Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 64 (2015) 369 – 378
Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2015 October 7-9, 2015

Document Centric Modeling of Information Systems

Bálint Molnár [a]*, András Benczúr [a]

[a] *Information Systems Department, Eötvös Loránd University of Budapest, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary*

Abstract

Most recently, the concept of business documents has started to play double role. On one side, a business document (word processing text or calculation sheet) can be used as specification tool, on the other side the business document is an immanent constituent of business processes, thereby essential component of business information systems. The recent tendency is that the majority of documents and their contents within business information systems remain in semi-structured format and a lesser part of documents is transformed into schemas of structured databases. The semi structured documents can be stored and processed in the modern database management systems, in compliance with the requirements of business processes. In order to keep in hand the emerging situation, we suggest the creation (1) a *theoretical framework* for modelling business information systems; (2) and a *design method* for practical application based on the theoretical model that provides the structuring principles. The modelling approach that focuses on *documents* and their interrelationships with business *processes* assists in perceiving the activities of modern information systems. The interrelationships between documents-centric modelling, the Enterprise Architecture and systematic approach for design provides an opportunity for a *unified modelling*.

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge

Keywords: Information System, Document Centric Process and Data Modeling, Process Algebra, Information System Architecture, Web services, Service-Oriented Architectures, Zachman Framework.

1. Introduction

The current Business Information Systems displays new behavioral properties, namely the documents, unstructured and semi-structured data have high relevance beside the structured data. One of the directions within

*Tel.: +36-1-372-2500/8042; fax: +36-1-372-2500/8044.
E-mail address: molnarba@inf.elte.hu

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge
doi:10.1016/j.procs.2015.08.501

management sciences is the service-orientation. The business processes of companies organized by the service-orientation pattern; consequently the structure of information systems' functions follows the model of IT services as either Web , REST or other appropriate technology that are based on the notion of services. These two trends slowly modify the requirements against the modeling methods for behavior of information systems[19,20]. The documents, interactive documents and the emphasis on Web interfaces led to the concept of modern Information Systems.

Beneath Information Systems there is a set of data that delivers the required information either to the business activities or to the information processes. During analysis, the question that is investigated is as to whether what information should be kept in the system. The data and their collections exist independently from business documents that may or may not related to decision processes. The modeling of Information Systems focused on document should adhere to established practices of data and information modeling. The model should be perceivable by users at high level. The approach for modeling and analysis should be semantically powerful in order to serve as a basic model and be understandable by users. The document-centric model is unlike to the data model but they are interdependent on each other. The document model tries to gather the transformations and the results of activities to extend information on documents with new facts.

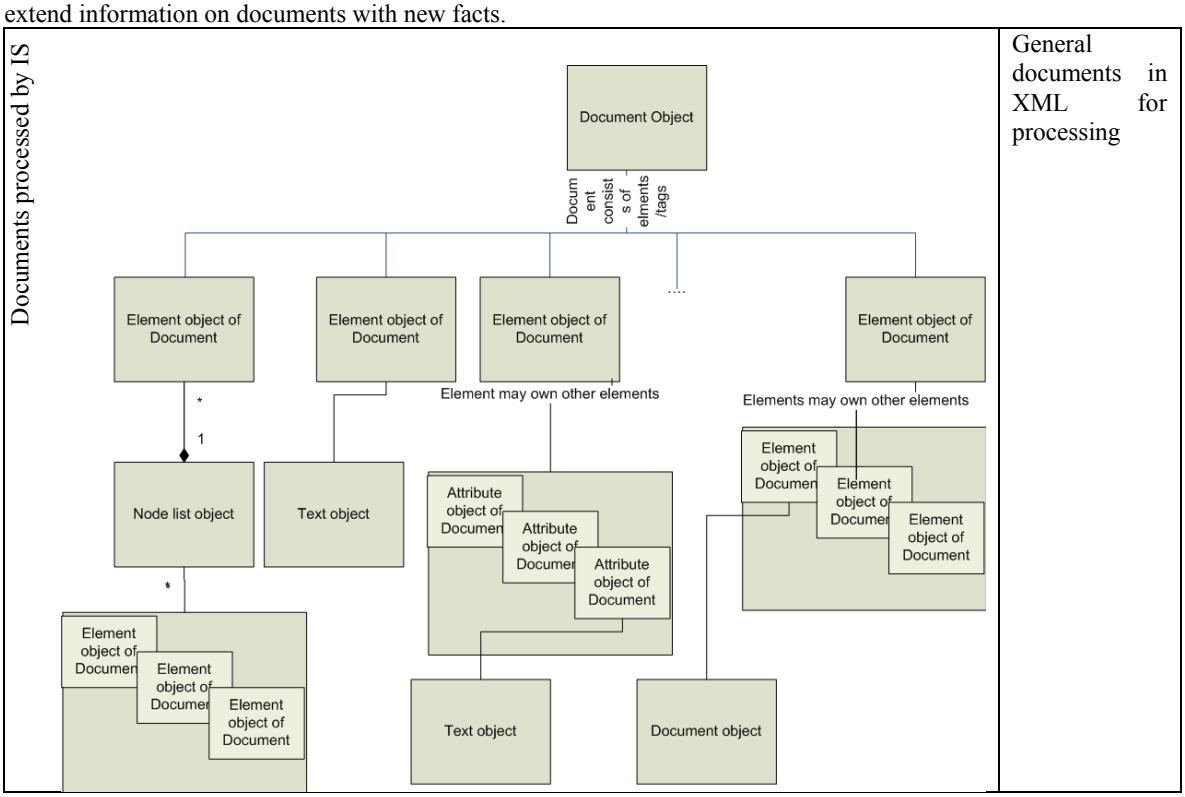

Figure 1: Levels of collections related to documents

The manipulation of documents happens through business processes; moreover the document model mirrors the structure of organization and events. Within the data model, those alterations should be tracked that change elements that are significant for business, i.e. create new ones, transform existing ones, set up new dependencies or adjust existing relationships. The actor or role that performs the data conversion should be distinguishable during transformation processes and the collection of data that are linked to documents and roles should be identifiable as well. In an e-government environment, a case study is planned and designed to verify and validate the results of proposed modeling approach with theoretical background.

In Section 2, we present the previous researches reported in the literature, in Section 3 we outline our method making use of the previous approaches in a document centric approach, and Section 4 provides a summary and conclusions.

2. Literature review

Joeris[16] proposed a document based approach for modeling control and data flow for business activities and data interchange among them. Wewers et al.[26] present a system that supports a framework for inter-organizational, document oriented workflow.

To help the perception of the complex behavior of IS (Information Systems) the enterprise architecture approaches offer support, namely the Zachman ontology and TOGAF, both was developed for information systems[23,24,30].

The artifact-centric business process model uses three basic concepts: artifact classes, tasks, and business rules[4,29]. The tasks handle the artifacts, the business rules govern which tasks should be triggered and which artifacts will be manipulated[14,15].

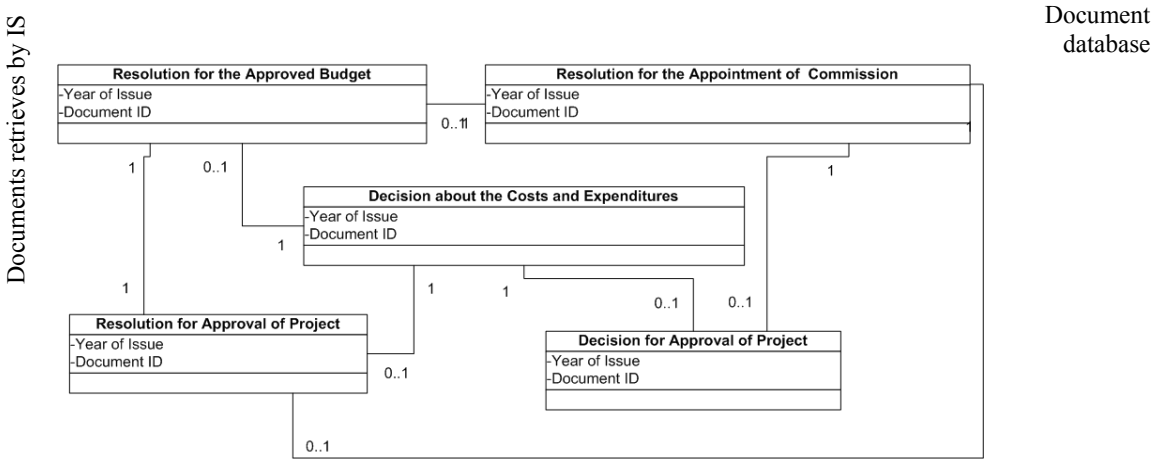

Figure 2: Class Diagram for Documents at "Document Conceptual Level" within a Case Study

SOA and the related technologies as XML, SOAP, WSDL and UDDI permits that services to be available through the Web[27]. The central notion of IS on Web is Web documents typically in XML format. There are analysis and design problem that should be improved. Emphasizing the problems with IT rather than business processes hinders the modeling and abstraction of stable and reliable Information Systems[1]. The approaches as *Service Oriented Computing* (SOC), and *Cloud Computing* concentrates on services as a standardized and general information exchange interface towards users. There are different input data format for interchange between services: (1) HTML pages, (2) SOAP messages (XML), (3) unstructured documents (XML)[3,9,22]. Unstructured documents may contain text, images, and other binary data, only the metadata may have formalized in XML using tags. Set of documents without uniform XLST, DTD or any other "style-sheets", we consider them as set of unstructured files since there is no general principle that can be enforced on each single document. Unstructured documents are the typical office documents without pre-defined style-sheets for tagging as the meta-data of documents may be tagged but the textual information is not. SOAP messages as XML tagged data can be regarded as structured but it may transport unstructured data.

XML documents can be considered as application-relevant "things", i.e. they can be perceived as new data objects to be stored and managed by a DBMS. This type of XML documents, in this sense, is document-centric, since their meaning depends on the document as a whole. The XML structure is more irregular in contrast to

structured data, and data contained in them are heterogeneous. Chidlovskii provides a formal grammatical description of XML[8,11].

The alignment and fitting between Business Processes and organization can be analyzed on the base of ontologies and semantic approaches[12,17]. The e-commerce, e-banking, e-tourism, Web-based Enterprise Resource Systems can be considered as typical IS on the WEB.

To combine the previously referred approaches to model modern Information Systems, there are various proposals[19,20,21]. Blokdijk's assembly of Information System Models[5] offers structuring principles, moreover the axiomatic design approach[25] can be employed for the Information Systems provides clues for both theoretic and practical modeling point of view. The concept of generalized *hypergraph*[6] seems to be a proper mathematical formalism that fits to unifying all viewpoints, perspectives, artifacts and modeling elements.

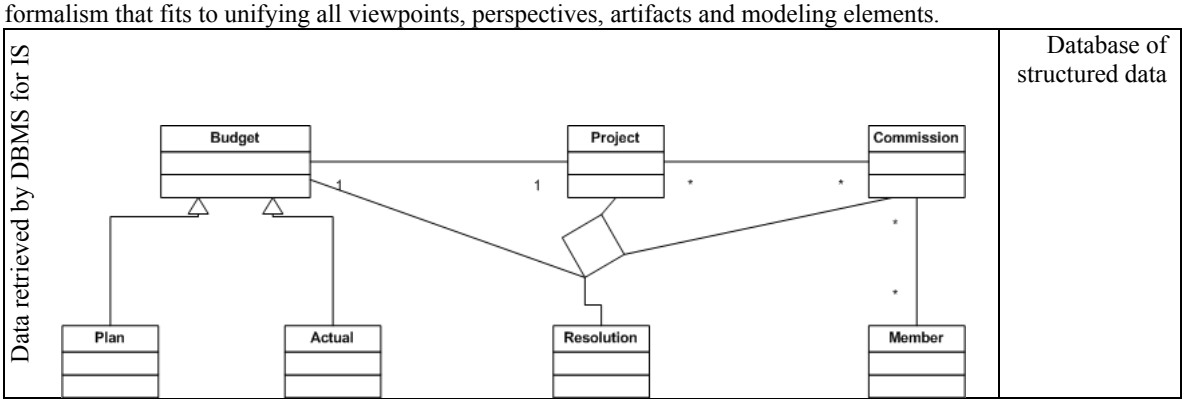

Figure 3: Class Diagram for Documents Class Diagram for Documents at "Data Model Conceptual Level" within a Case Study

3. Document centric approach

On modeling Information Systems, the data model plays a central role traditionally. To harmonize the traditional data model with document model, we generalize the concept of data models. In this representation, data models consist of **collections** (of data) so that each collection has a designation. The collections are *sets of data* or multi-set (bag) of data or *data types* with well-defined properties and structure; the most typical representation of *data model* is either relational data model or object-relational data model (However, there are several concurrent representation and implementation technology). The extension of data types as occurrences compose subsets of dataset that can be deduced from document structures. The *collections* includes identified data that are significant as their modifications over time are linked to documents (but that is not the same as the logging of database activities, in opposite it depicts the impact of activities related to document manipulation).

3.1. The Document-centric Modeling

The proposed approach is unlike to the traditional database modelling methods and the recent fashionable artefact-centred approaches. The document-centric modelling should exist with a strong *correspondence* to the Enterprise Architecture of the given organization, with a definite emphasis on the Business Processes. The structure of documents within an organization can be mapped onto the organigram and structure of business processes through homomorphism. The representation of both business processes and organization structures appear within Business Owner/Manager perspective of Enterprise Architecture[30]. The needs for flexible Information Systems lead to tendencies that can be formulated as a *Customer-centric* paradigm. The Customer-centric paradigm can be partly captured by a highly flexible document structure at the User Interface level. The documents should be adaptable to changes both in their structure and in their related content. (There is correlation between the software architecture and the project structure of the software development[13]). The document model should mirror the life cycle of documents, the manipulation, the events, and effects by business processes. The *modifications* that affect the data

included in *documents* should be traced, i.e. creating, modifying data items, establishing new relationship; the *precedence* analyses is an available option for tracking the impacts[5].

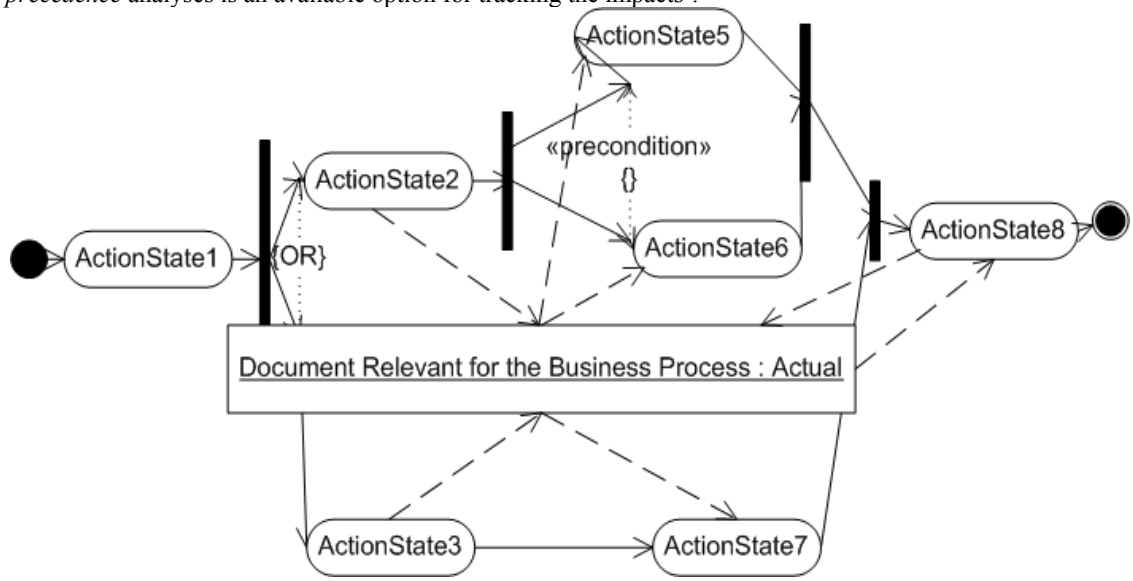

Figure 4: Business Process Model - A Case

The chain of events and processes can be monitored through *roles*/*actors* and their *handling* of identifiable data items within documents. The human roles stimulate data processing activities that effects documents, consequently the data items as well that are included in the documents. The *document-subdocument* structure (Figure 1, Figure 2, Figure 3) is able to represent both the organization and the information model at the same time. Whilst the data model is not structured as set of *sub-data models* instead it displays rather uniform configuration[5] (Figure 5). Blokdijk's model offers a structuring approach for perceiving Information Systems. The model's major components are: (1) **organizational model** (2) **information model**; (3) **data model**; (4) **process model.** The process model provides the composition of business *activities* and it is strongly coupled to the *control structure*. However, the data model is not an exact representation of the organization structure. For the reason that *patterns of data model* reflect the relevant facts about the organization but it does not map the *organization structures*. The document and data model requires a common representational method in which the services, and functions of documents, the coupled business activities can be depicted in a uniform manner. Furthermore, the interrelationships between the data and document model can be shown as much the same way as possible. There is logic of inheritance to identify data items within documents. Within the document chain, data elements are inherited from the previous element of documents. However, the responsibility for recognizing the proper data items relates to the currently valid *system role* (human or business process). The previously published figure[19] (Figure 5) has been extended to designate the *name space* of Document's DBMS, emphasize the mutual mapping between the information related to *control* and *business processes*, and to pinpoint to set of models that play a crucial role in integration of *enterprises* and *information systems*.

3.2. Types of Documents

The document model is composed of **document types**. The types of documents designate the state of their variables. We define the concept of ***binding*** by this way: a free field within a document; or a free variable is set for a value, i.e. valuated. The status and the type of documents can be inferred from the bindings, i.e. how many variables are already set to specific values. A **generic document** is a hierarchy of classes of documents. Finalizing or finishing a document instance within a hierarchy of a generic documents leads to that all free variables/fields are set

to a certain value step-by-step. The finalization of documents ensues from overarching business processes that can be linked to the flow of documents. The documents flow can be represented by data flow, Event Process Chain or Business Process Modeling Notation.

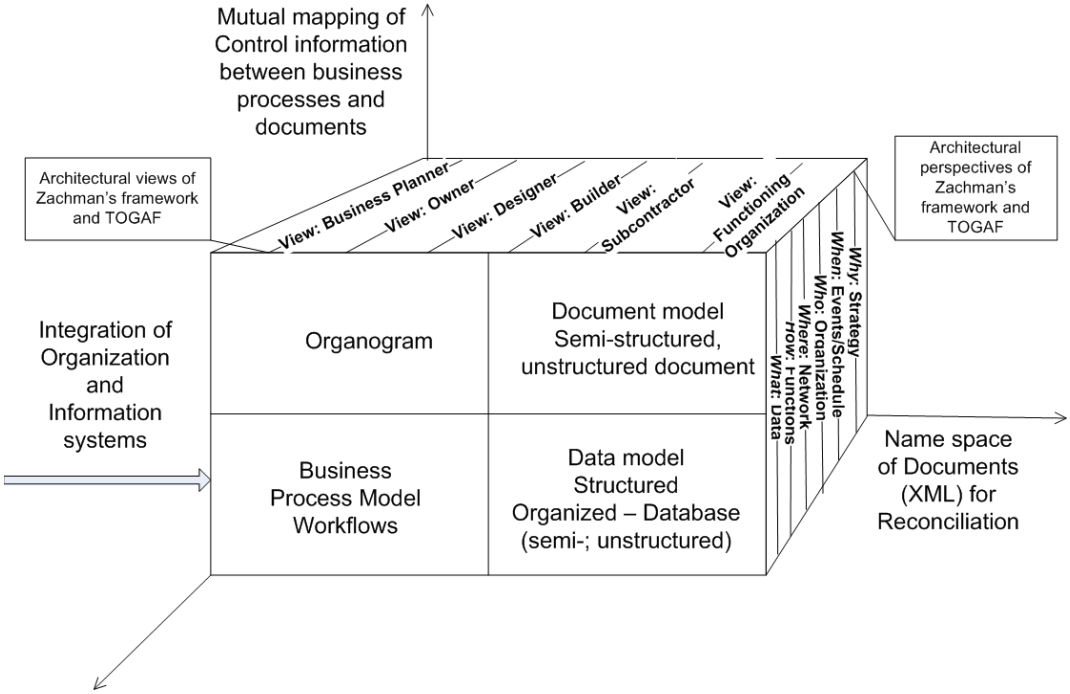

Figure 5: A Multi-dimensional model for interaction of Information Systems and Documents

The **free-documents** – like free tuples from tableau queries – can be perceived as documents that contain unbounded variables. As the document evolves more and more variables valuated, finally the documents achieve a state in that the documents cannot contain any unbounded variables. We can call documents in this state as ground-documents. The parts of documents can be regarded a finalized one from the viewpoint of one of the system roles; however the parts of the documents may still contain some free variables that require further processing by some other system roles. The external information is supplied by system roles out of the organization. i.e. outside of IS, the steps of the fulfillment process and their sequences are defined by business rules of the organization Valuation of a free variable needs external information what is supplied by system roles out of the organization, i.e. outside of IS, the steps of the fulfillment process and their sequences are defined by business rules of the organization.. For that reason, we make differences between the states of *finalized* and *ground-document*. A finalized document may contain free-variables and/or error signaling variables /fields that designate the necessity for further processing by some certain roles. The defect resolution of documents happens typically by organizational roles, i.e. outside of the automated information systems. In the case of algorithmic approach for error handling, the further document processing requires an intensional treatment, and usage of intensional documents, i.e. generate document instances based on business rules that are fulfilled by the automated IS to create extensional occurrences. A stable state of an instance of an overarching business process within an Information System can be achieved in the case if all documents that were involved in it are already ground-documents. The document handling finally results in ground-documents, ground sub-documents and assembled documents through several stages of development of to-be-finalized documents. The initial state is an uppermost documents and derived (intensional) documents. The intensional documents may contain free variables at meta-data and data level at the same time. On finishing their processing, the ground-documents build up a network. To establish interdependencies among ground-documents

may require some extra information. The supplemental information may assist to finish building-up the network of documents.

3.3. Representation of Documents

The current standards for describing the structure of documents are the XML, DOM (Document Object Model), JSON[10,18,22]. The conceptual data model is either represented by entity-relationship or object-oriented modeling methods. The interdependency between document model and data model can be represented by RDF[7].

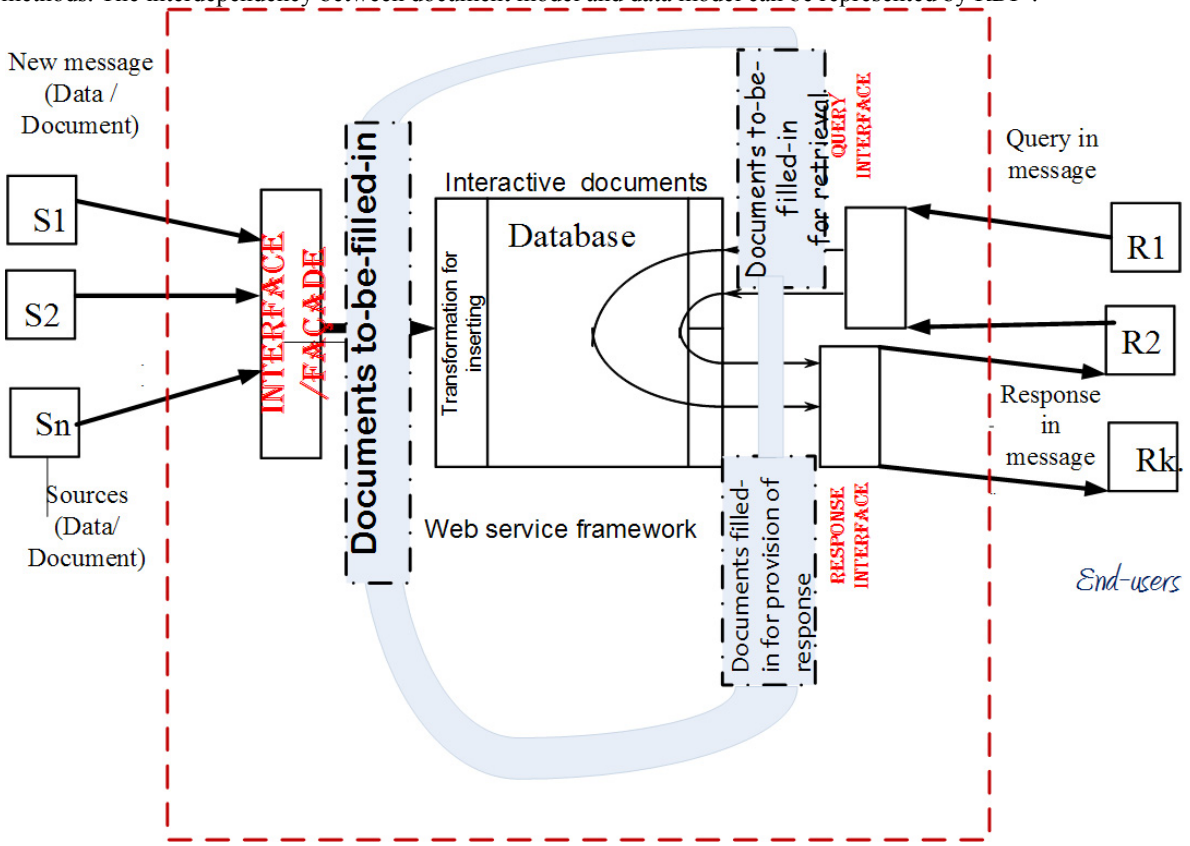

Figure 6: Information Systems' model in a document-centric approach.

To support enterprise architecture, the recent IT architectures (SOA, REST, etc.) offer procedures as orchestration and choreography to create complex services along with documents. The documents may belong to various categories as generic, intensional, to-be-finalized, and ground-document type. The architectures provide the opportunity to create protective, security and safety mechanisms[23,28].

The document handling finally results in ground-documents, ground sub-documents and assembled documents through several stages of development of to-be-finalized documents. The initial state is an uppermost documents and derived (intensional) documents. The intensional documents may contain free variables at meta-data and data level at the same time. On finishing their processing, the ground-documents build up a network. To establish interdependencies among ground-documents may require some extra information. The supplemental information may assist to finish building-up the network of documents.

3.4. The proposed document model

A database-centric Information System model that is based on an information theory approach[2] outlines a framework that describes the input, output and query processing. Figure 6 contains the Information System model indicated by the dashed red line; the previously published version[19] is enhanced to express that the source code data outside of the system and the code generated by the information system towards destination are communicated through a *crust consisting of documents*. In computerized systems, interactive documents and Web services become visible on the source and the output side. Free documents appear at the *interface/façade* level. The system roles (either human or automated system) perform variable valuation, or binding at each single variable through simple tasks. The business activities consist of tasks, a task can be composed of elementary tasks. An elementary task can be coupled to specific variables and its manipulation. The end-users who typically use information can retrieve data through documents, e.g. querying and fetching data from database and then processing the obtained responses.

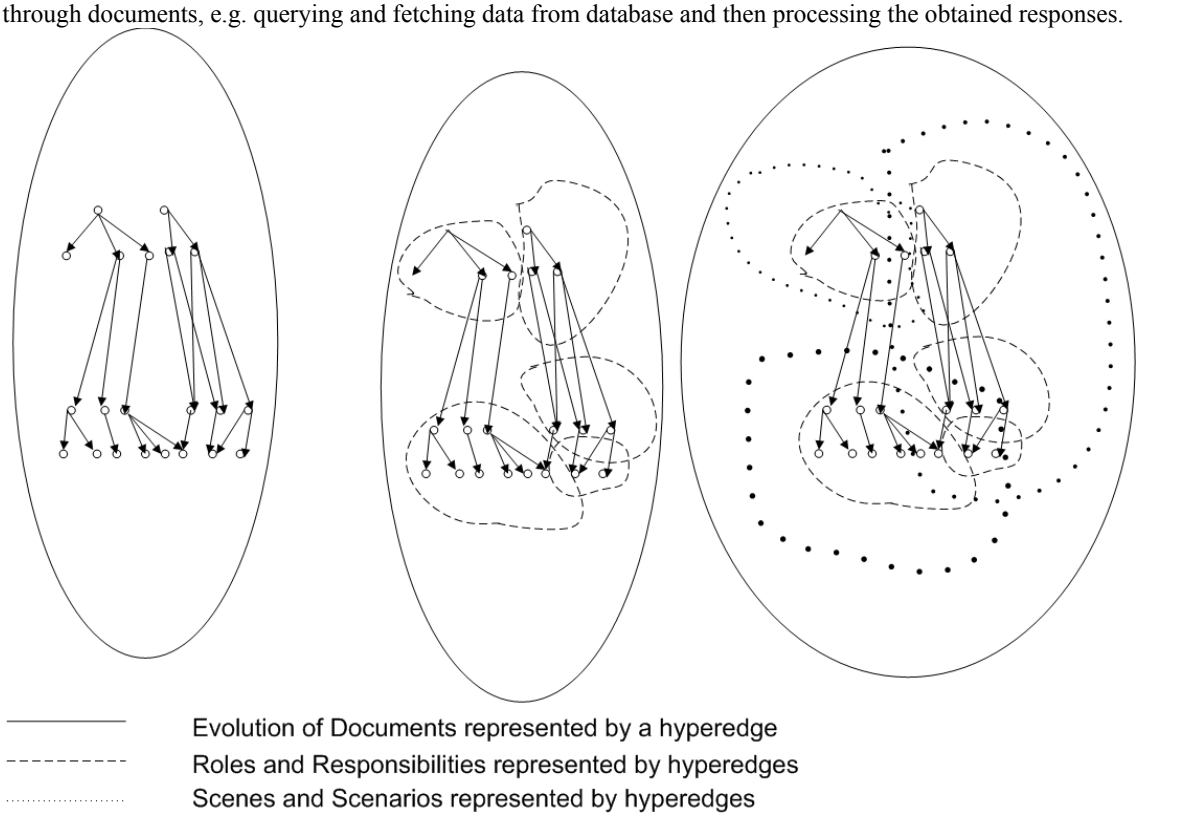

Figure 7: Representation the Life of Documents by Hypergraph

The two sides of the model, the input and potential output data are separated by the document model in the figure (Figure 6). Although the various possible states and instances of document types integrates both sides at the same time. The two sides show the same behavior but provide different services. The twofold behavior is actually either querying or alteration like.

In front of data model and its manifestation in the form of a database system, a new, document model should be placed in. Beside the logical formulation of data retrieval and modification, the model should contain the description for sequences of interaction among documents; moreover they should deal with collection of documents.

We can make difference between documents as being static or dynamic. The structure of a dynamic document may change as the response that is triggered by the system or system roles indicate it. The response may create instances of a general dynamic document that results in a sequence of free documents. The free documents are

gradually converted into ground-documents starting from generic ones through intensional ones to be finalized and ground-documents. The ground-documents do not include any free variables thereby the names of variables in the ground-documents can be placed into the name-space of database.

The system of documents – generic, intensional, free-documents, to-be-finalized and ground-documents – can be perceived as a meta-database. This meta-database encloses not only static structures, but it includes active component as well that can be realized by web services. The *active component* incorporates the potential *program code* for interactions among the system roles, documents etc. The active components encapsulate the codes for database management too. The above-mentioned techniques can be integrated into a unified framework (Figure 3, Figure 4.). Although, there is a lack for an comprehensive and not too complex scheme that combines all elements required for modeling Information Systems from a document-centric viewpoint and it is computable. Our proposal takes a step into the direction that both theoretic modeling and engineering viewpoint can be vindicated in a unified approach. The hypergraph as an appropriate mathematical tool may serve as a unifying approach to reconcile the before-mentioned heterogeneous model into a unified schema.

Table 1. Representation of Information Systems by Hypergraph.

Concept of Information System Theory	**Representation of concept in the domain of hypergraph theory**
Information System (IS)	A result of a system-development exercise that created a set of design artifacts. The set of elements and a *relationships* among them can be represented as nodes and edges within the graph. We can map the model elements to a *hypergraph* that consists of nodes and hyperedges.
Node/vertex in a hypergraph	Each node (or vertex) corresponds to an element within an IS, e.g. *documents*, elements of documents (constituting a tree structure), business processes, workflows, layers of workflows, web services, networks of web services, etc. The documents may represent one of the aspects for the information flow both inwards and outwards.
Edge in a hypergraph	*Edge* is a specific *hyperedge* with cardinality equal to two. Edge denotes binary relationships between two nodes, as e.g. free documents is processed 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, etc.
System graph	A hypergraph that includes a disjoint node for modeling the environment of the system, plus all the nodes and hyperedges of the 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.
Interconnecting sub-systems - hyperedges graph of the generalized hypergraph	A graph consisting of all the nodes in a sub-system and all hyperedges connecting together subsystems.

4. Conclusion

We have described issues and problems of modeling Information Systems. The recent evolution of technologies at user interface level and database handling raised questions that can be solved through new modeling approaches taking into account of ubiquitous documents as data holder.

Using of successful methods for single particular views, viewpoints and models, a framework for unifying the various approach is outlined. To provide a theoretically sound but reasonable complex and comprehensive approach for description and research of information systems a hypergraph based method is proposed (Table 1.). The direction of future research is to exploit the hypergraph as mathematical model to formalize the Information Systems' model from a document centric view.

References

1. Baghdadi, Y. (2005). A business model for deploying Web services: A data-centric approach based on factual dependencies. *Information Systems and e-Business Management*, *3*(2), 151-173.
2. Benczúr, A., 2003. The Evolution of Human Communication and the Information Revolution – A Mathematical Perspective, *Mathematical and Computer Modeling*, vol. 38, pp. 691-708.
3. Bernauer, M. , Schrefl, M. 2004. Self-maintaining web pages: from theory to practice, *Data & Knowledge Engineering* 48 , 39-73 2004.
4. Bhattacharya, K., Gerede, C., Hull, R., Liu, R., Su, J., 2007. Towards Formal Analysis of Artifact-Centric Business Process Models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007. LNCS, vol. 4714*, pp. 288–304. Springer, Heidelberg (2007)
5. Blokdijk, A., Blokdijk, P., 1987. *Planning and Design of Information Systems*, Academic Press, London, 1987.
6. Bretto, A., 2013. *Hypergraph Theory: An Introduction.* Springer.
7. Broekstra, J., Kampman, A., & Van Harmelen, F. 2002. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *The Semantic Web—ISWC 2002* (pp. 54-68). Springer Berlin Heidelberg.
8. Chidlovskii, B., 2002. Schema extraction from XML collections. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries* (pp. 291-292). ACM.
9. Chiua, C.-M., Bieber, M., 2001. A dynamically mapped open hypermedia system framework for integrating information systems, *Information and Software Technology* 43, 75-86, (2001).
10. Crockford, D. 2006. The application/json media type for javascript object notation (json). http://tools.ietf.org/html/rfc4627 (Accessed: 2014-01-04)
11. Daum, B., 2003. *Modeling business objects with XML schema*. Elsevier.
12. Gábor, A., Kő, A., Szabó, I., Ternai, K., & Varga, K. (2013, January). Compliance Check in Semantic Business Process Management. In *On the Move to Meaningful Internet Systems: OTM 2013 Workshops* (pp. 353-362). Springer Berlin Heidelberg.
13. Herbsleb, J. D. and Grinter. R., E. 1999. Architectures, Coordination, and Distance: Conway's Law and Beyond. *IEEE Softw.* 16 DOI=10.1109/52.795103, http://dx.doi.org/10.1109/52.795103
14. Hull, R., 2008: Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges, In: R., Meersman, Z., Tari, *On the Move to Meaningful Internet Systems: OTM 2008*, Springer Berlin / Heidelberg, 2008.
15. Hull, Richard., 2013. "Data-Centricity and Services Interoperation." In: B., Samik, P., Cesare, Z., Liang, F., Xiang, *Service-Oriented Computing* (2013): 1-8.
16. Joeris, G. , 1997. Cooperative and integrated workflow and document management for engineering applications, In: *Proceedings of Eighth International Workshop on Database and Expert Systems Application* (1997) 68–73.
17. Kő, A., & Ternai, K.: A Development Method for Ontology Based Business Processes. In *eChallenges e-2011* Conference Proceedings. IIMC International Information Management Corporation Ltd., Florence. (2011)
18. Marini, J. 2002. *Document Object Model: Processing Structured Documents*. Osborne/McGray-Hill, 2002,. ISBN 0-07-222436-3.
19. Molnár, B., & Benczúr, A. 2013. Facet of Modeling Web Information Systems from a Document-Centric View. *International Journal of Web Portals (IJWP)*, *5*(4), 57-70.
20. Molnár, B., Benczúr; A., Tarcsi, Á., 2012. Formal Approach to a Web Information System Based on Story Algebra, *Singidunum Journal of Applied Sciences: Economy Management Tourism Information Technology And Law* (ISSN: 2217-8090) 9: (2) pp. 63-73. (2012).
21. Molnár, B., Máriás, Z., Suhajda, Z., & Fekete, I., 2014. Amnis-Design and Implementation of an Adaptive Workflow Management System, *9th International Symposium on Applied Informatics and Related Areas - AIS2014*, Székesfehérvár: Óbudai Egyetem, 2014.
22. Nama, C.-K., Jang, G.-S., Ba, J.-H., 2003. An XML-based active document for intelligent web applications, *Expert Systems with Applications,* 25, 165-176, (2003).
23. OASIS 2006. *A reference model for service-oriented architecture*, White Paper, Service-Oriented Architecture Reference Model Technical Committee, Organization for the Advancement of Structured Information Standards, Billerica, MA, February, 2006.
24. Open Group, 2010. TOGAF: *The Open Group Architecture Framework, TOGAF® Version 9,* http://www.opengroup.org/togaf/ , 2010.
25. Suh, N.P.. 2001. *Axiomatic Design: Advantages and Applications*. Oxford University Press, New York, 2001.
26. Wewers, T., Wargitsch, C., 1998. Four dimensions of interorganizational, document-oriented workflow: a case study of the approval of hazardous-waste disposal, In: *The 3lst Hawaii International Conference*, vol. 4, 1998, pp. 332–341.
27. W3C 2001. *Web Services Description Language (WSDL)* 1.1. Web Site (2001). URL http://www.w3.org/TR/wsdl
28. Webber, J., Parastatidis, S., & Robinson, I. 2010. *REST in Practice: Hypermedia and Systems*
29. Yongchareon, S., Liu, C., 2010. A Process View Framework for Artifact-Centric Business Processes. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) *OTM 2010. LNCS, vol. 6426*, pp. 26–43. Springer, Heidelberg (2010).
30. Zachman, J.A. , 1987. A Framework for Information Systems Architecture, *IBM Systems Journal Volume*, 26, No. 3, pp. 276--292, 1987.