Annales Univ. Sci. Budapest., Sect. Comp. 42 (2014) 261–282

APPLICATIONS OF HYPERGRAPHS IN INFORMATICS: A SURVEY AND OPPORTUNITIES FOR RESEARCH

Bálint Molnár (Budapest, Hungary)

Dedicated to Prof. András Benczúr

Communicated by Péter Racskó

(Received June 1, 2014; accepted July 1, 2014)

Abstract. After brief introduction about hypergraphs and their specific capabilities that make them apt to be applied in various fields of research within information systems, modeling and analytics. In the literature review, we survey the fields that fit to hypergraphs theory and the application of their capability to describe complex relationships. We outline our research on modeling of information systems and architectures, and then we propose a research agenda to exploit the rich representation ability of hypergraphs in the context of various information and IT systems.

1. Introduction

Hypergraphs are able to describe complex systems as their descriptive power is fairly strong because they are one of the most general graph and mathematical structures for representing relationships. The hypergraphs are the generalization and extension of concept graphs and finite sets. The mathematical

Key words and phrases: information systems, hypergraph, modelling, enterprise architecture, process algebra
2010 Mathematics Subject Classification: 68U35, 68M99, 05C65, 97M99, 68Q85
This work was partially supported by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).
We are very thankful to Prof. András Benczúr for raising the research issue and helpful discussions.

theory of hypergraphs was developed in the past decades; the generalizations of definitions as trees, cycles, and coloring for hypergraphs have been elaborated with the accompanying theorems.

The growing importance of computational science and informatics brought in composite systems of large size. Several areas of modeling and structuring of phenomena in cyberspace required more flexible representational approach than traditional graphs. The system of complex relationships has made necessary the further generalization of original definition of hypergraph to provide a powerful descriptive method. Thereby the hypergraphs have found new application areas and got a new impetus.

The reason is why hypergraphs seem apt to depict relations in information systems, social networks, document centered information processing, Web information systems, are the relationships among services within a service-oriented architecture. One of the basic problems is that the concepts that potentially may be mapped to vertices of graphs are heterogeneous. Consequently, a theoretically and mathematically correct representation of the structure to be dealt with cannot be achieved. For example, it seems to be a good idea to make use of a bipartite graph for describing a system that contains two distinctive set of things. Each single thing may be identified as a node of the graph, the graph may contain two disjoint sets of nodes. However, this approach at the same time loses homogeneity and heterogeneity feature of the system. The two separate disjoint sets represent two distinct groups of things that are definitely different but homogeneous within their own group. Although the operations that required to handle components of the system may need a uniform manipulation of elements in some cases in spite of their dissimilarity from some viewpoints.

The hypergraph and generalized hypergraph as concepts of discrete mathematics are appropriate for describing such systems. The basic differences at conceptual level between the traditional graph and hypergraph theory is that a specific edge tow nodes in a graph, however, in a hypergraph - the so called hyperedges - can group together more than two vertices. The *generalized hypergraph* yields the opportunity that a hyperedge can be perceived as a vertex and be a component of other hyperedges as a node. Consequently, the hypergraph is a generalization of several mathematical concepts as graphs, projective planes, affine planes [8].

Firstly, we will survey the application areas of hypergraphs in informatics and information systems, and then we will present our research in the context of Web information systems and their architecture. Finally, we will outline a research agenda in the domain of information systems for further application opportunities.

The structure of our paper is organized as follows. In Section 2, we outline applications of hypergraphs for social networks analysis. This is followed by Section 3 that describes hypergraph applications in a SOA and Cloud environment. We provide in Section 4 a description of our research on modeling and perceiving the architecture of information systems. Finally, conclusions and future work directions will be shown in the last section.

2. Social networks analysis

We have studied earlier the alternatives of software and system architectures that fit to the investigation of social networks [23]. There are studies on structure and relationships within social networks as e.g. Twitter [29]. Within the representation of On-line Social Networks (OSN) or Social Network Sites (SNS) as graphs, the vertices designate individuals or group of persons. The links among them represented as the edges of graphs and may describe friendship (with various granularity), cooperation, business relationships etc.

However, Twitter as social network has a fairly complex construction, namely it has concepts of relationship as following, reply to and mention. That leads to a more complex graph representation that requires directed, differentiated edges between vertices that can be represented e.g. by coloring. The interdependencies between the various messages should be taken into account if any semantically rich interpretation is required for the analysis, namely either the topic of texts or the relationship among the messages or posts should be shown in an appropriate representational way [2, 31].

Facebook basically and originally contained a friendship relation. On evolving of services at Facebook, the relationship of friendship has been refined as close friends, friends, and friends except acquaintances, public. These attributes are used in sharing micro-blogs and posts. The basic relation between people and groups became a more sophisticated system. There are solutions to represent such structure by graphs but hypergraphs offer a uniform treatment of the above mentioned phenomena.

One of the specificity and area of interests of Twitter is that the Twitter data are not only a set of data items and texts but through the contained data items a complex network can be built up. One of the structure is the *follower graph* (tweeters that follow specific other users), the other relationship that is fairly loose and ad hoc - is created by the mechanisms of re-tweet and hash-tag. The previously mentioned network structures can be mapped to graph structures but to discover the basic organizing principles of sub-graphs leads to challenging graph theoretic questions and empirical investigations. The various features of relationships between individual entities can be reflected better by hypergraph than traditional graph structure. The links displaying multi-faceted characteristics in the case of Twitter and Facebook as well can be mapped by hyperedges more faithfully than other graph theoretical approaches.

A research question can be formulated whether the relationship graphs of two deeply different SNSs as e.g. Twitter and Facebook provide some clues about the evolution of networks, features of users behavior that can be spotted by graph structures.

OSN frequently contains so called *supra-dyadic* relationships [6], i.e. the dyadic relationships appear between pairs of actors. On investigation of tweets within Twitter, the attributes of tweets that are worth analyzing not only the actors playing a role in actual tweets but e.g. geographical positions of actors through geo-tagging furthermore the applied language, date and time of tweets etc. In database worlds or entity-relationship modeling, we may call these types of relationship as ternary or quaternary relations among entities. When complex social networks are represented by hypergraphs the supra-dyadic relationships can be examined through the eigenvector centrality of hypergraph [6]. The vertices may represent individuals that produce tweets, the nodes linked by hyperedges manifesting follower, re-tweet, hash tag relationships extended by geographical, time and other necessary information.

3. Service-oriented architecture

Service-oriented architecture (SOA) turns into a popular software architecture as enterprise architecture that supports the business processes at conceptual, system analysis and logical design viewpoint. Service-oriented architecture provides a toolset to assist in application development and enterprise wide application integration. The essential characteristic of SOA is, that the computational resources are accessible in the form of services [15]. The past decade has brought success for this architectural approach at several companies because this solution is considered as cost-effective and one that takes care about reusability.

The most recent development on architecture area led to the concept of SaaS (Software as a Service) within a Cloud environment. The tenants of Cloud use various services in the form of SaaS applications that are built up from components out of SOA services. In a multi-tenant environment, the services are shared among several customers. The multi-tenant applications yield benefits both for customer and service provider since the multi-tenant solution decreases service delivery costs for service provider and reduces the subscription price for the customer or enterprise. Although, the multi tenant environment leads to issues to be solved. Namely, the *Quality of Service (QoS)* is a critical point how to assure a service level when multiple tenants share the same set of applications [30]. The performance issues should be handled as e.g. (virtual) servers housing specific services, may be overloaded by the tenants; the service level as e.g. response time, transaction output, and availability and as a consequence of saturation reliability for these services will be degraded. The service levels described in QoS should be mapped onto services consisting of applications for each single tenant. The load balancing of (virtual) servers is vital to keep the service levels at the pre-defined values in order to fulfill the QoS for each single tenant. It is supposed that public clouds has a practically infinite capacity and elasticity [32]. In the case of private and/or community cloud, the statement mentioned before is not necessarily valid therefore a model for load balancing and sharing of services is needed.

The services in a cloud environment can be grouped into three classes namely: business-independent, business-dependent and compound business services. The types of mutual dependencies between services from the cloud could be functional, business and platform. The dependency appears at logical and physical level that primarily relates to technology architecture layer that is typically hidden before tenants. The services can be mapped bijectively on vertices of hypergraphs whereby the vertices in the hypergraph can be placed into partitions of the hypergraph by their mutual dependencies, by their deployment on service centers, and the requirement of algorithms dedicated to load balancing [12]. Elastic Service Placement Problem (ESPP) applies a combinatorial auction problem that may be formulated in a hypergraph terminology as well [7]. Hypergraph partitioning is a very beneficial approach for load balancing purposes in the case if data about dependencies and connectivity are obtainable. The general opinion is that hypergraph partitioning provides a better and superior method than the traditional graph partitioning in the case of load balancing [5]. There are concurrent approaches in the literature [11]. The hypergraph model fits to concurrent, communicating processes e.g. services in a SOA for business processes. One approach is that the data or document objects can be designated as vertices and then the communication demands can be specified by hyperedges. The hypergraph can be represented by matrices by which the communication cost and volume very precisely can be rendered.

The hypergraph model and partitioning are well-suited to accurate description of communicating processes, their resource and capacity requirements, and the sub-optimal but efficient algorithms can be designed to solve practical problems.

4. Systems modelling

As the literature proves it, the graphs as a representational structure are apt to system modeling, analysis and specification, especially to information and software systems [10, 14]. One of the opportunities is the use of the combination of graph and algebraic means [16]. In spite of their expressive power and flexibility the challenge is to find an appropriate representation of systems being investigated. The representation should accurately describe the states of systems, i.e. system configurations, keeping up the isomorphism between the structure of systems and representations through the graphs. The isomorphic mapping between systems and graph representation has several rewards. The graph theoretic approach yields a formalism that provides uniformity in some sense and a *normal form* representation. The mathematically sound representation makes the reusability possible. Moreover, the exploitation of mathematical tools for system analysis and formal consistency checking are available. The methods that can be applied are as follows: graph transformations, matrix representations of graphs, furthermore computation of properties and feature based on mathematical theorems. These methods can assist to assess soundness of model represented by graphs. For example, the correctness of mapping can be checked by test algorithms for isomorphism. The validity of transliteration of systems into graph representation is critical for exploiting the available mathematical tool set to model both the dynamical and static facet of systems as e.g. operational semantics, re-configurations, controls, security and the model transformation in analysis, design and operation time.

The representation of systems' characteristics as e.g. states, configurations through algebraic methods and techniques can be transcribed into graph representations. The typical algebraic method can be one of the versions of process calculi. The system representation appears in algebraic terms. The states and system configurations formulated in process algebraic syntax can be mapped into graph representation through inductive procedures. For Web Information Systems, a combined method was developed that contained process algebraic descriptive tools and methods of conceptual modeling that are especially suitable for designing information systems operating on the Web. The *story algebra* provides a rich tool set that can capture the sides of processes and the use-cases [27]. The processes within information system are derivatives of business processes; the actual use of them is described by scenarios and scenes. The story algebra offers the opportunity to treat with the Business process model and workflow side and the end-user/ organization side of Information System Models [4]. Moreover, the story algebra makes possible to link the process and end-user perspectives with the documents that are transporters of data and information, the structure of documents has some correlation with the structure of organization. Conway's thesis claims the correspondence between the structure of development team and produced software architecture [19]. The correlation between the document structure linked to an Information System and the structure of the organization can be assumed.

5. Modeling document intensive web information systems

The most recent information systems can be regarded as Web Information Systems (WIS) as Web became ubiquitous and a *lingua franca* for man-machine dialogues through various protocols that are used in the context of end-users. The WIS decisively differs from a set of Web pages. The *differentia specifica* between collection of Web pages and WIS can be perceived by the following way; WIS supports business processes and typically strongly integrated to other information systems within organizations.

The *Service Oriented Computing (SOC), Cloud Computing* creates a new context in which the information systems modeling should be re-thought [21]. The other important aspect of changes is the emphasis on documents as medium for information exchange, the documents can appear in the forma of HTML pages, SOAP messages, semi-structured documents (XML), and unstructured documents [3, 13, 26].

The information systems modeling is a challenging task, for this reason several modeling frameworks came into existence. To understand in a holistic way and a top-down approach of information systems, the Enterprise Architecture frameworks offered a comprehensive view as Zachman and TOGAF method [18, 33]. Blokdijks *Four Models in Three Views* of Information Systems presents principles for analyzing and designing information systems. The information systems are inherently complex systems so that the modeling efforts should keep an eye on following principles: graphical modeling languages provide the best human interpretable description of applications; the model instances should be transformed into the refined model instances. The documents in the form of XML should join some of the model instances that represent significant business processes. The combination of Zachman Enterprise Architecture, the axiomatic design, document object model (DOM) and the use of UML as visual language form an approach that can be used for modeling WIS [22, 24, 25].

5.1. A formal hypergraph model

Web Information System (WIS)	We create an abstraction of WIS in the form of a generalized hypergaph that consists of vertices and hyperedges.
Node/vertex in a hypergaph	Each node (or vertex) corresponds to an element within a WIS, e.g. documents, elements of documents (constituting a tree structure), business processes, processes in scenarios and scenes, workflows, layers of workflows, web services, networks of web services, etc.
Edge in a hypergaph	Edge is a specific hyperedge with cardinality equal to two. Edge denotes binary relationships between two nodes, as e.g. free documents is consumed by a certain Web service, a generic document is the ancestor of an intensional documents, a free-document resulted in a ground-document after binding, valuating of variables etc.

Hyperedge	A hyperedge represents a relationship among a subset of nodes as e.g. Web services belonging to a specific workflow, Business Process containing workflows, nodes of a document tree structure composing a document etc.
System graph	A hypergaph that includes a disjoint node dedicated to the modeling environment of the system, plus all the nodes and hyperedges of the WIS.
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 documents, Web services and related entities out of data model etc. Subsystem may be created by operations that conforms with graph theoretical concepts as: Induced subhypergraph; Subhypergraph; Partial hypergraph;
Interconnecting sub-systems - hyperedges graph of the generalized hypergaph	A graph consisting of all the vertices in sub-systems and all hyperedges connecting subsystems along with the disjoint, environment node.

Table 1: Mapping the Concepts of Web Information Systems onto the Notion of Hypergaph.

Firstly, we should bring to mind the basic definitions of hypergraphs in order to apply for modeling of WIS.

Definition 5.1 (Hypergraph)**.** *A hypergraph H is a pair (V, E) of a finite set V = v1, .. ., vn and a set E of nonempty subsets of V. The elements of V are called vertices, the elements of E edges [8].*

Definition 5.2 (Generalized hypergraph)**.** *The notion of hypergraph may be extended in a way 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.

5.2. Hypergraphs and information system modeling

The proposal that we want to present is to use hypergraphs in the Web Information Systems modelling in various viewpoint and perspectives of Information Systems Architecture. We hope that the hypergraphs can provide a unified and uniform framework to handle the complexities and diversity of models that are employed during the analysis, design and operation of information systems. From theoretical viewpoints, the hypergraphs may provide better understanding of organization of information systems and structuring principle. From practical viewpoints, it gives clues for designing and operational principles and takes care of controlling and security mechanisms.

5.3. Web information system architecture

Generally, the Information System Architecture describes the structure of modules of information processing systems, the links among components, the design and analysis principles at information system level of which the main purpose is to buttress business processes.

The logical design of information systems in the sense of technology appears in the form of software architecture and their building blocks. The Zachman Framework [33] was one of the first attempts that stressed the fact that software architectures were not enough to capture the essential characteristics of Information Systems Architecture. While software architectures describe details of systems structure (using, for example, E-R and DFD diagrams, Gang of Four architecture pattern), Information System Architecture focuses on the high-level business and information system processes [17]. *Enterprise IT Architecture* can be understood as the collection of strategic and architectural principles that includes the Information, Business System, and Technical Architectures:

- Business (systems) architecture. The major constituents are the business processes, the related roles and actors, furthermore, the structure and information content and functional services of all business systems within the organization.
- Information (and Data) Architecture renders the data types, information

and containing documents that assist business processes; moreover the interrelationships among pieces of information.

- Technical Architecture contains the software and hardware technologies in the form of logical and physical platforms used in application implementation in eh form of information systems. Technical architecture sustains the integrity of the hardware, software, and infrastructure environment that is necessary to support the Business Systems Architecture and Information Systems Architecture.

Each single architecture layer can be refined from the viewpoints of stakeholders. The Zachman framework demonstrates a solution for a higher granularity of viewpoints and views that is able to express more accurately the complex relationships between the models of information systems.

Up to now, one of the best systematic structure of information models is given by Blokdijks work [4]. The main parts of Blokdijks structuring framework are: *organizational model* that represents hierarchy of business units and elements and the method of *work; information model* that contains information, structured, semi-structured, unstructured and other media format material, furthermore their origin and course of action for manipulation; *data model* that represents things of the physical world about which information is stored along with their links to each other , it lays the ground for the implementation model of data; *process model* that yields a description about the structure of business *activities* and the strongly coupled to the *control structure*.

The Zachman framework and the Blokdijks information systems model are in fact two orthogonal views of information systems. Thereby, a three dimensional structure came into existence. Some components of Blokdijks information systems model are used in specific viewpoints and perspectives of Zachman Enterprise architecture. We can arrange that complex structure in a cube. The interrelationships can be captured by the generalized hypergraph approach.

The Blokdijks model differentiates between data and information model that fact provides the opportunity to handle separately the documents as information resources. A document in itself is a tree structure that can be described by the document object model along with XML [22]. The structure of the document can be represented by hyperedges, but at the same time, the set of hyperedges manifests the document can be perceived as a vertex that belong to other hyperedges that demonstrate the use of the document in other parts of the model. For example, the document is used in business process within workflows, the document may play the role of input and output for use cases in

J. A. Zachman S. H. Spewak	Entities = what Data Architecture	Activities= how Applications Architecture	Locations= whereTechnology Architecture	People = who	Time= when	Motivation =why	
Planner Objectives / Scope (Contextual)	List of Business Objects Class = Class of Business Thing	List of Business Processes Process = Class of Business Process	List of Business Locations Node = Major Business Location	List of Organizations important to the Business People = Major Organizations	List of Events Significant to the Bus. Time = Major Bus. Events	List of Bus. Goals / Strategies Ends / Means = Maj. Bus. Goals / Crit, Suc. Factor	Scope
Owner Enterprise Model (Conceptual)	Semantic Model Object Class = Business Entity Association = Bus. Relationship	Business Process Model Proc. = Bus. Process, Web Services I/O = Bus. Resources, Documents	Business Logistics System Node = Busin. Location Link = Business Linkage	Work Flow Model People = Organization Unit Work = Work Product, Documents	Master Schedule Time = Bus. Event Cycle = Bus. Cycle	Business Plan Ends = Bus. Objective Means = Bus. Strategy	Enterprise Model
Designer Information Systems Model (Logical)	Logical Data Model Ent. = Data Entity Reln = Data Relationship	Application Architecture Proc. = Application Function, Web Services, Method of Object Class, I/O = User Views, Semi-structured documents	System Geographic Deployment Architecture e.g. Distributed System Arch. Node = I/S Service. (Processor, Storage, Logical Application Component. etc.) Link = Relationship between Logical Appl. Comp.	Human Interface Architecture People = Role Work = Deliverable, Semi-structured documents	Processing Structure Time = System Event, Orchestration Cycle = Processing Cycle	Business Rules Ends = Structural Assertion, Means = Action Assertion,	System Model
Builder Technology Model (Physical)	Physical Data Model Ent. = Segment / Table / etc Reln = Pointer/ Key / etc	System Design Proc. = I/S Services I/O = Data Elements / Sets, XML / HTML documents	System Architecture / Technology Architecture Physical Application Comp. Node = Hardware / Systems Software Link = Line Specifications	Presentation Architecture People = Screen Format, HTML / XML interface Work = User	Control Structure Time = Execute, Choreography Cycle = Component Cycle	Rule Design Ends = Condition Means = Action	Technical Model
Subcon-tractor Detailed Specifications (Out-of-context)	Data Definition Repository Ent. = Field Reln = Address	Programs Supporting Software Components Proc. = Language Statement I/O = Data Item, XML Field	Network Architecture Node = Address Link = Protocol	Security Architecture People = Identity, Authentication, Authorization, Work = Job	Timing Definition Time = Interrupt Cycle = Machine Cycle	Rule Specification Ends = Sub-condition Means = Step	Components
Functioning Enterprise	Data	Function	Network	Organization	Schedule	Strategy	

Table 2: A schematic mapping between Zachman's architecture and the WIS component

the sense of UML. The use cases can be formalized in scenarios and scenes for WIS. The scenarios contain processes that are formulated by process algebra. The hypergraph is able to display the before-mentioned complex relationship.

5.4. Hypergraphs and systems engineering

The hypergraph approach has the advantage of reducing the degree of complexity and variety of relationships between models thereby the WIS can adapt to the changing environment better, and enhance the stability of system.
The hypergraph approach can be applied to the information and data model. As we have discussed earlier, the hypergraph is capable to represent even the complex document structure. The document object model (DOM) can be rewritten in hypergraph structure and this way the document-centric feature of WIS can be represented by a better way, the structure of interrelationships among process, organization, core data structure and document elements can be expressed [9]. The evolution of documents and the related data can be tracked more easily both in design and operation time [24]. The clash between the inherent nature of documents and disciplined data structure (e.g. database) can be resolved through hypergraphs. The data structure, for example, in relational database format can not be divided into parts - i.e. partitions of databases - that correlates to the organization structure; nevertheless the document structure follow more or less the structure of organization and the roles within it. Utilizing the hypergraph approach, the dual nature and behaviour of documents can be handled, i.e. the structure of documents aligned with the organization and the data structure being a definite part of database and contained in documents can be reconciled into a unified framework.
The design artifacts of WIS can be perceived as the set of certain documents and the strongly coupled set of processes incorporated in the form of scenarios and scenes in the sense of story algebra approach [27].

The generalized hypergraph approach makes possible to build-up structures that are orthogonal to each other although the complex of structures can express the systems of relationships in a disciplined way. The information space consists of documents, data contained in documents, and the information bases manifesting in traditional structured and semi-structured databases.

The documents as input and output media and the elements of story algebra compose together the design artifacts of WIS. The basic concepts of story algebra can be represented in a hypergraph by the following way:

- *Story space*, is the information space of WIS. The elements of information space as e.g. documents and group of relevant data appear as nodes of hypergraph.

- *Story board*, can be perceived as directed, generalized hypergraph. Hyperarc (directed hyperedge) represents the evolutionary development between documents or document types. The documents or document types are represented by by document vertices that are indeed hyperedges. The hyperedges reflect the internal structure of the documents.

- *Scenario*, is a subdirhypergraph (directed subhypergraph) of the story board that describes the transformation of inputs in the form of documents and data through steps of processes.

- *Story*, is a directed *path* within the story board. The story board is a directed hypergraph (dirhypergraph). The starting point of the path is an instance of document type that contains a collection of free variables; the end-point is a document that contains all variables in fully evaluated form, moreover linked to the underlying databases.

- A *scene* can be represented as hyperedges and hyperarcs. Some of hyperarcs consist of documents and document types that can play the role of tail and head of the hyperarc and at the same time designate the changes of documents. Other incident hyperarcs contains as *head* a role (job responsibility) and/or actors of interested organization units so that the hyperarc designates the responsibility for manipulation on the specific document by roles and belonging actors; and as *tail* the role or actor who is responsible for receiving the document. From the process-oriented view, a *scene* is elementary part or task of information systems services, or a kind of Web services;

- The life cycle of documents can be described by a *directed path* through the dirhypergraph, the documents may be contained one or more *scenarios*. The documents without free variables appear as *head* of certain hyperarcs as the final state of documents manipulation.

	v_2	v_3	v_4	v_5	v_7	v_8	v_{10}	v_{11}
E_1	1	1	0	0	1	1	1	0
E_2	1	0	1	1	0	0	0	0
E_3	0	0	1	1	0	0	1	1

Table 3: The incidence matrix representing the hyper network

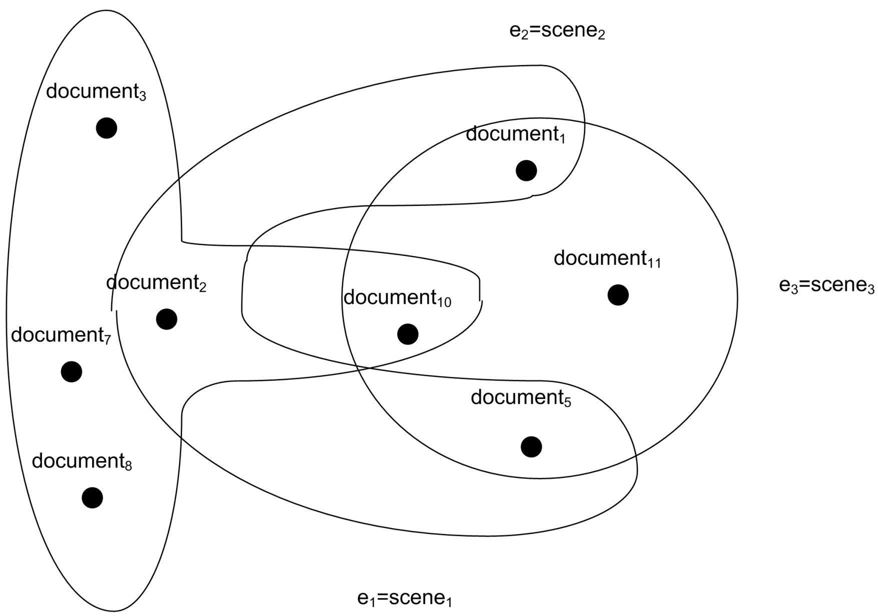

Figure 1: A part of a directed hypergraph for story board

5.5. Model transformations

To put the disparate and different models into a unified framework, the first attempt is to map the WIS and its possible models into the Zachman architecture that contains the important aspects and views of an IS within an organization context. To make the various models comparable, verifiable in the sense of correctness, the object-oriented modeling approach can be used for each modeling artifact as all the views of Zachman can be modeled by object-oriented paradigm from the classical IS and database technology to the document-oriented WIS.

Beside the representation of document and data, the description of application in the form of WIS could be done in an object-oriented manner. The UML provides a visual language, at the syntactic level, the systems analysis and design methods offer the systematic way how to organize a system into an operational application [20].

Axiomatic Design (AD) Theory [28] is a design method that offer assistance for designers to structure design problems by a systematic way. The axiomatic design approach can be applied to modeling and design of WIS. The

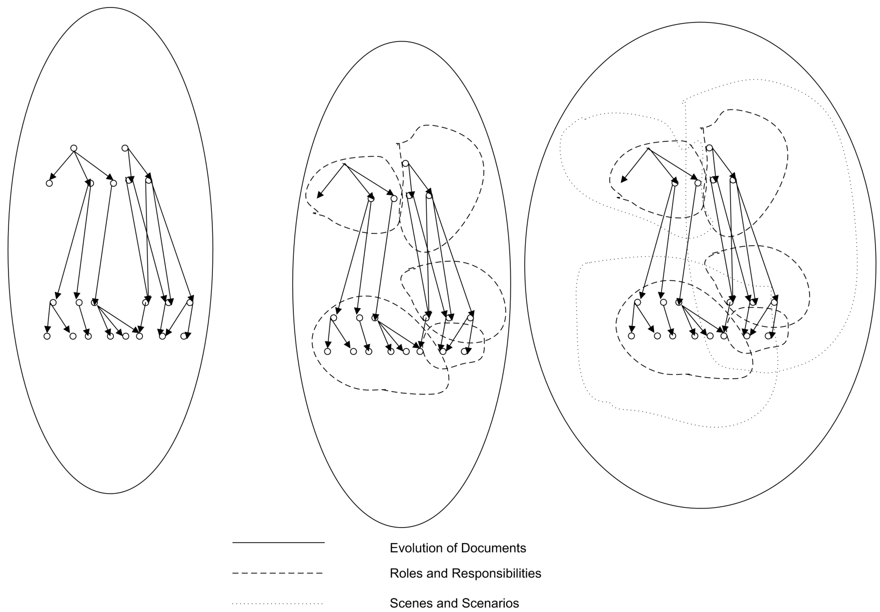

Figure 2: Partitions of the hypergraph according to document-centric story algebra

story algebra, the processes and documents as design artifacts appears at the surface of WIS. The processes captured through the scenes and scenarios of story algebra should be further refined to meet the underlying information and data bases. The consistency and integrity between the document-centric story algebra representation and the internal part of WIS that consists of the data structure, the behavior of entities and the impacts triggered by events.

The axiomatic design can be aligned with the Zachman architecture as perspectives can be considered as domains as all domains of axiomatic design theory is contained within the set of Zachmans perspectives and represents the life cycle of system development and operation.

The models that appear in the columns of Zachman architecture can be created using of the object-oriented paradigm. The framework matrix consists of a vertical axis that provides multiple perspectives of the overall architecture and a horizontal axis, which provides a viewpoint of the stakeholder. Each single model appears as a specific design artifact within the overall architecture. The perspectives and classifications within the framework are illustrated in Table 2.
According to axiomatic design theory, the requirements can be captured by the

following concepts: Customer Needs (CNs), Functional Requirements (FRs), Design Parameters (DPs), and Process Variables (PVs).

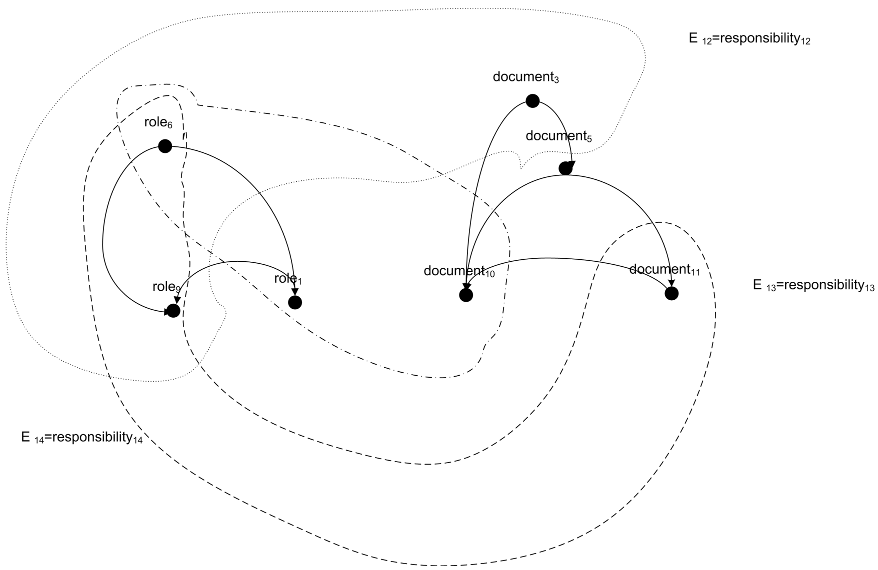

Figure 3: Hyperarcs expressing roles and responsibilities, documents evolution and interdependencies between them

Combining the hypergraph approach and axiomatic design methods yields a robust design tool. The hyperedges of the hypergraph can show us which DPs satisfy which FRs, which model refines which models, which models fulfill certain DPs or FRs.

The analysis and design of WIS consists of refinement and process of decomposition that proceed through the matrix both vertically and horizontally. The mapping process between the perspectives and viewpoints within the Zachman matrix can be expressed mathematically in terms of the characteristic vectors that define the design goals and design solutions, furthermore the relationships and their constraint. At a given level of design hierarchy, the set of functional requirements that define the specific design goals constitute a vector {FRs} in the functional aspect. Similarly, the set of design parameters in one of the perspectives for the FRs also constitutes a vector {DPs}. The relationship between these two vectors can be written as

$$\{FRs\} = A\{DPs\}. \tag{5.1}$$

The matrix A type represents the actual method, the transformation and mapping between the functional requirement and design decision (5.1). The verification and validation for properties of *correctness, faithfulness, consistency* and *integrity* among the design decision is represented by matrix type B (5.3). Matrix B designates the state transition between the relevant pairs of models situated in the columns, i.e. aspects of Zachman's architecture Table 4.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{5.2}$$

The User Requirements, the *Customer Needs (CNs)* transformed into Functional Requirements (FRs). The Functional Requirements can be represented by models that refers to classes of objects in Table 4.

$$\{DPs\} = B\{DPs\} \tag{5.3}$$

Matrix B representents the permanent refinement of the design that defines the characteristics of the process and is similar in form to A. The navigation rules for transformation can be deduced from the incidence and adjacency matrix of the hypergraph that represents WIS. The design hierarchies, the perspectives of Zachmans architecture are described in a unified and uniform manner by the hypergraph.

Aspects / Perspectives	Entities = what Data Architecture	Activities = how Applications Architecture	Locations = where Technology Architecture	People = who	Time = when	Motivation = why	
Contextual	CN	CN	CN	CN	CN	CN	Scope
Conceptual	FR	FR	FR	FR	FR	FR	Enterprise Model
Logical	DP	DP	DP	DP	DP	DP	System Model
Physical	DP	DP/	DP	DP	DP	DP	Technical Model
Detailed representation (out-of-context)	PV	PV	PV	PV	PV	PV	Components
Functioning enterprise/organization	Data	Function	Network	Organization	Schedule	Strategy	

Table 4: Model transformation based on Zachman architecture and CNs, FRs, DPs, PVs

6. Conclusion

The mathematical formalism has put the disparate approaches as Blokdikjs Information Systems Model, Zachman architecture/ontology and the Axiomatic Design into a unified framework. The hypergraph approach provides a tool for integrated management of that the orthogonal viewpoints of an information system as data and information, the procedural aspect representing the system behavior, and the models as design artifacts. The Axiomatic Design offers the systematic and mathematical formalization of requirements and their refinement during the system life cycle. This environment helps to change management of both design and operational time of WIS. The hypergraph accurately represents the interdependencies and set of relationships among the elements of WIS. These relationships are mathematically described by the adjacency and incidence matrices. On one side, the formal matrix descriptions can assist in checking the consistency, integrity, accuracy of WIS. However, the changes cannot be avoided either in design or operational time. The changes in the design time should be escalated through the models, the design and engineering activities. The propagation of modification will criss-cross the aspects and perspectives of Zachman matrix, between the various meta models for handling certain single models. The hypergraph approaches take care of consistency of models.

The most modern databases provides opportunities to operationalize our ideas. The formalized mathematical description can be implemented by graph databases. Moreover, several hypergraph databases emerged in the open source domain that offer services for implementing hypergraph structures directly in this database environment [1]. As further research, a prototype implementation of the formalized approach will be attempted.

References

[1] Hypergraphd. `http://www.kobrix.com/hgdb.jsp`, 2014.

[2] **Abraham, A., A.-E. Hassanien and V. Snasel,** *Computational Social Network Analysis: Trends, Tools and Research Advances,* Springer, 2009.

[3] **Bernauer, M. and M. Schrefl,** Self-maintaining web pages: from theory to practice, *Data & Knowledge Engineering,* **48** (1) (2004), 39-73.

[4] **Blokdijk, A. and P. Blokdijk,** *Planning and Design of INformation Systems,* Academic Press Professional Inc., 1987.

[5] **Boman, E., K. Devine, R. Heaphy, B. Hendrickson, V. Leung, L.A. Riesen, C. Vaughan, U. Catalyurek, D. Boydag, W. Mitchell et al.,** *Zoltan 3.0: Parallel parttioning, load balancing and data-management services. User guide,* Technical report, NM, Sandia National Laboratories, Albuquerque, 2007.

[6] **Bocacich, P., A.C. Holdren and M. Johnston,** Hyperedges and multidimensional centrality, *Social Networks,* **26** (3) (2004), 189-203.

[7] **Breitgand, D. and A. Epstein,** Sla-aware placement of multi-virtual machine elastic services in compute clouds, *Integrated Network Management (IM). 2011 IFIP/IEEE Int. Symp.,* IEEE, 2011, 161-168.

[8] **Bretto, A.,** Applications of hypergraph theory: A brief overview, *Hypergraph Theory,* Springer, 2013, 111-116.

[9] **McBrien, P. and A. Poulovassilis,** A semantic approach to integrating xml and structured data sources, *Advanced Information Systems Engineering,* Springer, 2001, 330-345.

[10] **Bruni, R., A. lluch-Lafuente, U. Montanari and E. Tuosto,** Style-based architectural reconfigurations, Bulletin of the European Association for Theoretical Computer Science, **94** (2008), 161-180.

[11] **Cambazoglu, B.B. and C. Aykanat,** Hypergraph-partitioning-based remapping models for image-space-parallel direct volume rendering of unstructured grids, *IEEE trans. on Parallel and Distributed Systems,* **18** (1) (2007), 3-16.

[12] **Catalyurek, U.V., E.G. Boman, K.D. Devine, D. Bozdag, R.T. heaphy and L.A. Riesen,** A repartitioning hypergraph model for dynamic load balancing, *J. of Parallel and Distributed Computing,* **69** (8) (2009), 711-724.

[13] **Chiu, C.-M. and M. Bieber,** A dynamically mapped open hypermedia system framework for integrating information systems, *Information and Software Technology,* **43** (2) (2001), 75-86.

[14] **Corradani, A., U. Montanari and F. Rossi,** An abstract machine for concurrent modular systems: Charm, *Theoretical Computer Science,* **122** (1) (1994), 165-200.

[15] **Erl, T.,** *Service-oriented architecture: Concepts, technology and design,* Pearson Education India, 2005.

[16] **Ferrari, C.L., D. Hirsch, I. Lanese, U. Montanari and E. Tuosto,** Synchronised hyperedge replacement as a model for service oriented computing, *Formal Methods for Components and Objects,* Springer, 2006, 22-43.

[17] **Gamma, E., R. Helm, R. Johnson and J. Vlissides,** *Design Patterns: Elements of Reusable Object-Oriented Software,* Pearson Education, 1994.

[18] **Open Group,** Togaf: The open group architecture framework, version 9, http://www.opengroup.org/togaf/, 2010.

[19] **Herbsleb, J.D. and R.E. Grinter,** Architectures, coordination and distance: Conway's law and beyond, *IEEE Software,* **16** (5) (1999), 63-70.

[20] **Larman, C.,** *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development,* Pearson Education India, 2012.

[21] **MacKenzie, C.M., K. Laskey, F. McCabe, P.F. Brown, R.M. and B.A. Hamilton,** Reference model for service oriented architecture 1.0, 2006.

[22] **Marini, J.,** *Document Object Model,* McGraw-Hill Inc., 2002.

[23] **Molnar, B. and Z. Vinczeller,** Comparative study of architecture for twitter analysis and a proposal for an improved approach, *IEEE 4th Int. Conf. on Cognitive Infocommunications (CogInfoCom),* IEEE, 2013, 11-16.

[24] **Molnár, B., A. Benczúr and Á. Tarcsi,** Formal approach to a web information system based on story algebra, *Singidunum J. of Applied Sciences,* **9** (2) (2012), 63-73.

[25] **Molnár, B. and Á. Tarcsi,** Design and architectural issues of contemporary web-based information systems, *Mediterranean J. of Computers and Networks,* **9** (2013), 20-28.

[26] **Chul-Ki Nam, Gil-Sang Jang and Jae-Hak J. Bae,** An xml-based active document for intelligent web applications, *Expert Systems with Applications,* **25** (2) (2003), 165-176.

[27] **Schewe, K.-D. and B. Thalheim,** Reasoning about web information systems using story algebras, *Advaces in Databases and Information Systems,* Springer, 2004, 54-66.

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

[29] **Szucs, I., G. Gombos and A. Kiss,** Five ws, one h and many tweets, *4th IEEE Int. Conf. on Cognitive Infocommunications (CogInfoCom),* IEEE, 2013, 441-446.

[30] **Dimitrakos, T., J. Martrat and S. Wesner,** *Service oriented infrastructures and cloud service platforms for the enterprise: A selection of common capabilities in real life business trials by the BEinGRID consotium,* Springer, 2009.

[31] **Wasserman, S.,** *Social Network Analysis: Methods and Applications,* **8**, Cambridge University Press, 1994.

[32] **Weinman, J.,** *Cloudonomics: The Business Value of Cloud Computing,* John Wiley & Sons, 2012.

[33] **Zachman, J.A.,** A framework for information systems architecture, *IBM Systems J.,* **26** (3) (1987), 276-292.

Bálint Molnár
Information Systems Department
Eötvös Loránd University
Pázmny Péter sétány 1/C
H-1117 Budapest, Hungary
molnarba@inf.elte.hu