains complex n-ary relationships. The hypergraph provides the opportunity to represent recursive construction, to describe logical relations, to store compound structures along with their values and to follow variable lifetimes across various processes. [5, 14, 15]

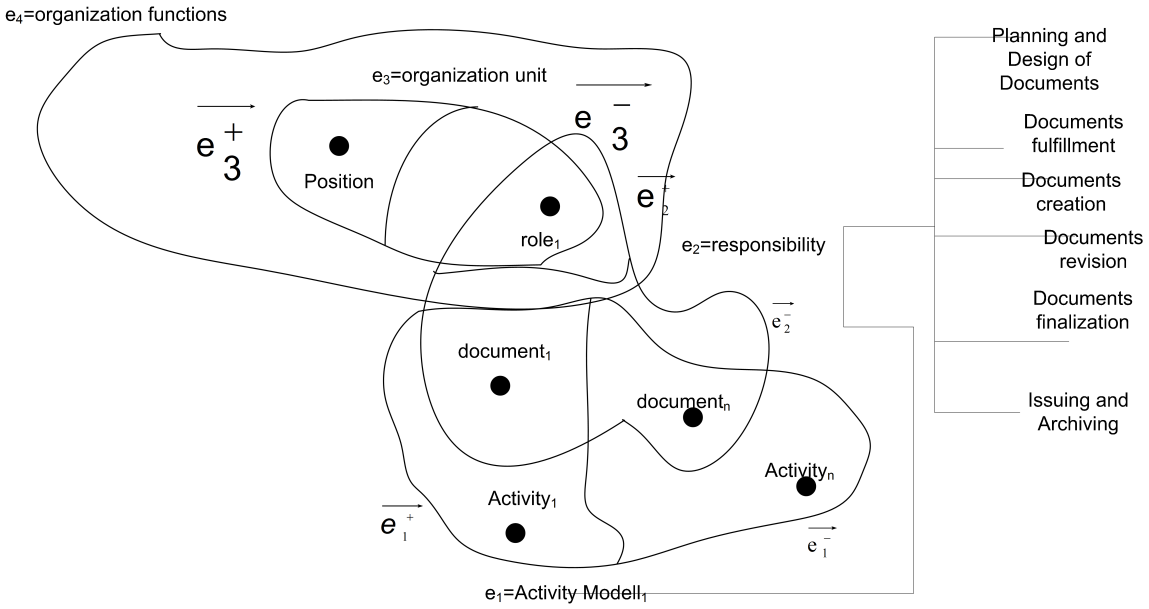

Figure 1: Example for Directed Hypergraph Representing a Sample of Essential Relationships

As an illustration of the basic concepts of directed hypergraph, an example can be seen in Figure 1. that

makes sense of the representation for the domain by hypergraph. The essential characteristics is that vertices contain composite constituents that are themselves may be graphs; generalized hyperedge may contain other hyperedges but not itself and nodes. Detailed description about the Architecture Describing Hypergraph can be found in [16].

4 Using HypergraphDB

The HypergraphDB is an extensible, portable, distributed open-source data-storage mechanism. It is a graph-databased designed specifically for artifical intelligence and semantic web projects, however because of it general mindset, it is a perfect tool to represent heterogeneous relationships between different types too. The following key facts are convincing enough to use the HypergraphDB as tool to store our model [17]:

- The mathematical definition of a hypergraph is an extension to the standard graph concept that allows an edge to point to more than two nodes. HyperGraphDB extends this even further by allowing edges to point to other edges as well and making every node or edge carry an arbitrary value as payload.
- The basic unit of storage in HyperGraphDB is called an atom. Each atom is typed, has an arbitrary value and can point to zero or more other atoms.
- Data types are managed by a general, extensible type system embedded itself as a hypergraph structure. Types are themselves atoms as everybody else, but with a particular role.
- The storage scheme is platform independent and can thus be accessed by any programming language from any platform. Low-level storage is currently based on BerkeleyDB from Sleepycat Software.
- Size limitations are virtually non-existent. There is no software limit on the size of the graph that are managed by a HyperGraphDB instance. Each individual value's size is limited by the underlying storage, i.e. by BerkeleyDB's 2GB limit. However, the architecture allows bypassing BerkeleyDB for particular types of atoms if one so desires.
- The implementation is solely Java based. It offers an automatic mapping of idiomatic Java types to a HyperGraphDB data schema which makes HyperGraphDB into an object-oriented database suitable for regular business applications.

Since there aren't any first-party user interface for the HypergraphDB, the firs step to start using it was to design and develop a middle-ware software which can create complete hypergraphs by creating the appropriate nodes and edges based on various input. These inputs can be mostly XML-based descriptors - like OWL - but can be also some custom, user defined XML schema. In our case, we had a tool - written in C++ using the Qt Framework [18, 19] - which is capable of designing Workflow Models based on Petri-nets. This tool generates a custom XML file consisting of the *places*, *trasitions*, *arcs*, *flow relations*, *presets* and *ofsets* of transitions and all other required data.

To test the capabilities of the database, we created a hypergraph based on a business process. This process was first designed in the above-mentioned tool, then the XML output was passed to the middle-ware. After the middle-ware finished the processing of the XML file it created the necessary nodes and the edges. Also utilizing the HypergraphDB efficiency the nodes and hyperedges can be labeled with custom JAVA classes, therefore the potential of the object-oriented class hierarchy is exploitable.

5 Conclusion and Future Work

There is a bunch of aspects to analyze the relations among different model-types. For every designing step mentioned before [1, 2] there are several models to depict the fairly similar aspects of the given system. This means that there has to be a transformation or mapping function which selects and creates the relevant relationships between the elements of these models. This mappings can be stored as a sub-hypergraph also in the HypergraphDB, so the information that are about an *IS* can be handled in uniform way. The proposed approach for uniform representation of IS from an architectural viewpoint offers the opportunity for united handling of models and exploring the graph theoretical tool sets for further analysis.

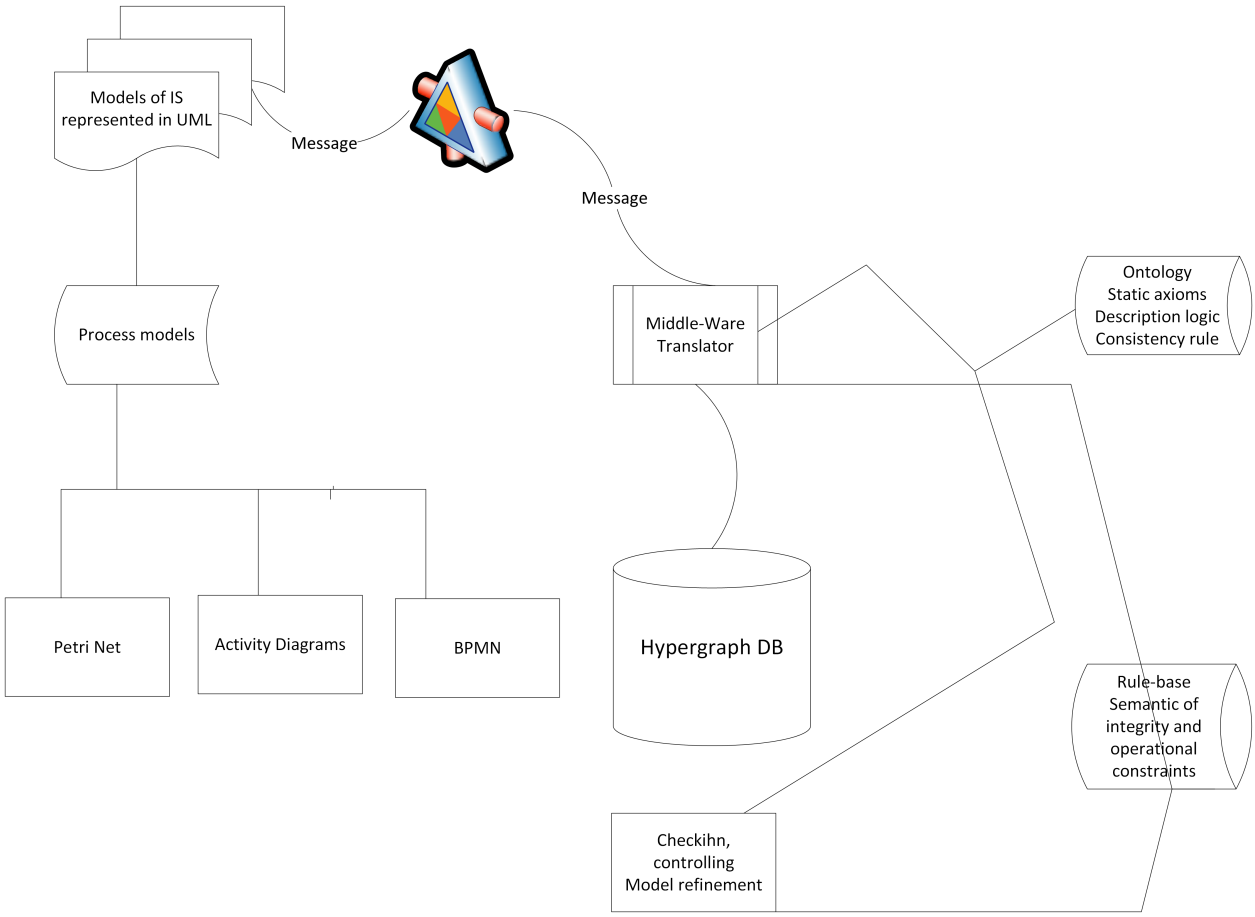

Figure 2: The Framework's Components and their Interactions

References

[1] Zachman, J.A., 1987. A Framework for Information Systems Architecture, IBM Systems Journal Volume, 26, No. 3, pp. 276-292

[2] Open Group, 2010. TOGAF: The Open Group Architecture Framework, TOGAF® Version 9, http://www.opengroup.org/togaf/

[3] Suh, N.P.. 2001. Axiomatic Design: Advantages and Applications. Oxford University Press, New York

[4] Bretto, A. 2013. Hypergraph Theory: An Introduction. Springer.

[5] Gallo, G., Longo, G., Pallottino, S., Nguyen, S. 1993. Directed hypergraphs and applications. Discrete applied mathematics, 42(2), 177-201.

[6] Köppen, E., Neumann, G. , 1999. "Active hypertext for distributed web applications", in: *Proceedings of The Eighth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'99)*, pp. 297—302,

[7] Rossi,G., Schwabe,D., Lyardet,F., 1999."Web application models are more than conceptual models", in: *P. Chen et al. (Ed.), Advances in Conceptual Modeling, LNCS, vol. 1727*, pp. 239—252, Springer-Verlag, Berlin

[8] Blokdijk, A., Blokdijk, P. 1987. Planning and Design of Information Systems, Academic Press, London

[9] Gábor, A., Kő, A., Szabó, I., Ternai, K., Varga, K. 2013. Compliance Check in Semantic Business Process Management, in: *On the Move to Meaningful Internet Systems (OTM)* 2013 Workshops. 353-362. Springer Berlin Heidelberg

[10] Molnár, B. 2014. Applications of hypergraphs in informatics: a survey and opportunities for research. *Ann. Univ. Sci. Budapest. Sect. Comput. 42*, 261–282.

[11] Molnár, B., Tarcsi, A. 2011. Architecture and System Design Issues of Contemporary Web-based Information Systems, in: *Proceedings of the 5th International Conference on Software, Knowledge Information, Industrial Management and Applications* (SKIMA 2011), September 8-11, 2011, Benevento, Italy.

[12] Molnár, B., Benczúr, A. Facet of Modeling Web Information Systems from a Document-Centric View, in: *International Journal of Web Portals (IJWP)*, 5(4), 57-70, 2013, IGI Global

[13] Molnár, B., Benczúr, A., Béleczki, A. 2016. A Model for Analysis and Design of Information Systems based on a Document Centric Approach, in: *Intelligent Information and Database Systems (IIDS)*, 290-299, Springer-Verlag, Berlin

[14] Ausiello, G., Franciosa, P. G., & Frigioni, D. 2001. Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach, in: *Theoretical Computer Science* pp. 312-328, Springer Berlin Heidelberg

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

[16] Molnár B., Benczúr A., Béleczki A., 2016. Formal Approach to Modelling of Modern Information Systems, *International Journal of Information Systems and Project Management*, (to be published)

[17] Kobrix Software. 2010. *HypergraphDB - A Graph Database.* [ONLINE] Available at: http://hypergraphdb.org. [Accessed 27 May 2016].

[18] The Qt Company. 2012. *Qt - Home.* [ONLINE] Available at: https://www.qt.io. [Accessed 27 May 2016].

[19] Stroustrup, B. 1995. The C++ programming language. Pearson Education India