Issues of Modeling Web Information Systems
Proposal for a document-centric approach

Bálint Molnár a, András Benczúr a *

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

Abstract

There have been already several researches on modeling various aspects of WIS. In our approach, we concentrate on the organizational and process modeling and we put the emphasis on documents that represent the information of enterprises in the form of unstructured and semi-structured documents. The assembly of documents reflects implicitly or explicitly the organization of enterprise and the network of business processes and the activities and tasks within processes. The documents represent at the same time the organizational roles along with tasks and activities. Our modeling approach concentrates on the co-existence and co-operation of documents and activities of business. The Story Algebra, or more generally the process algebra approach provides a formal framework that promises a formal describing method for modeling precisely the event triggered processes coupled with data in document format within an Enterprise Architecture Framework.

© 2013 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of CENTERIS/HCIST.

Keywords: Web Information System; Document-centric Process and Data Modeling, Story Algebra, Process Algebra;

1. Introduction

As the information technology for Enterprise Resource Planning systems and generally Information Systems moves toward Web Services, Cloud Computing environment and SOA, the documents that – at the same time – incorporates the end-user interfaces and significant content of the information exchange plays outstanding role...
within the information processing procedure. The user interface is populated by various document types as structured, semi-structured and unstructured.

The Web Information System (WIS) within an enterprise or organization has an architecture that can be mapped to one of the enterprise architecture paradigms. A document-centric approach attempts to depict both the life cycle and life history of Information System – i.e. the changes during design and operational time – from a viewpoint that puts the changes in structure and content of documents in the focus. Moreover, the documents assist to grasp both the dynamic and static facet of IS behavior.

A WIS contains beside conventional elements documents described by XML that can be handled as semi-structured documents. Within WIS XML documents draw on active hyperlinks and on other characteristic that can be exploited by Web services.

In this article a proposal will be shown that outlines a modeling approach that takes into account the inherent document-centric operation style of modern enterprise IS, the enterprise architecture approach and most recent software architecture approaches.

We will present the proposed approach and the relationship to the previous and similar methods that can be used in phases of analysis, design, implementation and operation.

We will outline the issues that were created by the most recent software architecture and technological development as SOA, Cloud Computing and their combination. The topics that should be investigated and requires some response from modeling methods: (1) the focus point of information exchange between end-user and IS moved slowly from structured documents towards semi-structured and unstructured document; (2) the dynamic side of man-machine communication is manifested in services, Web services that processes the input data.

There is a strong coupling among the data, document and the service of IS at information architecture level. To understand the proper behavior an adequate approach is needed.

There are attempt in the literature that try to describe these phenomenon by various method as e.g. Business Artifact [1], adaptive documents (ADoc)[3], adaptive business objects (ABO) [4], and lately business entities [5].

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

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 [6].

Another article [7] shows a design methodology for a disciplined design process to structure and sustain large amounts of data in a Web site. For large-scale WIS design, Ref. [8] encompasses a method.

To assists the understanding the complex behavior of WIS the enterprise architecture approaches provide a helping hand. Zachman ontology and TOGAF was developed for information systems ([9], [10], [11]). The two frameworks could help to structure both the behavior of WIS considering static, dynamic and operational side.

The service orientation is emerged as new software architecture paradigm and SOA is a kind of reference architecture or architecture style that independent from technology. The SOA helps to arrange and to make use of business services in the form of business capabilities and resources that may belong to various business functions [11]. 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 set of principles to guide the design of software architecture that has, as a focus point, the concept of “service” or “Web service”. [12]:


“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

The above-mentioned definition focuses primarily on the software technology side of Web services. However, the Web services functions within an organizational environment. The Web services should supply information for the end-users; consequently the structure of information mapped onto a specific document should reflect the organizational structure, roles and their access rights, and network of activities emerging within business processes.

The emerging paradigms of Service Oriented Computing (SOC), Cloud Computing put emphasis on services as a uniform and general information exchange interface towards end-users. There are various input data format for communication to services: (1) HTML pages, (2) SOAP messages, (3) unstructured documents (XML). The semi-structured documents comprising hypermedia or hypertext document has a significant role in WIS ([13], [14], [15]). Documents played key role in displaying, interchanging, and retaining information previously. In a technology, electronic environment documents that are designed and structured for end-users have become the emblematic layout for the WIS user interfaces using the HTML.

The service-centric approach in a socio-technological environment can be formulated that services are implemented by electronic means to satisfy demands. The adaptive documents (ADocs) can be understood as a “domain artifact” that presents state-dependent behavior. Within this context the domain artifact designates the representation tool by which the problems and their solution can be described. The relevant properties of ADoc [3]:

1. dynamically varying data environment;
2. state-dependent behavior as the domain artifacts responds to organization level events;
3. support for cooperation between “artifacts” and “services”.

The adaptive business objects (ABOs) [4] represents and abstracts a business entity. The steps of the life cycle depict the history of the entity. The state transition of ABO is represented by a finite state machine (FSM). The external events and events at organizational level creates impact on WIS and the IS responds with state transition. The data is not contained in an ABO even though it uses a data graph that dynamically combines information from different sources. The ABO can be perceived as a holistic approach.

The business artifact can be regarded as synthesis of the process and document aspects ([1], [2]). A business artifact consists of requirements related to data and processing that are business-relevant. The business artifact is another modeling method that deals with information entities along with processing goals in a unified framework. The business services or tasks cause effects on business artifacts by business rules. In this framework, a business process model comprises business artifacts, services/tasks, and business rules.

The business entity as self-contained notion is formulated as „A business entity is a dominant information entity with an associated data model and an associated behavior model in the context of a process scope” [5].

The tasks of business processes have goals and some rules that can be described by algorithms according to some business process modeling approaches. There are competing theories and concepts that use the following notion: business activity, business process, business service, business function and task. There is an attempt in [5] to provide a comprehensive and unified set of definition for these business related notion.

3. The document-centric modeling approach

As the document-centric flow within business IT environment has become the basis; this phenomenon can be perceived as evolution of the information processing paradigm, shifting from the process-oriented, data intensive application typically represented and designed in object-oriented style towards document-oriented
computing. The development of information technology has led to the active semi-structured documents. The documents – e.g. the form documents intended to collect data –, contain procedures for information processing as it is required at a given time at a certain position within the workflow, thereby the documents demonstrate active behavior. The document-centric approach can be characterized by the document structure, content and behavior.

Table 1. Zachman architecture’s relationship to WIS

<table>
<thead>
<tr>
<th>J. A. Zachman</th>
<th>what</th>
<th>how</th>
<th>where</th>
<th>who</th>
<th>when</th>
<th>why</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designer (Logical)</td>
<td>Logical Data Model</td>
<td>Application Architecture</td>
<td>Web Services, Semi-structured documents</td>
<td>System Geographic Deployment</td>
<td>Human Interface Architecture</td>
<td>Processing Structure, Orchestration</td>
</tr>
<tr>
<td>Builder (Physical)</td>
<td>Data Model</td>
<td>System Design, XML/HTML documents</td>
<td>Architecture/Additive/Technology</td>
<td>Presentation</td>
<td>Control Structure</td>
<td>Rule Design</td>
</tr>
<tr>
<td>Functioning Enterprise</td>
<td>Data Function</td>
<td>Network</td>
<td>Organization</td>
<td>Schedule</td>
<td>Strategy</td>
<td></td>
</tr>
</tbody>
</table>

Axiomatic Design Theory ([16], [17]) is an approach for structuring design problems, especially complex information systems. Axiomatic Design Theory (AD) defines four concepts, namely: domains, hierarchy of design artifacts, refinement and decomposition steps between domains and element of hierarchy, and furthermore two axioms. “Independence Axiom: Maintain the independence of the functional requirements; Information Axiom: Minimize the information content of the design”.

The axiomatic design approach can be applied to modeling and design of complex systems with combining an enterprise architecture approach, e.g. Zachman ontology and TOGAF (Table 1, Table 2.). The documents can be modeled using formal and semi-formal approaches ([18], [19]). The aim of the combination these disparate methods are to improve the quality of software in complex Web information systems that are ubiquitous in a mobile environment. This method provides an approach to the full life cycle of WIS that is complex, integrated systems consisting of software, hardware, humans and business processes. The proposed methodology promises a systematic framework to model WIS using disciplined design and engineering method. The descriptive and design methods for enterprise architecture refer to models that can be used to represent a common piece of a specific viewpoint and a perspective.

Using the set of basic concepts of axiomatic design they can be mapped to more adequate concepts of WIS as the User Requirements (UR), the Functional Requirements (FR), the Design Constraints and Parameters (DC&P), and Implementation Factors and Operational Variables (IF&OV) (Table 2.). The table tries to pinpoint the role of documents in the process of design refinement and the continuous cross-checking between the
various models representing on the one side of decomposed requirements and the other side of the results of creative design steps.

![Diagram of WIS integrated model based on Blokdijk and Zachman approach (24), (9); (b) Document processing in Web service perception](image)

Blokdijk’s models: (1) **organizational model** that represents the structure and method of work in the enterprise; (2) **information model** which reflects information, content, its origin and method of descending; (3) **data model** that displays the real world objects about which information is kept along with their relationships that lays the ground for a data implementation model; (4) a **process model** that depicts the composition of the real world **activities** and the strongly coupled **control structure**.

### 3.1. The document-centric perspective

The Internet technology accelerates that business processes will be implemented by Web-based applications; the information system services appear as Web services, sometimes as a Software-as-a-Service through Cloud Computing. Web-based document maintain the structured and semi-structured data through the computer-user interface of the WIS, and describe the dialogue between users (clients) and the WIS. The Web pages, the messages between Web services conveying data and / or documents are important component of WIS. Semi-structured data occur in the form of XML/HTML documents for the users on the network (Intranet, Extranet, and Internet) and in WIS. Displaying information for the users and requests input from the users are the two facets of WIS considering the information exchange. The documents formulated by extensible markup language (XML) can describe the relationship and information exchange between the Web pages and the associated business processes.

As we have discussed in the “2. Literature review” section, the construction of document is aligned with the (1) organogram, moreover (2) with the tasks and responsibilities belonging to the roles (end-users), and furthermore (3) with the configuration of activities within business processes.

The dissimilarity between a collection of Web pages, typical Web Applications and a WIS can be described by the following way: A Web IS serves **business processes** (Business Process Modeling, BPM) and is usually tightly coupled to other IS. The WIS itself can also be perceived as database for structured, semi-structured as
well as unstructured document (XML-based). We have cited a Web service definition ([12]) that put emphasis on the (software) technology side, in our approach we focus on the organizational aspect and the co-habitation of documents and business processes.

Besides formalized and structured data – as there exists in database –, documents hold more semi-structured and unstructured information. There is a discernible relationship between data and documents, the system analysis and design methods, and their models generally concentrate on data instead of documents. Data and documents are generated and processed within workflows of organizations. Referring the Zachman ontologies, the co-habitation between the “how” and “who” columns have an impact on the “what” column that represents the “Business Objects” that are manifested in the form of documents and data. The organization structure and business processes are tightly coupled within the enterprises’ workflow which forms the context within that the information can be interpreted. Hence, the modeling efforts of Information Systems – especially WIS – require that data and documents should be considered together with business processes during the modeling exercises. The data and documents and the business processes that create and manipulate them constitute an overarching entity. We may perceive an overall document of enterprise that maps both the organizational structure and the network of activities of business process whereby the two facets of enterprise model appear in one concept.

Based on the before-mentioned approaches, there is a need for an integrated method that deals with the information in either data or document form in a unified framework with the business processes of workflow. Documents and data may incorporate the same information but they represent the information in different manner. In order to put the data, documents and processes into unified framework, the relationship between these three core modeling concepts should be elucidated.

It needs to be taken into account that each document has its own history of life, consisting of creation, editing, review, approval, finalization and release processes. For a complete description of documents, different types of information models are required, e.g. object models representing the documents themselves [19], the relationship models among the documents in the form of some taxonomy, behavior model depicting the dynamic aspects of documents, content models that show the structure and internal dependencies of various documents. The documents are represented typically by XML (eXtensible Markup Language) and DTD (Document Type Definition), JSON (JavaScript Object Notation) and has an object structure shown by DOM (Document Object Model) [19].

To support the modeling effort of WIS, the document as important component of WIS should be captured either in a formal or semi-formal way, depending on the actual modeling and design requirements. A document type hierarchy can be defined a recursive way to capture the characteristic of document within a WIS:

1. A generic document type can be perceived as “free-document” or “free-tuple”, i.e. a generic document purely consists of variables (place-holders) without binding and itself generates a document type hierarchy. The variables can belong to types; the types can be primitive, enumerative as basic types and tuples, sets, bags as composite types.

2. The intensional document type is a generic document type that owns an instruction or algorithm set for filling in the variables of the document instance; either from a data source that represents the extension of variables and their types or through Web services that maps the human activities into algorithm that manipulates data.

3. If there is a generic document type that is generalization of an intensional document then a new document type hierarchy can be created. An intensional document is an instance of a generic document type through some process of instantiation. During the processing of intensional document, the variables are being bound to data step-by-step. The variables or components of documents can be basic or composite types creating an aggregation hierarchy.

4. The bottommost document is that does not contain any unbound variables. However, we should make some subtle distinction, i.e. a finalized document may be or may not a bottommost document, it depends
on as whether it needs further processing or not, if further manipulation is required it means that it contains implicitly or explicitly at least an unbound variable before it achieves the state of bottommost document.

5. In order to link the documents and Web services of WIS to each other, there is an opportunity for mutual mapping:
   a. The intensional document type can be mapped to a scenario that comprise of steps and/or scenes,
   b. A step can consist of Web services and a service is composed of processes in WIS.
   c. The scenarios or scenes contain the tasks, activities in a disciplined graph structure following BPM notation. The access rights and roles within the organization are attached to either activities or tasks respectively.

If there is a document hierarchy and each document component of hierarchy associated to a task (an elementary step of an overall business process) then the associated tasks can be mapped to a scene as a part of a scenario. A task has a goal and an algorithm consisting of basic procedures that can be identified as processes of WIS.

Thus the static structure of document can be linked to processes of WIS then through the processes the document can be mapped and associated to the relevant structural element of the underlying database or information base.

The Organizational Model (Fig.1, [24]) contains the description of division of work, method of work, the structure of the tasks, activities within business processes in form of BPMN description language ([25]). An overarching business process and the generic document belonging to it that the process manipulates constitute an entity. The specific business units are responsible for definite parts or structure of the documents. During the manipulation of documents, the responsible roles within the organization finalize parts of the document that reaches the state of being bottommost document. The activities and tasks of responsible roles transform the parts of document from a generic document type to a version of intensional document. The particular parts of the intensional document become bottommost parts as the progress of the overall business process approaching the end-state of workflow.

The enterprise architecture and system modeling approaches can be aligned with the document-centric view of WIS. The SOA as a modeling approach concentrates on the separation of concern, i.e. it pursues the classical system analysis pattern, and each model represents a definite view or viewpoint of the system.

Table 2. A schematic mapping between Zachman architecture and requirements and constraints in the framework of Axiomatic Design

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Entities</th>
<th>Activities</th>
<th>Locations</th>
<th>People</th>
<th>Time</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perspectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contextual</td>
<td>UR Intensional</td>
<td>UR</td>
<td>UR</td>
<td>UR Intensional document</td>
<td>UR Intensional document</td>
<td>Scope</td>
</tr>
<tr>
<td></td>
<td>document hierarchy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td></td>
</tr>
<tr>
<td>Logical</td>
<td>DC&amp;P Document</td>
<td>DC&amp;P Document</td>
<td>DC&amp;P</td>
<td>DC&amp;P</td>
<td>DC&amp;P</td>
<td>System Model</td>
</tr>
<tr>
<td></td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td></td>
</tr>
<tr>
<td>Detailed representation</td>
<td>IF&amp;OV Document</td>
<td>IF&amp;OV Document</td>
<td>IF&amp;OV</td>
<td>IF&amp;OV</td>
<td>IF&amp;OV</td>
<td>Components</td>
</tr>
<tr>
<td>(Actual implementation)</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td>hierarchy</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. A schematic mapping between Zachman architecture and requirements and constraints in the framework of Axiomatic Design.
3.2. Modeling steps

The Zachman ontology and TOGAF architecture provide a framework to create models of WIS that strongly coupled to documents appearing at the system interface.

The Contextual and Conceptual perspective can be considered as information technology independent models. A part of User Requirements can be formulated using “Use Cases” either Business Use Cases or System Use Cases. Both use case method describe the concepts of the domain, the basic information exchange patterns consisting of messages, structured data and semi-structured documents. The requirement description in the form of use cases concentrates on the business and system environment, the needs and demands of actors, however the details of processing structure is implicit or not determined.

On the level of Logical Perspective the specification of models is independent from both software and technology platforms. (Fig. 1)

The Physical and Detailed Representation Perspective can be mapped to the software and technology platforms and their components. The software and technology components may include operating system, network configuration, and integrated programming environment. At this level the models concentrates on the specifics of WIS as whether which programming environment, which database management systems should be used, which method should be followed to map the semi-structured documents represented by XML or RDF to the underlying database schema.

A set of models that can be considered as an orthogonal view of Zachman’s Enterprise Ontology are described in ([24], Fig. 1). The mutual mapping to each other results a sparse cube, each elementary cube contains a specific model that fits to the particular architecture view and perspective.

The mappings between models (Table 2.) can be carried out by transformation rules that support the cross-checking for consistency between the models. The Axiomatic Design framework provides a disciplined method to formally safeguard the consistency during the refinement process of design artifacts in either vertical or horizontal direction. The models, the constraints can be specified in a formal language, the document structures can be formalized using a document description algebra based on the XML, DOM (Document Object Model), JSON. The dynamic behavior of WIS is described by process algebra. The transformation rules between the models can be conceptualized in process algebra as well.

The basic steps of the proposed modeling method:

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 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 bottommost type according to binding procedure 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 organized into a model that combines organization and process aspect of the whole system. The document structure as network of document objects can be mutually mapped onto the networks of elementary processes. Both structures – the documents and processes – contain hierarchy and grouping sub-structures that can be mapped mutually i.e. the structures of sub-documents onto scenarios, scenes and processes.
3.3. Combining the approaches to provide a formal description.

As we have tried outline above, the information objects appear typically in the form of documents at the user interface of WIS 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:

1. The pre- and post-conditions of processes is 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, the intention of organization roles – i.e. binding procedure of the free-variables – is 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 fulfills 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, history of variables within documents.

The post- and pre-condition of processes and relationship to the input document content and event can be described by story algebra as it 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 condition, 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 related to a particular scenario, scene or process.

The architecture of WIS is a complex structure as we have attempted outline in Fig 1 (a) and (b) and in [18]. The algebraic reasoning with the processes can be used to verify the design and to provide a mechanism for control in run-time to maintain compliance and security [23].

4. Conclusions

We have outlined an approach that focuses on the organization model of WIS and its document-centric description. The proposed approach combines the enterprise architecture, systematic design and document-centric approach that are conceptually feasible, understandable and implementable. The documents belonging to the business processes and the coupled process structure can be systematically modeled. The models reflect the behavior of WIS that can be realized by Web services. The semi-structured documents can be mapped on to underlying databases that can be relational, object-oriented or XML-based. The mapping of documents can be formulated by transformation rules that using e.g. wrapper technology to link the document object model, the semi-structured object-oriented data model for relational data base management system or any other database technology using structured schema.

The framework in this paper makes possible an integrated handling of business requirements against WIS and software architecture paradigms. Our model of framework that combines three methodologies makes the
modeling efforts more clear, concise that brings into play a synergetic effect whereby achieving business agility through Enterprise and Software Architecture approaches.

Acknowledgements

The Project is supported by the European Union and co-financed by the European Social Fund (grant agreement no. TÁMOP-4.2.2/B-10/1-2010-0030.)

References

[23] Michele Chinosi, Alberto Trombetta, BPMN: An introduction to the standard, Computer Standards & Interfaces, Volume 34, Issue 1, January 2012, pp.124-134,