

# Modeling information systems from the viewpoint of active documents

## Proposal for a modeling approach putting emphasis on the ubiquitous documents

Bálint Molnár<sup>1</sup>  · András Benczúr<sup>1</sup>

Received: 14 July 2014 / Accepted: 30 June 2015 / Published online: 29 July 2015  
© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** The development of document handling by organizations at the level of business processes and business information systems leads to the phenomenon that the majority of documents and their contents remain in semi-structured format and definite minorities of documents are directly mapped onto structured databases. The rapidly evolving technology on the database field provides the opportunity to manage directly the semi-structured documents in line with the requirements of business processes. A continuum of possible document formats may exist in business environments. The documents can be categorized by the organization of the underlying data collections and according to the necessity and capability of data included in documents as whether wholly or partially is to be structured. The most modern database technology yields tools for handling and retrieving data making use of semi-structured and unstructured data. Our proposed approach (1) on one hand provides a *theoretical framework* for modeling IS to shape into some structure, (2) on the other hand yields guidance for *design method* to employ it for practical application with the extension of specific elementary models. The proposed modeling method that places the emphasis onto the *documents* and their symbiotic life with *processes* helps in understanding the behavior of most modern information systems. As a result, the combination of document-centric modeling and the enterprise architecture approach gives an opportunity for a *unified modeling* approach that keeps an eye on Conway's thesis that

states that software architecture congruent with the structure of development team, and this statement may be paraphrased that the overall document structure may reflect the structure of the specific organization.

**Keywords** Information · System · Web information system · Document-centric process and data modeling · Process algebra · Information system architecture web services · Service-oriented architectures · Zachman framework

## 1 Introduction

The most recent business information systems presents two novel properties, on one side the documents, unstructured and semi-structured data play central and important roles, on the other side the information systems (IS) services formulated as *Web services*. These two trends slowly modify the requirements against the modeling methods for behavior of IS [20,21]. The documents, interactive documents, and the emphasis on Web interfaces led to the concept of web information system (WIS).

Underlying the IS, there is a collection of data that delivers the necessary information to the operational systems either business processes or IT processes. The design issue is what data should be stored and what the reason is for they are kept in the system. The *data* and *their collections* exist separately from decision-related documents and they have some role in the operation of organization. The document-centric modeling of IS has to follow the patterns of traditional data and information modeling. The model should be understandable by end-users at conceptual level. The modeling framework should be semantically rich enough to work as a referential model and be semantically interpretable by

✉ Bálint Molnár  
molnarba@inf.elte.hu; molnarba@gmail.com

András Benczúr  
abenczur@inf.elte.hu

<sup>1</sup> Information Systems Department, Eötvös Loránd University of Budapest, Budapest, Pázmány Péter sétány 1/C, 1117, Hungary

end-users, i.e., the complexity of the model should be reasonable.

The document model differs from the data model but they are interdependent on each other. The document model attempts to capture the transformations and the extending acts with new facts of documents by business processes; moreover, it tries to reflect the structure of organization and events. Taking into account the data model, those changes should be documented that modify identified business elements, i.e., create new ones, modify the existing ones, establish new dependencies, or alter the existing relationships. The actor or role that carries out the modifications should be identifiable during the course of action and the collection of data that are subject to manipulation should be identifiable as well.

In an e-government environment, a case study is planned and designed to verify and validate the results of the proposed modeling approach with theoretical background.

The contrast between a collection of Web pages, typical Web Applications, and a WISs can be described by the following way: A WIS serves business processes (business process modeling, BPM) and is usually tightly coupled to other IS. WIS can also be perceived as database for structured, semi-structured as well as unstructured documents (XML-based). The alignment and fitting between Business Processes and organization can be analyzed on the basis of ontologies and semantic approaches [12, 16]. The e-commerce, e-banking, e-tourism, Web-based Enterprise Resource Systems can be considered as typical WIS. Nevertheless, the most recent IS and WIS cannot be differentiated as the IS apply the Web technologies intensively and generally. For this reason, we will use the concept of IS afterwards.

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

## 2 Literature review

The use of semi-structured and active documents described in the form of XML and a methodical design approach to construct web-based applications are discussed in [17].

Another article [2] presents a design methodology for a well-organized design process for a Web site. For large-scale WIS design, Rossi presents a method [26]. To assist in understanding the complex behavior of WIS, the enterprise architecture approaches provide assistance, namely the Zachman ontology and TOGAF, both were developed for information systems [24, 25, 31]. SOA as a reference architecture can assist to organize the utilization of software technology within a given enterprise, or consortium of organizations that should take part in information exchange to communicate with each other. In this sense, SOA can be

regarded as a set of principles to guide the design of software architecture that has, as a focus point, the concept of “it service” or “Web service” [29].

The emerging paradigms of service-oriented computing and *Cloud Computing* put emphasis on services as a uniform and general information exchange interface towards end-users. There are various input data formats for communication to services: (1) HTML pages, (2) SOAP messages, and (3) unstructured documents (XML) [5, 9, 23].

There were several attempts to cast in an appropriate mold the previously outlined approaches, issues, and solutions [20, 21]. The Enterprise Architecture frameworks such as Zachman or TOGAF provide a supporting environment [24, 25, 31]. Blokdijs’s assembly of IS Models [6] offers structuring principles; moreover, the axiomatic design approach [28] employed for the IS environment offers guidelines not only for theoretic modeling point of view but yields a support for practical design methods.

## 3 Document-centric approach

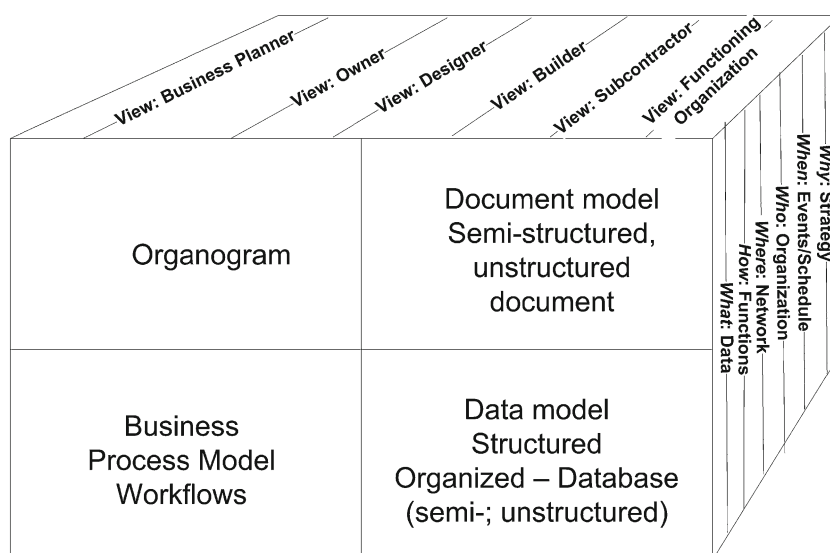
For modeling IS, we generalize the concept of data models. Data models consist of *collections* (of data) so that each collection has gotten a name. The collections are *set of data* or multi-set (bag) of data of *data types* with well-defined properties and structure; the most typical representation of *data model* is either relational data model or object-relational data model. The instances of data types make up finite subsets of potential dataset.

The *collections* contain identified data elements that play important role as their changes over time should be traced back to documents, but that is not identical with the logging of database activities, instead it describes the actions related to document manipulation.

### 3.1 The document-centric modeling

The proposed new approach *differs* from the *traditional database modeling methods*. The document-centric modeling should have a strong *correlation* with the enterprise architecture, more exactly to the Business Processes. The structure of documents within an organization is—probably—*congruent* with organigram and Business Owner/Manager perspective of Enterprise Architecture. The claim as a thesis needs empirical verification and clarification, but there is some indication that there is a mutual mapping between the software architecture and the project structure of the software development team in a software development and engineering environment [14]. The document model should reflect the evolution of documents, the changes, the events, the *stimulus–response* patterns in line with the business processes. The *changes* that affect the identifiable data contained in *documents* should be tracked, i.e., creating, modifying new data elements, setting

**Fig. 1** Document model of information systems from an enterprise architecture perspective



up new relationship, e.g., making use of precedence analyses [6]. The chain of course can be monitored through the *actors* and their *manipulation* of identifiable data elements within documents. The effects on documents and data elements exercised either by physical transformation within organization or by data processing activities triggered by human.

The *document–subdocument* structure is capable to describe both the organization and the information model at the same time while the data model is not structured as *sub-data models* [6] (Fig. 1). The main components of Blokdijs’ models are as follows: (1) *organizational model* that represents the line of business and the way a *work*, i.e., a task or a process is to be completed in the firm; (2) *information model* which describes information, textual, and other media format material, their source, and procedure of deriving; (3) *data model* that exhibits objects of the physical world—about which information is stored along with their links to each other—and provides the foundation for the implementation model of data; (4) a *process model* that gives a picture about the configuration of business world *activities* and the strongly tied to the *control structure*. However, the data model is not able to map the organization structure closely or fit to its organogram as *patterns of data model* profoundly differ from *organization structures*. Both models—document and data model—need a common descriptive approach in which the services and function of documents and the related business activities requiring documents can be formulated; furthermore, the interdependent relationship structure between the data and document model can be conceptualized as uniform as possible.

Processing of documents by human actors implies filling-in the free fields of interactive documents with data elements that can be identified within the document model. There is

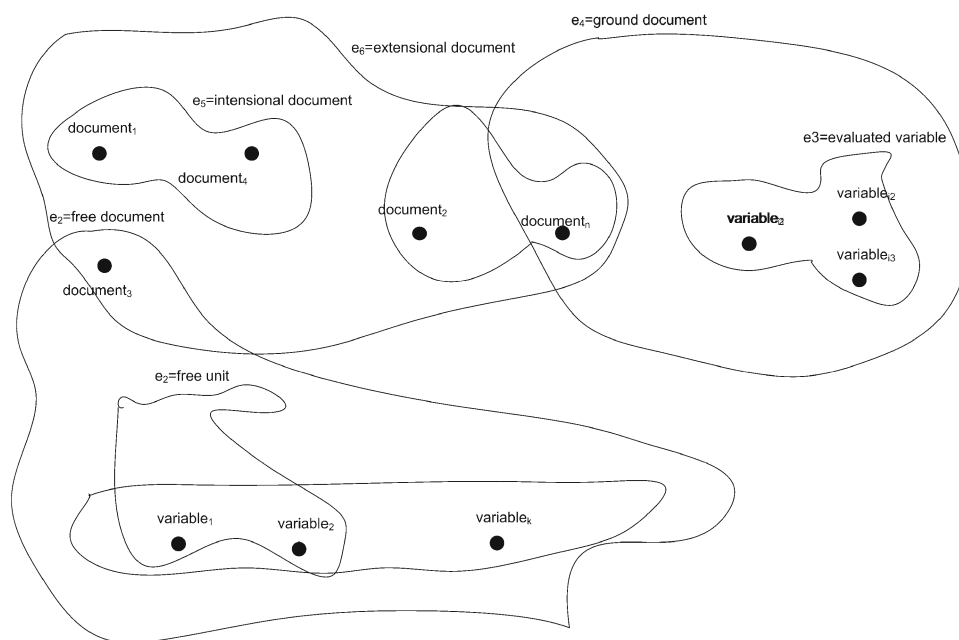
an inheritance mechanism regarding the identification of data elements within documents, i.e., the identified data elements are inherited from the previous documents within the document chain; however, the responsibility of identification belongs to the actual *system role* (human or business process) only in the case of the newest data elements.

A *generic document* itself is a hierarchy of documents; finalizing or finishing a document instance of a given hierarchy of documents leads to that all free variables/fields are set to certain value (Fig. 2). The finalization of documents at a certain point of time by overarching business processes can be described by the flow of documents and the documents flow can be represented by data flow, Event Process Chain, or BPM Notation.

### 3.2 The proposed document model

A database-centric IS model that follows an information theory approach [4] outlines a schema that depicts the input, output, and retrieval processing in a theoretic framework. Figure 3 contains the before-mentioned model denoted by the continuous black line in the figure. Within automated systems, interactive documents and Web services appear on the source side. *Free documents* with free variables, i.e., the to-be-filled-in documents occur at the interface/façade level. The system roles (either human or automated system) carry out variable valuation, or binding at each single variable through an elementary task. The business processes consist of tasks; a task can be decomposed to elementary tasks. An elementary task can be joined to a specific variable and its valuation or binding. The end-users who typically consume information can access data also through documents, however for querying and fetching data from database and then

**Fig. 2** An illustrative example for structure of documents via hypergraph representation



processing the obtained responses. Both sides of model, the input and potential output data are separated by the document model in the figure (Fig. 3); however, at the same time, the various possible states and instances of document types integrate both sides, essentially, present the same behavior but actually different services. The twofold behavior is actually either retrieval or modification like.

In front of the data model and its manifestation in the form of database system, a document model should be placed. Besides 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 and/or the template of dynamic document changes as the response (by system or system roles) triggers and indicates it. A sequence of *free documents* is generated. The documents step-by-step become ground documents starting from generic ones through intensional ones to finalized and ground documents. The *ground documents* do not contain any free variables, thereby the ground documents can be moved into the namespace of database.

To support document-centric modeling, we should apply a taxonomy of documents that assists to describe the relationship between the structure of documents and business processes. The document types are able to capture the characteristic of documents within IS. This definition contains an intrinsic and natural way of recursivity (Fig. 2).

1. For a specific organization, we can assume that there exists an *overarching document* that includes all theoretically and practically possible documents.
2. The *intensional document type* is a member of the bag of all possible documents that are able to instantiate *extensional document types* and/or *extensional documents* as well. The instantiation happens through some logical inference steps that contain rules aligned with the business processes and rules. The rule set incorporated in an intensional document operates on a collection of data, the existing documents, and messages coming from Web services.
3. Basically, the documents after instantiation contain “free variables”. The variables can belong to various types e.g., primitive, enumerative as basic types and tuples, sets, bags as composite types. Through *valuation* or *binding* of variables, a generic *document type hierarchy* can be brought into existence. The elements of a *document type hierarchy* by further valuation or variable binding are instantiated that will produce a document hierarchy consisting of *extensional documents*.
4. The *extensional documents* can be perceived as place holder for data collections that are under permanent manipulation. The free variables of extensional documents valuated or bounded step-by-step during business service activities; tasks of business processes fulfill specific variables depending on the context, i.e., organization unit, role, actor. We can differentiate between the concepts of ground documents and finalized documents. The variables of *finalized documents* can be partly fulfilled by definite parts of organization but it may contain some

free variables to be modified by other entities within the organization. All variables in a *ground document* have been valuated or bounded. *Ground documents* contain the facts relevant for the organization and for the underlying databases as they serve as sources for further processing.

The document model consists of *document types*. The major categories of documents reflect the state of variables. We understand under the concept of *binding* that a free field, free variable, is set for a value, i.e., valuated. The status of documents and consequently being in a particular category can be formulated in the sense of bindings, i.e., how many variables are already bounded to specific values. The free documents—like free tuples from tableau queries—can be perceived as documents that contain unbounded variables. As the document processing goes forward, more and more variables become filled-in; finally, the documents achieve a state in that the documents cannot contain any unbounded variables and we can call this state as a ground document. The documents can be considered from one of the system roles as a finalized one; however, the document may still contain some free variables that require further processing by some other system roles.

A stable state of an IS can be achieved in the case if all documents are already ground documents.

### 3.3 Representation of documents

The most recent standards for describing the structure of documents are XML, Document Object Model, and JSON [10, 18, 23]. The *conceptual data model* either follows the principles of entity–relationship or object-oriented class modeling approaches. The interdependency between document model and data model can be represented by RDF [8].

The most modern software architectures for supporting enterprise architecture—(SOA, REST, etc.)—delivers methods as orchestration and choreography to build up complex documents along with services, with various types such generic, intensional, to-be-finalized, ground document, and protective, security, and safety mechanisms, moreover with other automated processing [1, 24, 30].

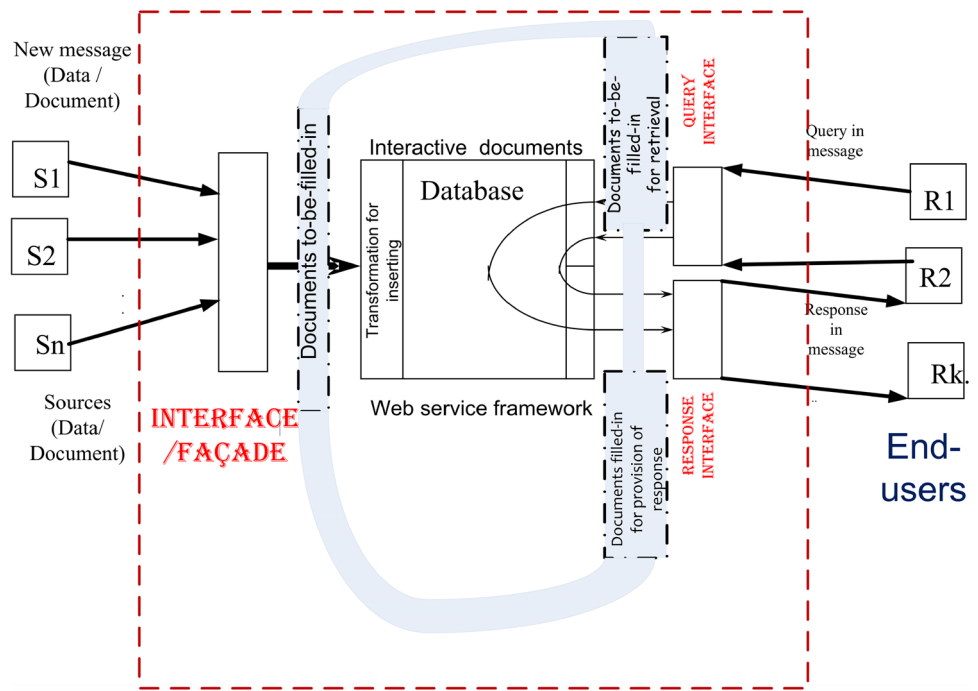
The document processing finally leads to ground documents, ground sub-documents, and assembled documents through several phases of to-be-finalized documents. The starting points are *uppermost* documents and some derived (intensional) documents that can be deduced from the ground sub-documents. The *intensional documents* may contain free variables at the meta-data and data levels at the same time. The ground documents after concluding their processing build up a network. The construction of interrelationships of ground documents may require some extra information for the reason that the structure of network could be finished.

In an *information theoretic framework*, a schema can be outlined for a compact representation. The input, output, and retrieval processing in a database-centric environment can be perceived as it follows: the inputting as a receiver, a kind of coding sub-systems that transforms data out of user interface that may appear in the form of an interactive document. In the case of WIS, the sides for source data and retrieved data for end-users may be manifested in Web pages and the coupled functions as Web services. The *coding sub-systems* receive pre-specified, fixed format messages in a specific language, in line with some standards. For the end-users, an intelligent user interface is provided, that is interactive for both directions, i.e., for sources and end-users. For both inputting and retrieving new information, the interactive end-user interface is a programmatic man–machine dialog (Fig. 3). The exchange of information between the IS and the outside world happens through documents. Through *three basic data interfaces*, the interchange of data can be performed as the source data for input, querying and generating response based on data within IS for end-users, and exploiting the dialogs that occur as “web-editable” interface. The documents that may be passive, interactive, or dynamic can be manipulated in specific languages and standards. The new properties of Web technologies are that a document (a Web document) is placed directly between the process of inputting, inserting, retrieving and the interactive end-user interface. The interaction is carried out by the document as information holder between end-users and database. The red line tries to reflect this operation. The links between the documents and database are handled automatically. The specific processes for data handling should be customized by programming both the interaction with end-users’ dialogs and with the database management system. The transactions of database management system are attached to documents, possible through several level hierarchy. The correct execution of transactions is the responsibility of database management system. The documents of end-users are semi-structured data, supplemented with a presentation layer. The *free documents* are document types containing “free” variables, i.e., *unbounded* variables (Fig. 4).

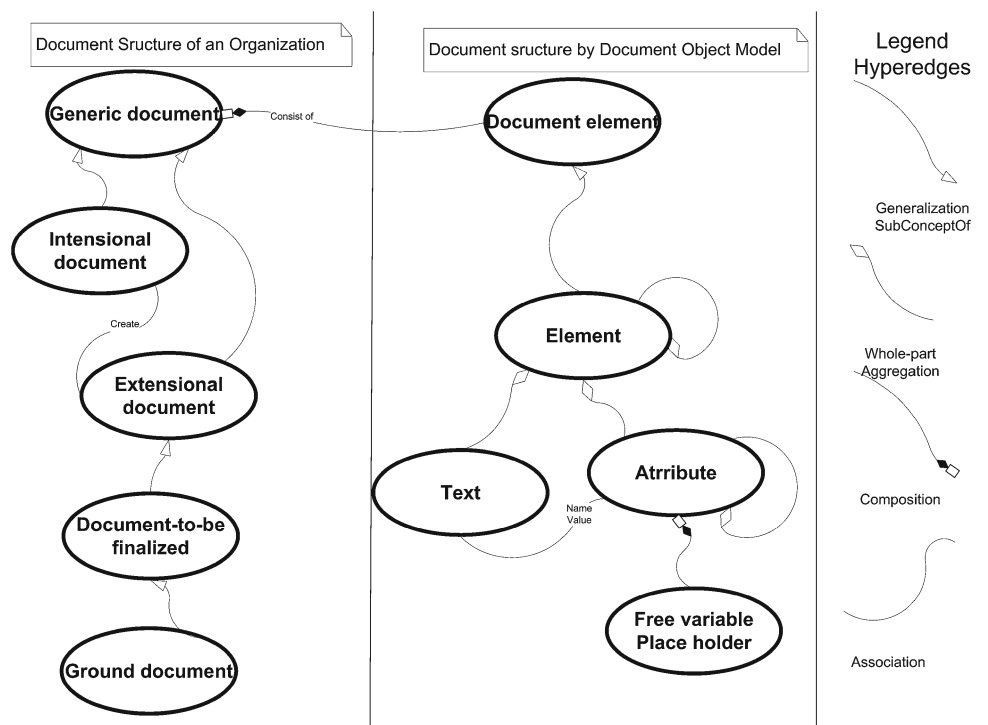
The accurate and valid handling of document is the responsibility of Web service management system. The semantic validity of documents as a property can be perceived as the set of relationships among documents that are semantically correct, valid in the context of other documents, business processes, and activities, and the general operation of business units. The Web pages, the messages between Web services conveying data and/or documents are important components of IS. Semi-structured data occur in the form of XML/HTML documents for the users on the network (Intranet, Extranet, and Internet) and in IS. Displaying information for the users and requesting input from the users are the two facets of IS considering the information exchange.



**Fig. 3** Information systems' model in a document-centric approach



**Fig. 4** Dependence hypergraph for document concept and structures



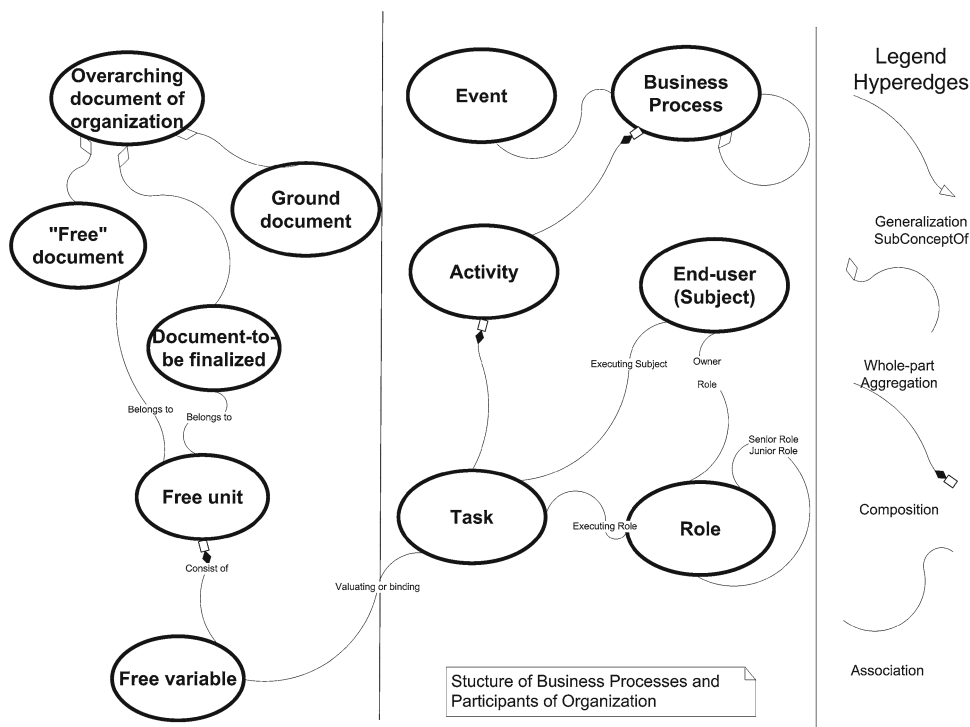
**3.4 Application of hypergraph theory for the document-centric approach**

Firstly, we should bring to mind the basic definitions of hypergraphs in order to apply for description of the interrelated phenomena between documents and business activities.

**Definition [7]** A hypergraph  $H$  is a pair  $(V, E)$  of a finite set  $V = \{v_1, \dots, v_n\}$  and a set  $E$  of nonempty subsets of  $V$ . The elements of  $V$  are called vertices, the elements of  $E$  edges.

The notion of hypergraph may be extended so that the hyperedges can be represented—in certain cases—as ver-

**Fig. 5** Dependence hypergraph for depicting the structure of ontological concepts for business processes and participants of organization



tices, i.e., a hyperedge  $e$  may consist of both vertices and hyperedges as well. The hyperedges that are contained within the hyperedge  $e$  should be different from  $e$ .

The hypergraph approach for modeling IS from a document-centric viewpoint provides an algebraic toolset for analyzing, conformance, compliance, and consistency checking of the model. Thus, the model can be exploited for design and operational purposes as well (Fig. 5) [7].

The concept of generalized *hypergraph* [7] seems to be a construction that is apt for unifying all viewpoints, perspectives, artifacts, and modeling elements. A *graph* represents relationships between pairs of nodes; a *hypergraph* represents relationships among sets of nodes. Thus, a *hyperedge* may interconnect multiple nodes/vertices. The concept of hypergraph can be generalized by allowing hyperedges to become nodes/vertices. Concepts related to IS are better represented as generalized hypergraph than ordinary graphs (Table 1).

We propose an attempt to use the hypergraph for the complex exercise of modeling IS for the reason that the graph representation approach is very apt to depict complex logical, conceptual, and computational relationships in a two-dimensional representation. As we have tried to outline in some earlier publications, there exists a set of complex relationships among the artifacts of IS that can be described by the assistance of concepts as the enterprise and information architecture [21,22]. The intrinsic property of IS is that the underlying models within the overarching architecture are strongly coupled to document types and documents

that serve—at the same time—as source of data and place holder for the output. We have drafted a mapping among the concepts and components of IS to basic concepts of a generalized hypergraph in Table 1. The immanent hierarchical structure of IS promotes the application of generalized hypergraph instead of plain hypergraph. The generalized hypergraph can be perceived as a structured graph, thereby we can represent complex relationships between documents and their variables and Web services that manipulate them. The generalized hypergraph provides the opportunity to support modeling and designing IS, i.e., describing requirements and then the requirement specifications. By exploiting features of generalized hypergraphs, the refinement, decomposition, abstraction, modularization, componentization, and composition can be described through vertices and hyperedges. Utilizing the mathematical theory of hypergraphs in the background, the model of IS can be scrutinized with the assistance of algorithms to check the consistency between models and their elements, for e.g., conformance and compliance to requirements at various architecture levels and at different granularities [7]. The architectural model of IS that combines Blokdiik’s approach, the axiomatic design, and the Zachman framework aims at representing the structure of overall system in the sense of particular models, components within models, the interactions among them, the patterns of architecture, of architecture building blocks [20,21,28,31].

The above-outlined model for IS contains some inherently directed relationship types, namely between input sources

**Table 1** Mapping the concepts of information systems onto the notion of hypergraph

Concept of IS theory	Representation of concept in the domain of hypergraph theory
IS	A generalization of a system development design artifact that is described by a set of elements represented by nodes and a relation on them, namely belonging to an information system. We abstract this general perception to a hypergraph that consists of nodes and hyperedges
Node/vertex in a hypergraph	Each node (or vertex) corresponds to an element within IS, e.g., documents, elements of documents (constituting a tree structure), business processes, workflows, layers of workflows, web services, networks of web services, etc.
Edge in a hypergraph	Edge is a specific hyperedge with cardinality equal to two. Edge denotes binary relationships between two nodes, e.g., <i>free documents</i> are consumed by a certain <i>Web service</i> , a generic document is the ancestor of an intensional documents; a free document results 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 <i>disjoint</i> node dedicated to <i>modeling the environment</i> of the system, plus all the nodes, and hyperedges of the IS
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 sub-systems along with the disjoint, environment node

and business processes, furthermore output sources and Web services, documents and service activities dedicated for data manipulation. The generalized hypergraph enables the representation of hierarchical relationships; however, we need to display the direction of relationships.

**Definition** [7] A directed hypergraph is an ordered pair

$$\vec{H} = (V; \vec{E} = \{\vec{e}_i : i \in I\}), \quad (1)$$

where  $V$  is a finite set of vertices end  $\vec{E}$  is a set of hyperarcs with finite index set  $I$ . Every hyperarc  $\vec{e}_i$  can be perceived as an ordered pair

$$\vec{e}_i = (\vec{e}_i^+ = (e_i^+, i); \vec{e}_i^- = (i, e_i^-)), \quad (2)$$

where  $\vec{e}_i^+ \subseteq V$  is the set of vertices of  $\vec{e}_i^+$  and  $e_i^- \subseteq V$  is the set of vertices  $\vec{e}_i^-$ . The elements of  $\vec{e}_i^+$  (hyperedges and/or vertices) are called *tail* of  $\vec{e}_i$ , while elements of  $\vec{e}_i^-$  are called *head*.

In order to link the documents and Web services and business processes of IS to each other, there is an opportunity for mutual linking. Relating certain models and their elements to each other provides the opportunity in modeling, designing, and operational time for controlling and checking for consistency, compliance, and conformance. The compliance

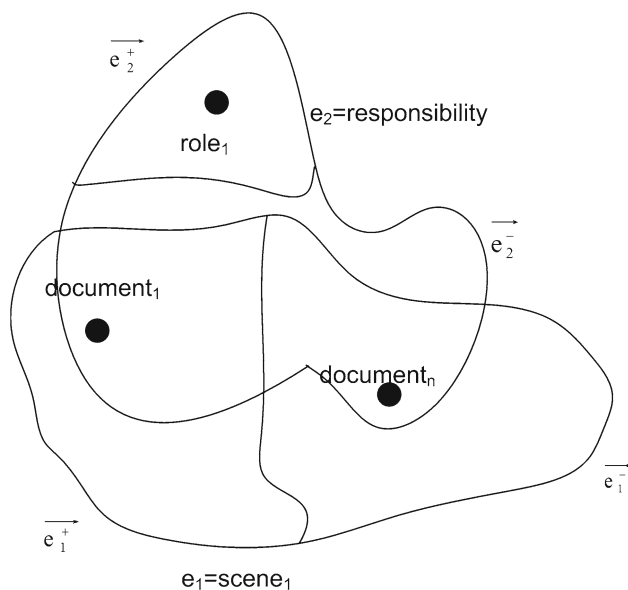
may include the issues related to security at document and process level as well.

To create an overall model exploiting hypergraph representation besides the taxonomy and hierarchy of documents, we need the basic concepts of business activities and processes embodying in the form of IS services. We have delineated taxonomy for documents and document types. In the case of IS, the business processes and information services have been ordered into a framework by the story algebra approach that is based on the process algebra formal method (Fig. 6) [27].

(a) *Story space* we can perceive it as a hypergraph that contains all the relevant element of information space that belongs to a specific IS. The elements of information space appear as vertices.

(a) *Story board* can be interpreted as directed, generalized hypergraph. *Hyperarc* designates the transformation that happens between documents or document types represented by vertices. The transformation is a *state transition* and can be identified with an elementary task within a business process and mapped to a service activity of IS services. Besides documents, the organization units and the roles within them can be represented as vertices. Documents in the form of vertices included in





**Fig. 6** A part of a directed hypergraph for story board

scenes can be related through hyperedges and hyperarcs to roles within organization unit.

- (b) *Scenario* is a subdirhypergraph (directed subhypergraph) of the story board.
- (c) *Story* is a directed *path* within the story board being a directed hypergraph (dirhypergraph).
- (d) A *scene* is a hyperedge that can be displayed as hyperarcs. Some hyperarcs contain documents and document types as tail and head of the given hyperarc representing the transition of documents. Other adjacent hyperarcs contain documents and document types as *head* and roles and/or actors of interested organization units as *tail* designating the responsibility for manipulation.
- (e) The *intensional document type* is manipulated by a scenario that comprises scenes in order to implement a task of business processes, the aim of which is to create extensional documents as instances of free documents.
- (f) A *scene* is elementary part or task of information systems services, or a kind of Web services; it is represented by a hyperarc that symbolizes the transition from one state of documents to the other.
- (g) The life cycle of documents can be described by a *directed path* through the dirhypergraph, the documents may contain one or more *scenarios*. The *ground documents* appear as *head* of certain hyperarcs as the final state of documents' manipulation.

A scenario contains scenes; a scene is linked to a task. The scenario can be perceived as a complex business process or activity, in practice this set of activities is described in

a disciplined graph structure following BPM notation. The directed hypergraph model provides the opportunity to treat all aspects of IS and to reason about the soundness of the model. The unified model framework can depict the access rights and roles within the organization are attached to documents, activities, and tasks, respectively.

## 4 Illustrative example

### 4.1 The method for modeling

The basic steps of the proposed modeling method are as follows:

1. The overarching business process is modeled in a BPM notation that has as theoretical background a formal, process algebraic transcription method as well. The element of the specific business process is systematically arranged into scenarios, scenes, and elementary processes that can be formalized by process algebra.
2. The documents manipulated in the business process should be collected. Firstly, a unification of documents should be created that represent a comprehensive document belonging to the overall business process.
3. The unified document should be modeled and analyzed. The document used in the real world by a business process should be described and fit into the major types and document states as whether it can be put into generic, intensional, and ground document types according to valuation of the free variables.
4. The access and manipulation *rights* of each single variable allow the definition of responsibilities of organization roles and the permitted operations.
5. The document structure and the network of elementary processes have been organized into a model that combines *organization* and *process* aspect of the whole system. The document structure as a network of document objects can be mutually mapped onto the networks of elementary processes. Both structures—the documents and processes—contain hierarchy and grouping substructures that can be mapped mutually, i.e., the structures of sub-documents onto scenarios, scenes, and processes.

### 4.2 Combining the approaches to provide a formal description

As we have tried to outline above, the information objects appear typically in the form of documents at the user interface of IS dedicated to support business services. The before-mentioned approaches (*Enterprise and Software Architecture, Axiomatic Design, Process/Story Algebra*) try to synthesize the data and process aspects.

As we outlined the modeling steps in the previous section, the model reflects the properties of organization and business processes. The *document* represents the dynamically changing *content* and *structure* that is strongly coupled to the *process* structure. The formalization of document–process symbiosis consists of the following:

1. The *pre-* and *post-conditions* of processes are formulated in the sense of input data and formally described by propositional logic.
2. The interrelationship among the processes can be formally described using process algebra within scenarios and scenes.
3. The document manipulation steps and the intention of organization roles—i.e., binding procedure of the free variables—are represented by processes within scenarios and scenes and the related free variables of generic document type.
4. The state—transition of the document modeled by versions and types of the document and the related processes that fulfill the free variables.
5. The organization roles along with their information need and permissions of data/variables manipulation.
6. The overall context is represented by the description of the overarching business process, scenarios, scenes, and steps. The context is consistent with the tasks and history of variables within documents.

The post- and pre-conditions of processes and relationship to the input document content and event can be described by story algebra as follows:

1. Reasoning about the process chain penetrating through the architecture tiers.
2. The business rules and processes provide the dynamic, operational side appearing in the form of pre-, post conditions of processes, including the document and data structure.
3. Stepping through the tiers of architecture results in data transformation from loosely structured (semi- or unstructured) document to rigidly structured data records.
4. The input event—as it can be seen above—can be interpreted as the pair of document or list of data and the instance of the particular event that is related to a particular scenario, scene, or process (Fig. 7).

The architecture of IS is a complex structure as we have attempted to outline in Figs. 1 and 3 and in Ref. [21]. The algebraic reasoning with the processes can be used to verify the design and to provide a mechanism for controlling runtime to maintain compliance and security [27].

### 4.3 Case study

In an e-government environment, the demand for a *secure* and *reliable* document and message exchange mechanism and supporting system has arisen. A general official document handling business model was created that focuses on documents and their manipulation. The major business processes for treatment of documents are as follows:

- Documents are sent by citizens as private individuals.
- Registering the arrived documents and identification of the sender citizen.
- Allocating the electronic document to the responsible public servant.
- Administration procedure for treating the issue of the document.
- The filing procedure of documents concerning various pre- and post-conditions.
- Creation of official documents.
- Revision, approval, digital/electronic signing of documents.
- Electronic document exchange between offices and agencies.
- Receiving, registering, and filing paper-based documents.
- Storing in the record office, archiving, disposal, and demolition.

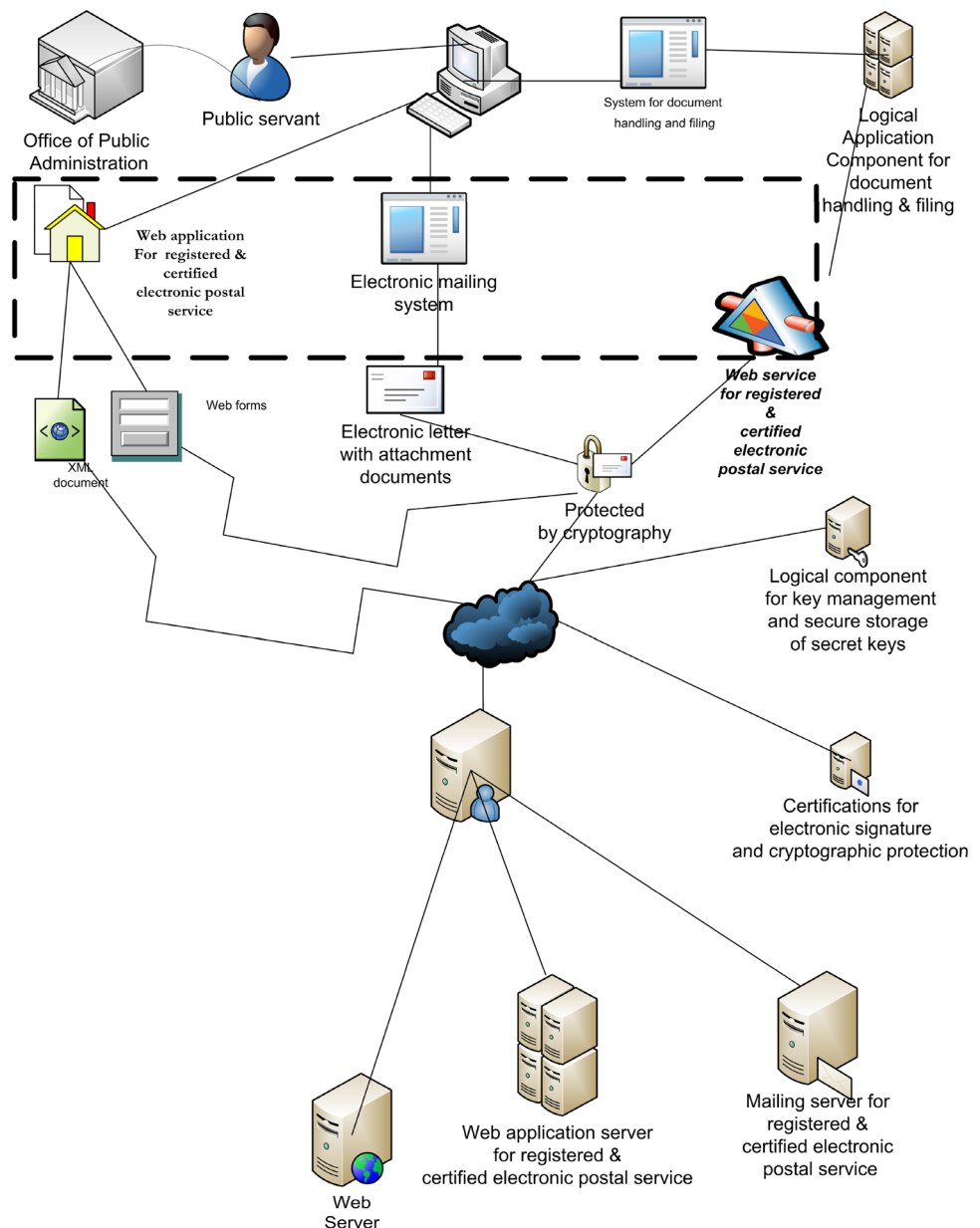
A system for transferring messages and documents are planned. For the reason of the unified and uniform handling, the paper-based documents are transformed into electronic ones. There are two types of electronic document in this system: registered and certified. In both cases, the basic processes are as follows:

- Sending/receiving electronic documents.
- Returning receipts.
- Acknowledging the reception of message envelope and/or message content.

The activities and tasks of processes can be carried out through Web interfaces that appear within human–computer dialogs as interactive documents. Through the Web interfaces, Web services can be accessed. The Web services are provided by Message-oriented Middleware, e.g., solicitation for receiving a document, retrieving the received documents, querying content of meta-data of received documents, filing documents into office record system, etc.

The focus of end-users' thinking is on documents. The public servants describe the administration processes in terms of documents, the employees of postal services discuss the life cycles of postal matters: firstly paper-based, secondly electronic ones. The window towards cyberspace is an inter-

**Fig. 7** A certified and registered postal matter services for exchanging documents with public administration



active document. The Web interface that is itself a document offers the services for handling electronic postal matters and their document content.

The proposed method helped give a structure of the chaotic interrelationships among document-like entities. The free variables as the subject of electronic operation can be mapped to roles that were abstracted away from heterogeneous organization structures, thereby laying the groundwork for a role-based access right system. The meta-data of documents constitute a specific collection mapped into a part of logical database. The other free variables—both in a single document and in an interactive document within the web interface—compose other specific collection in the database. The complex interrelationships that emerge from

various abstraction levels at the different layers of architecture and models can be refined and tracked systematically using the proposed method. The method during the refinement and decomposition of design created opportunity for cross-checking and exploiting the dichotomy of opposing perspectives for quality assurance.

## 5 Research method and evaluation

Our goal was to develop a formal approach that may be used as a method for describing and analyzing IS. The intrinsic nature of methodology development is that the applicability of the proposed method can be investigated through case

studies. The applicability of the approach can be evaluated through qualitative analysis rather than quantitative examination [13]. The case study paradigm as an evaluation tool is properly fitted to the study of formal methodologies devoted to IS [19]. A proposed methodology can be assessed by the impact on the functional and non-functional features of IS [3, 11, 15]. As the proposed method has been developed most recently, we had the opportunity to carry out only one case study and to observe it.

Our research focus was neither a thorough analysis of the proposed method on a significant statistical basis, nor have we planned a numerical measurement method for each single feature. Although as a side effect of the case study, we have observed the differences of our approach and the software engineering approach that was pursued by the development team, and assessed on a subjective basis to provide a “touch-and-feeling”. The applied software engineering approach used by the development team ensured conformance to Java object-oriented and API methods, although leaving out Web services concepts.

Our proposed model concentrates on the most modern Information Systems that present as immanent feature the document-centric communication and user interfaces, extensive exploitation of Web technologies, and a selection of functionalities provided for Service-Oriented Architecture.

Our outlined approach supports decreasing the *complexity* of systems, which facilitates the *transformation* of the models during the refinement of engineering process. Furthermore, the formulation of *Business Rules* through abstraction from documents can be alleviated. Even without automated tool support, the conversion of models can be *traced* for monitoring and later evaluation.

The formalized approach provides a *high level of abstraction* that makes possible to use *standardized* description and representation formalism and notation. The *conceptual modeling* is fully assisted by the document-centric approach along with the story and process algebra, moreover the mutually mapping between documents and data collections. The categorization of document types and coupling to organizational roles yields the opportunity for the systematic and meticulous definition of *interfaces*. The combination of document-centric approach and process algebra methods offers the chance of modeling of *content*, *presentation*, *navigiation*, and *business features*.

## 6 Conclusion

The structure description elements of documents—generic, intensional, free documents, to-be-finalized, and ground documents—can be perceived as a meta-database. This meta-database contains not only static structures, although active

component too that can be implemented by web services. The *active component* incorporates the potential *program code* for interactions among the system roles, documents, etc. The active components enclose the codes for database handling as well.

The contribution of this paper is that the above-mentioned techniques can be integrated into a unified framework (Fig. 1). Although, there is still a lack of a comprehensive and not too complex scheme that combines all elements required for modeling IS from a document-centric viewpoint. Our proposal takes a step into the right direction that both theoretic modeling and engineering viewpoint can be vindicated in a unified approach and it is tractable from a computational viewpoint.

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

Using of successful methods for single particular views, viewpoints, and models, a framework for unifying the various approaches is outlined. To provide a theoretically sound but reasonably complex and comprehensive approach for description and research of IS a hypergraph-based method is proposed (Table 1).

The above-outlined theoretical approach will be operationalized through leveraging most modern database technologies, namely hypergraph databases to represent the models of IS and carry out analysis and verification of models in the sense of compliance, conformance, consistency and soundness of design.

The experiences gained from the case study can be summarized as follows: The end-users and even developers think in term of documents when formulating either the requirements or information exchange ideas. The proposed method helped systematically translate the documents and coupled business processes into more rigid and exact representation that fit to the disciplined modeling principles of IS.

On having chance only for one case study, we have compared and evaluated our proposed methods in the light of the actually used software engineering approach and pinpointed the advantages of the proposed methodology, although by a qualitative assessment.

**Acknowledgments** This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (Grant No.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Allamaraju, S.: RESTful Web Services Cookbook. O'Reilly, Sebastopol (2010)
- Atzeni, P., Meriardo, P., Mecca, G.: Data-intensive web sites: design and maintenance. *World Wide Web* **4**, 21–47 (2001)
- Azuma, M.: SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality. In: European Software Control and Metrics Conference (ESCOM), pp. 337–346 (2001)
- Benczúr, A.: The evolution of human communication and the information revolution—a mathematical perspective. *Math. Comput. Model.* **38**, 691–708 (2003)
- Bernauer, M., Schrefl, M.: Self-maintaining web pages: from theory to practice. *Data Knowl. Eng.* **48**, 39–73 (2004)
- Blokdijk, A., Blokdijk, P.: Planning and Design of Information Systems. Academic Press, London (1987)
- Bretto, A.: Hypergraph Theory: An Introduction. Springer, Berlin (2013)
- Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: a generic architecture for storing and querying rdf and rdf schema. In: The Semantic Web—ISWC 2002, pp. 54–68. Springer, Berlin (2002)
- Chiu, C.-M., Bieber, M.: A dynamically mapped open hypermedia system framework for integrating information systems. *Inf. Softw. Technol.* **43**, 75–86 (2001)
- Crockford, D.: The application/json media type for javascript object notation (json). <http://tools.ietf.org/html/rfc4627> (2006)
- Desharnais, J.M., Abran, A., Suryn, W.: Identification and analysis of attributes and base measures within ISO 9126. *Softw. Qual. J.* **19**(2), 447–460 (2011)
- Gábor, A., Kő, A., Szabó, I., Ternai, K., Varga, K.: Compliance check in semantic business process management. In: On the Move to Meaningful Internet Systems: OTM 2013 Workshops, pp. 353–362. Springer, Berlin (2013)
- Gerring, J.: Case Study Research: Principles and Practices. Cambridge University Press, Cambridge (2006)
- Herbsleb, J.D., Grinter, R.E.: Architectures, coordination, and distance: Conway's law and beyond. *IEEE Softw.* **16**. doi:10.1109/52.795103 (1999)
- ISO/IEC FCD 9126-1.2, information technology—software product quality. Part 1: quality model (1998)
- 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)
- Köppen, E., Neumann, G.: 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 (1999)
- Marini, J.: Document Object Model: Processing Structured Documents. Osborne/McGraw-Hill, New York (2002)
- Miles, M.B., Huberman, A.M.: Qualitative Data Analysis: An Expanded Sourcebook. Sage, London (1994)
- Molnár, B., Benczúr, A.: Issues of modeling web information systems: proposal for a document-centric approach In: CENTERIS 2013—Conference on ENTERprise Information Systems—Aligning Technology, Organizations and People. Elsevier, Lisbon, Paper 65 (2013)
- Molnár, B., Benczúr, A., Tarcsi, Á.: Formal approach to a web information system based on story algebra. *Singidunum J. Appl. Sci.* **9**, 3–73 (2012)
- Molnár, B., Tarcsi, Á.: Design and architectural issues of contemporary web-based information systems. *Mediterr. J. Comput. Netw.* **9**, 20–28 (2013)
- Nama, C.-K., Jang, G.-S., Ba, J.-H.: An XML-based active document for intelligent web applications. *Expert Syst. Appl.* **25**, 165–176 (2003)
- OASIS: 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)
- Open Group: TOGAF: The Open Group Architecture Framework, TOGAF® Version 9. <http://www.opengroup.org/togaf/> (2010)
- Rossi, G., Schwabe, D., Lyardet, F.: Web application models are more than conceptual models. In: Advances in Conceptual Modeling. LNCS, vol. 1727, pp. 239–252. Springer, Berlin (1999)
- Schewe, K.-D., Thalheim, B.: Reasoning about web information systems using story algebras. In: Benczúr, A., Demetrovics, J., Gottlob, G.P. (eds.) Advances in Databases and Information Systems. LNCS, vol. 3255, pp. 54–66. Springer, Berlin (2004)
- Suh, N.P.: Axiomatic Design: Advantages and Applications. Oxford University Press, New York (2001)
- W3C. Web Services Description Language (WSDL) 1.1. Web Site (2001). <http://www.w3.org/TR/wsdl> (2001)
- Webber, J., Parastatidis, S., Robinson, I.: REST in Practice: Hypermedia and Systems. O'Reilly, Sebastopol (2010)
- Zachman, J.A.: A framework for information systems architecture. *IBM Syst. J.* **26**(3), 276–292 (1987)