FORMAL APPROACH TO A WEB INFORMATION SYSTEM BASED ON STORY ALGEBRA

Bálint Molnár*1, András Benczúr1, Ádám Tarcsi1
1Eötvös Loránd University, Faculty of Informatics, 1518 Budapest, POB. 120, Hungary

Abstract: Web Information Systems (WIS) are becoming more and more ubiquitous along with the changes in information technology. There have already been several studies on modeling various aspects of WIS. This paper attempts to investigate the most recent paradigms of WIS description taking into account the Web service approaches, Service Oriented Architecture and Cloud computing. One of the modeling and analysis approaches is the Artifact Centric Business Process that unifies both the behavioural and information aspects of business processes carried out in a WIS environment. The information aspect frequently incorporates in the form of unstructured or semi-structured document that indirectly influences the content of the underlying database through several tiers of the given architecture. A business event stimulates a business process along with some set of data and triggers a process at the user interface level. The process initiates a chain of processes that goes through the tiers of architecture until the content of the database is affected. 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.

Key words: Web Information System, Artifact Centric Process and Data Modeling, Story Algebra, Process Algebra.

INTRODUCTION

The Web Information System (WIS) has become widespread and it incorporates the traditional Information System (IS) that previously realized and fulfilled the business requirements through either data-intensive or transaction-intensive systems. WIS consists of a set of interacting components, web applications for specific functions, information components such as databases as well as other non-web components.

The user interaction side, also known as user interface is dominated by various media types. The business or enterprise related information systems have different document-centric approaches – structured, semi-structured and unstructured.

A business-oriented WIS implicitly or explicitly follows one of the enterprise information system architecture paradigms (Zachman, 1987). The architecture consists of several tiers and views through which the information flows down to the data structure and store and back to the user. The document-centric phenomenon causes theoretical and design problems, both in the static and dynamic aspects of IS.

The Service-oriented Computing (SOC) has made it more clear that the end-user communicates with the service through documents.

The issue raised by the technological development can be summarized as follows:

- the emphasis of communication with IS moved from the structured data towards the semi-structured and unstructured data or document format;

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.).

* E-mail: molnarba@inf.elte.hu
The first encounter of input data with Information System is in the form of services (or processes).

The input data, document and the service that handles it are strongly coupled at logical level, which means that this pair of things can be considered as one unit. Some approaches call that perception Business Artifact (Cohn and Hull, 2009; Hull, 2008). However, the similar concepts have been called by various names, including adaptive documents (ADoc) (Nandi et al. 2003), adaptive business objects (ABO) (Nandi and Kumaran, 2005), and lately business entities (Kumaran et al. 2008).

The paper is organized as follows. In Section 2, we present the literature review. Section 3 outlines our method making use of the previous approaches in a document centric approach. Section 4 provides the summary and conclusions.

BASICS OF STORY ALGEBRA AND BUSINESS ARTIFACTS

The SOC paradigm and the concept of Web-services together yield a suitable style for building collaborative Web information systems (WIS). The advantage of using SOC in the WIS development is that it enables the direct connection to the organization business processes, workflows, activities and tasks. The Business Process Modelling (BPM) realizes this direct mapping between the organizational level services and the information system level services. This direct mapping between the information services of WIS and the business activities and processes can be described by using storyboarding approaches, or more formally, by Story Algebra as a part of the Process Algebra.

The difference between a set of Web pages, typical Web Applications and a WIS can be formulated in the following way: A WIS supports business processes (Business Process Modeling, BPM) and is usually tightly integrated with other IS. WIS can also be viewed as database applications for structured, semi-structured as well as unstructured documents (XML-based).

The semi-structured documents embodying hypermedia or hypertext documents, play an important role in the most modern IS as WIS (Bernauer and Schrefl, 2004; Chiu and Bieber, 2001; Nam et al. 2003). Historically, documents have been used in displaying, interchanging, and retaining information. The form of documents has become the typical layout for the WIS user interfaces using the HTML.

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 to the document-oriented computing. The development of information technology has led to the active semi-structured documents. The documents intended to collect data contain procedures for information processing, as it is required at a given time and at a specific position within the workflow, and thus they demonstrate active behavior. The document-centric approach can be characterized by the document structure, content and behaviour.

The business artifact can be regarded as synthesis of the process and document aspects. A business artifact contains all the data and processing requirements that are business-relevant.

The adaptive documents (ADocs) are perceived as “domain artifact” that exhibits state-dependent behavior. Within this context, the domain artifact means the representation tool by which the issue and resolution can be depicted in the problem area. The relevant aspects of ADoc are as follows (Nandi et al. 2003):

1. dynamically changing data environment;
2. state-dependent behavior in compliance with stimulus-response metaphor as the domain artifacts react to organization - level events;
3. Assistance provided for cooperation between “artifacts” and “services”.

The adaptive business objects (ABOs) (Nandi and Kumaran, 2005) represent and abstract business entities. However, the concept of business entity is not defined exactly within this context but it supposedly designates a group of data and business activities. The states in the business entity’s life cycle describe the history of the entity. The state transition of ABO is depicted by a finite state machine (FSM) that displays all possible states of the entity that may occur during its life cycle. The organizational level and external events cause effects and respond with state transition. The data is not held within an ABO, but it uses a data graph that dynamically combines information from various places. The ABO approach can be considered again as a holistic view, a kind of synthesis of process and data side, but in a specific way.

The business artifact is another modelling method that perceives information entities along with pro-
cessing goals. A business rule regulates which service can be performed, by whom, under what condition, and how artifacts’ states change.

The business services or tasks alter the business artifact by business rules. In this framework, a business process model is composed of business artifacts, services/tasks, and business rules.

The business entity as a self-contained concept is defined as „A business entity is a dominant information entity with an associated data model and associated behavior model in the context of a process scope” (Kumaran et al. 2008).

According to some business process modelling approaches, the tasks of business processes have goals and some rules that can be described by algorithms. There are competing theories and concepts that use the following notions: business activity, business process, business service, business function and task. There is an attempt in (Kumaran et al. 2008) to provide a comprehensive and unified set of definitions for these business related notions.

The Story Algebra tries to formalize the inherent concurrent characteristic of WIS user interface appearing in the form of document or media types (Schewe and Thalheim, 2004; Thalheim and Düsterhoft, 2001) and to create an environment that can be described by the navigation features of user interfaces together with navigation scenes formulated in scenarios.

The end-user and WIS interaction through some document can be grasped as the execution of a particular scene. The SiteLang method uses the concept of media object and media type as its class. The storyboarding provides a tool to express an activity in the form of information exchange between the end-user and a media object. Such a dialogue can be depicted in an activity diagram (e.g. in UML style).

The concepts of modeling in SiteLang can be summarized as follows:

- **Scenario** is a sub-graph of the story board.
- **Story** is a path within the story board.
- **Scene** is a vertex in the story board.
- **Actor** is an end-user, dialogue object, or media object.
- **Media object** is a unit of business- and representation functionality, mediating between the customer and database of the WIS.
- **Content** represents data or information.
- **Stories** are specific scenarios that make possible the separation of concern in order to further reduce the complexity to be dealt with at once.

In the storyboard, the processes representing the usage of WIS can be described by the actors, message exchange and message passing mechanism. The processes should represent a solution for a particular business problem and the processes can be identified with some single scenario when the process contains splits or joins (forks and merges) and maybe loops, if not the process can be considered “only” as a story. The modeled processes can be used to identify the data and operations accessed by the end-user driving the process. For each process, database schemas are created describing the data accessed in a process.

From a methodological point of view, storyboarding is a particular variant of analysis and synthesis, altogether providing a holistic view. The Story Algebra that can be used to specify story spaces can be represented as Kleene algebra with tests (KATs) (Schewe and Thalheim, 2004).

**COMBINING APPROACHES TO PROVIDE A FORMAL DESCRIPTION**

The information objects appear typically in the form of documents at the user interface of WIS dedicated to provide support to business services. The above-mentioned approaches attempt to synthesize the data and process aspects. A document can represent a dynamically changing content and structure. The pre- and post-conditions of processes can be formulated in the sense of input data that can be represented partly by the structure and contents of documents. A business event triggers a chain of processes that goes through the tiers of architecture. The most important components of architecture for the first-cut investigation are the user interface level,
conceptual model and operational or physical data model level. There are several models that business information systems can use to describe the architecture (Zachman, 1987) and to reconcile the most recent architectural styles as e.g. SOA (Molnár and Tarcsi, 2011).

Documents play a vital role in business information systems as a workflow constituent. Data are generally handled in the form of documents which can contain various data items and which build a unit of information within the business process. The exchange of information between different parties of actors, roles and partners within a business is manifested in the form of documents. The format and the structure of the information contained in documents are typically prescribed, either by the application through which the processing of documents happens or by standardized document templates that are quite common in business processes. Different definitions for the document as a technical term can be found in the literature. It usually occurs within these definitions that documents are things or artifacts that are created and used within workflow and business processes. Based on the several competing definitions, a document can be defined as a logical unit of information displaying the product of some business activity. Documents differ widely in form and content, ranging from informal documents containing natural language text, to more formal ones like tables or structures complying to some rules with well-defined syntax and more or less definite semantics. It needs to be taken into account that each document has its own life cycle, consisting of creation, editing, review, approval, and release processes. For a complete description of documents, different types of information models are required, e.g. object models representing the documents themselves (Marini, 2002), relationship models among the documents in the form of some taxonomy, behaviour models 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), structured by XML Schema and DTD (Document Type Definition) and have an object structure shown by DOM (Document Object Model) (Marini, 2002).

The approaches that make use of the concepts such as adaptive business objects (ABOs), business entity, business artifact, attempt to model all aspects of business files and their behaviour in a unified way.

To advance further the modeling of WIS and the recent business-oriented information systems, we attempt to take into account and formalize some aspects of the recent information systems. We can differentiate between the input document containing data and the input event triggering the chain of processes to deal with the document. The input data and the input event define together the set of preconditions that provides the environment to execute a piece of scenarios, stories and consequently processes.

This paper attempts to present a document centric approach that uses the results of the previously mentioned methods.

As a case study, we can make use of a university example that deals with course registration that may demonstrate typical issues at a WIS:

- Student registration to a specific course
- Admission of students to a course
- Closure of the registration period
- Student registration to a specific course examination
- Entering the grades for students
- Closing the exam sheet
- Modifying the exam sheet after closure

A motivating example

A typical business scenario at universities is the preparation for the examination period. The necessary information is handled by a WIS and it is represented through documents at the user interface level. The business activity consists of two essential parts: firstly, the system and the administration collect the list of students that potentially participate in some of the exams. Then, the documents for each single exam will be created at a certain point of time so that the students can concurrently modify only the content of a document and some specific data items. The overall list of students belonging to an instance of a specific course is generated automatically by a business activity and data processing providing a superior summation of data. The relationship between the stories, scenarios and documents linked to the lists of potential examinees is a hierarchical one.

The specificity of the situation is that there is a document produced before the beginning of the course containing information on the admitted students, which then governs the business activity re-
lated to exam preparation. The particular document for a single exam is a derivative of the superior documents in the sense of content as well as structure.

A virtual or *intensional* document exists at the user interface and conceptual level and it contains semi-structured form of all relevant information.

The manifestation of the particular sample of the virtual document is a document accessible at the Web interface. Such a manifestation, which is connected to a business event and coupled to a story/scenario, represents the mapping of business task within a business process. The documents reflect a hierarchical relationship with the virtual documents; the hierarchical relationship can be coupled to the relation of business tasks within business processes. The processes in a story or scenario can be linked to business tasks.

The particular documents that can be considered as a specific view of the virtual document are dynamically created from the virtual, overall document. A particular document as a specific view is strongly coupled to a certain business task.

The business process consists of various tasks. Every task has a goal and reflects the intention of the actor executing the task through some *activities* (Schäfer, 1988). The actors fulfilling some role have

### Table 1. Document for registration to a course in a semester

<table>
<thead>
<tr>
<th>Status-Event</th>
<th>Attributes</th>
<th>An instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Course Id: String</td>
<td>Course Id: IPM-08irÜVIRE/1</td>
</tr>
<tr>
<td></td>
<td>Course Type: String</td>
<td>Course Type: Web technologies in IS (MSc)</td>
</tr>
<tr>
<td></td>
<td>Credit: Integer</td>
<td>Credit: 2</td>
</tr>
<tr>
<td>Open for sign in/sign out</td>
<td>Designated Time: Time</td>
<td>Designated Time: Monday, 08:00-10:00</td>
</tr>
<tr>
<td>Closed</td>
<td>Designated Room: String</td>
<td>Designated Room: Database lab</td>
</tr>
<tr>
<td></td>
<td>Max. Permitted number of students: Integer</td>
<td>Max. Permitted number of students: 40</td>
</tr>
<tr>
<td></td>
<td>Estimated number of students: Integer</td>
<td>Estimated number of students: 40</td>
</tr>
<tr>
<td></td>
<td>Number of students signed-in: Integer</td>
<td>Number of students signed-in: 25</td>
</tr>
</tbody>
</table>

### Table 2. Document for a specific exam

<table>
<thead>
<tr>
<th>Status-Event</th>
<th>Attributes</th>
<th>An instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>Course Id: String</td>
<td>Course Id: IPM-08irÜVIRE/1</td>
</tr>
<tr>
<td></td>
<td>Course: String</td>
<td>Course: Web technologies in IS (MSc)</td>
</tr>
<tr>
<td>Open for a sign in / sign out</td>
<td>Exam type: String</td>
<td>Exam type: Colloquium</td>
</tr>
<tr>
<td>Closed for sign in / sign out</td>
<td>Admitted students and scores: List (String, Integer)</td>
<td>Admitted students and scores: ...</td>
</tr>
<tr>
<td>Open for keying-in grade</td>
<td>Exam date: Date</td>
<td>Exam date: 2012-05-20</td>
</tr>
<tr>
<td>Saved</td>
<td>Exam Time: Time</td>
<td>Exam Time: 10:00 PM</td>
</tr>
<tr>
<td>Open for extending with examinees</td>
<td>Site, room: String</td>
<td>Site, room: Database lab</td>
</tr>
<tr>
<td>Finalized</td>
<td>Max. number of examinees: Integer</td>
<td>Max. number of examinees: 40</td>
</tr>
<tr>
<td></td>
<td>Examiners: String</td>
<td>Examiners: X.Y.</td>
</tr>
</tbody>
</table>
the right to start to execute certain tasks by creating events at business and information systems level.

The description of tasks can be composed of goals, variables / attributes, stories. During the business analysis, the tasks are decomposed into actions. We can map the actions of task to the actions of the story algebra (Schewe and Thalheim, 2004). The pre- and post-conditions for an action can be inferred step-by-step from the conditions belonging to the task. Within Story algebra, an action is directly represented by a process. A task description is a tuple \((G, Pre, Post, Story)\), an action description is again a tuple \((Pre, Proc, Post)\) where:

- **G** is the formulation of the goal of the task;
- **Pre** and **Post** is the first order logical expression of pre- and post-condition;
- **Story** is the associated story within the story space and a scenario;
- **Proc** is the collection of processes (one or more) in the associated story.

The intensional document of a WIS can be perceived as follows: documents and their behavior, i.e. the related set of tasks and the application logic of performing tasks.

The intensional document may be formalized in the following way:

**Intensional_Document**: Document_Object_Structure× Data_Items

**(1)**

**Data_Items** = List Data_Item

**(2)**

**View_of_Intensional_Document**: view_mapping (Document_Object_Structure× Data_Items \(\rightarrow\) Document_Object_Structure× Data_Items)

**(3)**

**View_of_Intensional_Document \(\subseteq\) Intensional_Document**

**(4)**

A specific view of the intensional document is a subset of structure, and the mapping function produces a meaningful structure in the form of DOM and erases the unnecessary symbols while preserving the structure (Hull, 2008).

An intensional document type (or class) may be defined in the following way

- An intensional document type \(IDT\) is a tuple \((Str, Data_Items, BP, Sc...,\).
- \(Str\) is a structure described by DOM and XML (trees or forest).

- **Data_Item** is a collection of attributes of primitive types (such as String, real, integer or document and other ids).
- **BP** is the Business Process linked to the **DT**.
- **Sc** is the general, top scenario that belongs to **DT** and is specified by the Story Algebra processes.

A particular document may be defined with an XML view definition, i.e. is an XQuery expression called a view query (Boag et al. 2011).

A particular document, as a view of an intensional document \(d\), represents an instance of an intensional document type that contains the data items as filled in attribute value pairs. The data items represent the actual status of the document (Cohn and Hull, 2009).

The status of the document is strongly coupled to a story within a specific scenario, and moreover to some process or processes that precondition matches to the status of the actual document.

The functionality can be specified by the inputs, outputs, preconditions, and effects of the intensional documents in a similar way as a service of WIS. These are together known as the IOPEs of the given functionality provided by the intensional documents and according to the style of OWL-S (Singh and Huhns, 2005).

The business tasks can be mapped to scenarios (Thalheim and Düsterhoft, 2001) and stories. The hierarchical relationship among the virtual documents and specific views appearing as single document can be represented by hierarchical scenarios.

The Story Algebra uses three important concepts and those are **story**, **scenario** and **scene**. The manifestation of a document associated with one or more actors and roles; vice versa the actors and roles can handle one or more documents in a business environment depending on the pre-conditions and related events. Due to the many-to-many relationship between the intensional documents and their views and manifestations in particular situations, the actors and roles lead to a similar many-to-many relationship among stories, scenes and scenarios. With our document-centric approach, we can interpret the usage of concepts in Story Algebra:

The acts of actors and roles and the behavior of an intensional document can be described with one or more scenarios, as the intensional document and its instances are linked to business processes.

The business processes consist of tasks; the tasks are composed of activities; the activities can be represented by processes of stories.
Scenes are interpreted as macro-steps of scenarios (Nam et al. 2003) so that they can be used as a structuring tool for clustering some logically related tasks and activities.

The actions are elementary component of tasks that have a goal to be achieved.

An action manipulates some interrelated data within a document instance, and the document as an environment defines the context of the interpretation and the opportunities for the behavior of the action.

When a course organizer creates a course registration opportunity, he/she keys in the data into some XML/HTML documents. The input data in the form of a document represents the data that will be fed into the system. In fact, the input event is enhanced with identifiers that are necessary to associate the input document, the event and actions/processes. The raw input data/document contains list of data items. The data items can be formally defined by attribute names appearing in the conceptual model and logical rules describing the domain restrictions.

\[ \text{Input Document} = \text{Document Object Structure} \times \text{Input Data} \]

\[ \text{Input Data} = \text{List Data Item} \]

The data items should correspond by an appropriate mapping to the attributes of conceptual models. The attributes belong to particular domains and are constrained by some (business) rules.

\[ \text{Data Item Mapping} : \text{Data Item} \rightarrow \text{Attribute} \times \text{Domain} \]

The input event is stimulated by an event from the business environment and it is transformed to a user interface level or information system level event that we consider input event. The decisive feature of the input event is that it enforces the start of process in order to generate data items that are necessary for further navigation within a scenario and to insert or modify the content of the conceptual model.

\[ \text{Input Event} = \text{Event} \times \text{Entity Identifier} \times \text{Document Structure} \times \text{Data Items} \]

The pre-conditions of a process \((p \in P\) set of processes associated to scenes) can be formulated as follows:

- **Pre conditions**
  - If \(\text{Course Identifier}\) is not defined
  - The flow of registration is at the initial state.

- **Post-condition**
  - The involved instances of data items in the document have been altered.

\[ \text{input items (Course Registration)} = \{ \text{document structure, (Course Identifier, string), , ……)} \} \]

\[ \text{create Course Registration (course, semester, type)} \]

The intensional document represents a document class or a document type. The element of the class is again classes or subtypes. The intensional document is a super type; the subtype documents inherit the attributes of the supertype and a part of the structure (Hull, 2008). The instances of a subtype document appear as single, individual entities. The supertype-subtype relationship is an is-a relationship. Thus, the is-a relationship produces a document hierarchy. The general “Course registration”, the more specific “Course registration” for a semester, for a certain type (PhD, MSc, BSc etc.), and an instance of “Course registration document for a student” for a given course, semester, type, embody a document type hierarchy. We can associate actors and roles with a document type and its instances. The document type and actor/role pairs can be mapped to scenes and scenarios.

If the document structure does not play any role in further processing, the effect of the processing goes through the tiers of architecture modifying the content of the conceptual models. The information
loss that may originate from the mapping of either data items or structure contained in the document could be handled by cohesion (Schewe and Thalheim, 2004) or proximity values (Vechtomova and Karamuftuoglu, 2008).

**WIS Architecture and Document and Event Processing**

The pre- and post-condition of processes and relationship with the input document content and event can be described by Story Algebra:

- Reasoning about the process chain penetrating through the architecture tiers;
- 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.
- Stepping through the tiers of architecture results in data transformation from loosely structured (semi- or unstructured) document to rigidly structured data records.

The input event – as can be seen above – can be interpreted as the pair of documents or list of data and the instance of the particular event.

The architecture of WIS is a complex structure as we have attempted to outline in Fig. 1 and in (Molnár and Tarczi, 2011). However, we shall try to restrict our discussion for the sake of simplicity to a clear-cut model Fig. 2.

We restrict ourselves to three architecture tiers in order to outline the problem and proposed approach. Those are the user interface, the conceptual model and internal/physical design. The most recent approaches to system architecture represent disparate architecture artifacts that differ in the sense of

---

**Table 3. Examples for conditions and processes**

<table>
<thead>
<tr>
<th>Event</th>
<th>Pre-condition</th>
<th>Task</th>
<th>Post-condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student makes a registration to a specific course c using the <strong>Course_registration_student</strong> document</td>
<td>current_state(Course_registration_student,initial) ∧ defined(Course_registration_student.c.identifier) ∧ ¬ defined(Course_registration_student.s.identifier)</td>
<td><strong>create_Course_registration_student</strong> (c, s, semester, type), <strong>set_state</strong> (Course_registration_student, created)</td>
<td>current_state(Course_registration_student, created) ∧ defined(Course_registration_student.s.identifier)</td>
</tr>
<tr>
<td>Create a course <strong>Course_registration</strong> document for a course c</td>
<td>current_state(initial)</td>
<td><strong>create_Course_registration</strong> (c)</td>
<td>defined(c. identifier), <strong>set_state</strong> (c, instantiation_enabled)</td>
</tr>
<tr>
<td>Create a course <strong>Course_registration_student</strong> document for a course c, for a semester sem, for a certain type t</td>
<td>current_state(initial) ∧ current_state(Course_registration_student.c identifier, initial) ∧ ¬ defined(Course_registration_student.sem.identifier) ∧ ¬ defined(Course_registration_student.t.identifier)</td>
<td><strong>create_Course_registration</strong> (c, sem, t), <strong>set_state</strong> (Course_registration_student, modifiable),</td>
<td>current_state(modifiable)</td>
</tr>
</tbody>
</table>

---

**Figure 1. A view of WIS integrated model**
design, implementation method and the architecture building blocks used for constructing the system.

The user interface tier handles the input data and/or document. The potential activities can be described by stories and scenarios and the acts within stories and scenarios can be mapped to processes as depicted in (Schewe and Thalheim, 2004). The algebraic reasoning with the processes can be used to verify the design and to provide a mechanism for control at runtime to maintain compliance and security.

We can separate out some basic cases:
- the user can modify the data items as the content of the document representing the static aspect of the user interface. The set of data items can be represented by the simple list or as a structure formalized by regular language.
- the user can modify the structure of document resulting in a more complex processing at conceptual model level that may lead to structural alteration within the model

**WIS Semi-Structured Document Viewpoint**

Most of Web-based applications are document centric. Most of business processes in an organization are carried out through workflows making use of documents as information source and transmitter. Moreover, the Internet technology accelerates business processes to be implemented by Web-based applications. From the viewpoint of human-computer interaction, Web-based document actualizes the structured and semi-structured data through the computer-user interface of the WIS, and specifies the dialogue between users (clients) and the WIS. A central part of WIS is the Web pages and Web semi-structured data in the form of XML/HTML documents for the users on the network (Intranet, Extranet, and Internet). Presentations of information to the users and requests for input from the users are the two facets of Web sites from the viewpoint of the WIS. With the documents implemented in the extensible markup language (XML), the relationship and information exchange between the Web pages and the associated business processes and knowledge representation are realized.

Formally, the document type hierarchy can be defined recursively:
1. The intensional document type $I_D_T$ is the generic document type and itself generates a document type hierarchy $I_D_T(Doc_Hierarchy)$.
2. If $D$ is a generic document type representing an intensional document. Let $D1(Doc_Hierarchy1)$ ... $Dk(Doc_Hierarchyk)$. If $Di$ is-a $D$ for all $1 \leq i \leq k$ then a new document type hierarchy can be created as $D(D1(Doc_Hierarchy1)$ ... $Dk(Doc_Hierarchyk))$.

To use the process algebra for reasoning about WIS, there is a straightforward mapping:
- The intensional document type mapped to a scenario.
- If $D \in I_D_T(Doc_Hierarchy)$ then the associated tasks are mapped to a scene, the actions of tasks are mapped to processes.
- If $D \in I_D_T(Doc_Hierarchy)$ then the pre-conditions formulated as first order logic statement for a task can be linked to processes with pre-conditions so called guarded processes. The post-conditions of tasks are related to so-called post-guarded processes.
DISCUSSION

The goal of this paper has been to outline and investigate a formal modeling framework for WIS that is focused on documents and the processes that manipulate them. The documents and the scenarios attached to documents are the entities that can formulate the actor/process objectives. The handling of documents, the impact on documents by processes and the final effects on the underlying data structure can be described formally by Process Algebra. As the WIS provides information systems services that are used by business services consequently, the documents are modified through business tasks grounded in business rules. As it is well-known, the underlying database scheme that reflects the business’s or enterprise’s data structure changes very infrequently. The basic arrangement of business services evolves slowly. However, the documents mirror everyday fluctuations of businesses or the outer environment of WIS. The proposed model consists of scenarios formulated by Process algebra and strongly coupled documents. Using the process algebra formalism, the business services, the information services, the related business rules can be depicted in a unified framework. This approach can handle the control, the data and event viewpoint of information system at the same time.

The contributions of this paper are as follows:

1. Combining the story algebra based on general process algebra and the document centric approach.
2. Formal description of the impact of documents and the related events on WIS
3. Using the architecture approach, a WIS architecture outlined that can be used to perceive the effects realized through documents on single components of WIS.

CONCLUSION

The proposed approach yields solutions for formal handling of the raised issue. The security and control requirements can be formulated against a document-centric WIS in the form of constraints. The important business related information is contained in documents; the processing requirements and the set of rules are captured in the scenarios of Story Algebra. The proposed method separates the data aspects from the process flow, but it also provides a unified framework to treat the document, the data, and the process involving the business rules and finally the event aspects together. The Story Algebra, or more generally the Process Algebra, provides a descriptive tool that helps capturing changes caused by a document and/or event at the outer interface level. The formal model of documents provides the opportunity for fixing precisely the relationship between the document and data changes as a result of events and the processing requirements at all levels of architecture.

REFERENCES


FORMALNI PRISTUP VEB INFORMACIONIM SISTEMIMA ZASNOVAN NA “ALGEBRIPRIČE”

Rezime:

Ključne reči: veb informacioni sistem, proces orijentisan na dokumenta, modelovanje podatka, algebra priče, algebra procesa.

Received: 05.09.2012.
Correction: 07.10.2012.
Accepted: 15.10.2012.