Modelling the Information Structure of the Document Based on Specialised Markup Language

Tatiana Penkova[*]
Institute of Computational modelling SB RAS, Krasnoyarsk, Russia
Siberian Federal University, Krasnoyarsk, Russia
[*]Corresponding author

***Abstract*—The paper presents an approach to modelling the document information structure to provide the operational document formation based on database information. The method of document structuring by highlighting the specific elements of variable information and applying a specialised markup language is proposed here. The algorithm for analysing the document information structure and constructing the adaptive model for visual setting of generation parameters is considered. The examples of method implementation are presented in this paper in case of bidding for government and municipal procurement.**

Keywords-modeling the document information structure; variable information elements; specialised markup language; adaptive model; data exchange

I. Introduction

Cross-system interaction of data is one of the most essential tasks in sphere of corporate information system development. Currently the data exchange regardless of the format used by interacting systems is becoming a prerequisite for business process automation [1, 2]. Documentation support aimed at the operational formation of various reporting, organisational and administrative documents is one of the most important aspects of any company's activity. The problem of document generation based on database information can be considered as data exchange in Relational Data Base (RDB) and Microsoft Word Document (DOC) formats [3].

One of the spheres where data exchange in RBD and DOC formats is especially asked for is bidding for government and municipal procurement [4]. The modules of Regional automated system of municipal bidding, engaged in information exchange using information of data warehouse, generate a huge number of various documents (e.g. notifications, protocols, plans, contracts etc.) that are compiled in the course of procurement in accordance with laws and regulations concerning order placement.

Complexity and variability of the document structures require special models and methods to provide automated modification and generation of documents depending on the current needs, on the one hand, and, on the other hand, development of modern office tools for document preparation. At present, there are a lot of different document generation tools and systems that take into account various categories of users and the level of technical equipment. Nevertheless, the analysis of existing solutions demonstrates that the most part of tools offer the programming way of forming the document structure and setting the generation parameters [5]. It means that any corrections in document form or in database structure require reprogramming. Another part of programmes in form of specialised applications with visual tools has an extremely complex interface for users and requires knowledge of database theory and SQL [6].

Consequently, the advancement in modern technologies, advantages of the existing approaches, and restrictions on existing solutions, all together substantiate the relevance of originally developed techniques and tools for operational document generation based on stored data within interacted informational systems. They provide a unified approach to modelling the document information structure and visual setting of document generation parameters with a possibility of flexible adaptation to changes in the systems' functioning conditions.

II. Method of Document Structuring

For modelling the information structure of the document, there has been proposed a document structuring method based on highlighting the structural elements by applying the specialised markup language. The novelty of this method is in the ability to implement a unified approach to presenting the information structure of the documents and to provide the visual setting of document generation parameters.

The document information structure contains two types of information: static and variable. The static information is unchanged in process of document generation while the variable information is dynamically formed based on the results of database queries. The variable information elements in document structure are identified by applying an explicit markup. Markup is specified in form of opening and closing tags using the special symbols. The main function of the markup language is to describe the document logical structure. In general, the document information structure can include one or more variable information elements with different functions. The content of variable information elements is located between the opening and closing tags. In process of document generation, the content of the elements is replaced by a value from database according to the query results.

Taking into consideration a lot of various documents that are formed within the framework of cross-system interaction in municipal procurement area, the types of variable information elements have been developed as follows:

- Variable
- Table Variable
- Cyclic Block
- Optional Block
- Sub-Document

In order to produce the specialised markup language and to describe the document information structure, the author employs the Backus Normal Form (BNF) as a syntactic notation [7]. Also, to improve perception and understanding of syntactic descriptions, the syntactic rules are presented here in the form of syntactic diagrams [8]. The movement over the graph defines a syntactically correct language construction.

In accordance with BNF-notation, the document information structure can be presented as the following construction:

$$< Document > ::= < Body\ of\ information\ structure >$$

$$< Body\ of\ information\ structure > ::= \{< Static\ information >\} < Variable\ information >$$

$$< Variable\ information > ::= \{< Variable >\}\{< Table\ variable >\}\{< Cyclic\ block >\}\{Optional\ block\}\{< Sub\_document >\}$$

Each element of the variable information can be considered in the following terms:

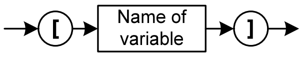
- *Function* – is a description of element purpose
- *Format* – is a BNF-definition and syntactic diagram of the element
- *Comment* – is an additional syntactic description that is not expressed in BNF-definition or an explanation of the element implementation.

A. Variable

Function: adding a unit of variable information in the document.

Format:

$$< Variable > ::= [< Name\ of\ variable >]$$



Comment: name of variable is indicated in square brackets. It presents a keyword that reflects the meaning of automatically added information and identifies the variable in the visual model of document information structure. The opening tag – is an opening square bracket; the closing tag – is a closing square bracket.

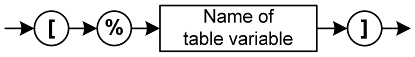
B. Table Variable

Function: constructing and filling a data table.

Format:

$$< Table > ::= \{< Table\ variable >\}\{< Static\ information >\}$$

$$< Table\ variable > ::= [\% < Name\ of\ table\ variable >]$$



Comment: table variable is located in the table. The name of table variable is indicated in square brackets with the percent sign (%) after the opening bracket. It presents a keyword that reflects the meaning of automatically added information into the table and identifies the table and its variables in the visual model of document information structure. The opening tag – is an opening square bracket with the percent sign (%); the closing tag – is a closing square bracket. The table variable is established in row of table when the table should be filling by adding the rows (Figure 1) and the table variable is established in column of table when the table should be filling by adding the columns (Figure 2). The number of variables in the table is unlimited. In addition, the table can contain the static information or other structural elements.

Heading 1	Heading 2	...	Heading n
[% Name of table variable 1] [% Name of table variable 2]	[% Name of table variable 3]	...	[% Name of table variable n]

FIGURE I. EXAMPLE OF TABLE VARIABLES FOR ADDING THE ROWS

Heading 1	[% Name of table variable 1] [% Name of table variable 2]
Heading 2	[% Name of table variable 3]
...	...
Heading n	[% Name of table variable n]

FIGURE II. EXAMPLE OF TABLE VARIABLES FOR ADDING THE COLUMNS

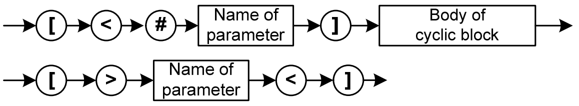
C. Cyclic Block

Function: multiple repeating the document fragment according to parameter.

Format:

$$< Cyclic\ block > ::= [< \# < Name\ of\ parameter >] < Body\ of\ cyclic\ block > [>< Name\ of\ parameter ><]$$

$$< Body\ of\ cyclic\ block > ::= < Body\ of\ document >$$



Comment: body of cyclic block is a document fragment that is repeated according to a given parameter. The opening tag – is a name of parameter that is placed in square brackets with left angle bracket and the pound sign (#) after the opening bracket. The closing tag – is a name of parameter that is placed in square brackets with the right angle bracket after the opening bracket and the left angle bracket after the closing bracket. The name of parameter is a name of variable by which the body of cyclic block is output cyclically (e.g. territory, procurement, supplier etc). The name of parameter identifies the cycle in visual model of document information structure. The body of cyclic block can contain the static information or variable information elements of any type.

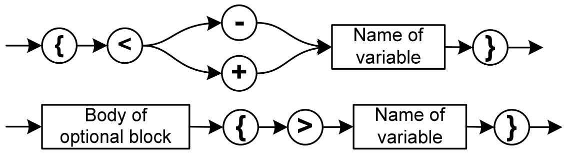
D. Optional Block

Function: visual displaying of the document fragment depending on variable or table.

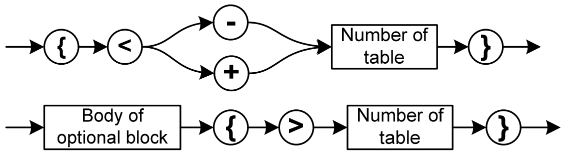
Format:

$< Optional\ block >::= (< Optional\ block\ by\ variable >$
$| < Optional\ block\ by\ table >)$

$< Optional\ block\ by\ variable >::=$
$\{< (+|-) < Name\ of\ variable >\}$
$< Body\ of\ optional\ block > \{>< Name\ of\ variable >\}$



$< Optional\ block\ by\ table >::=$
$\{< (+|-) < Number\ of\ table >\}$
$< Body\ of\ optional\ block > \{>< Number\ of\ table >\}$



$< Body\ of\ optional\ block >::=< Body\ of document >$

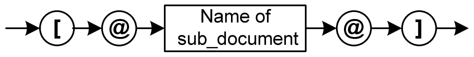
Comment: body of optional block – is a document fragment that can be either displayed or not in document depending on values of variable or table variable. The opening tag – is a name of variable or a number of table that is placed in curly brackets with left angle bracket and plus (+) or minus (-) symbols after the opening bracket. The closing tag – is a name of variable or a number of table that is placed in curly brackets with right angle bracket after the opening bracket. The plus symbol (+) in the opening tag means that the document fragment should be displayed when the values of variable or table are presented. The minus symbol (-) in the opening tag means that the document fragment should not be displayed when the values of variable or table are not presented.

E. Sub-Document

Function: adding the document fragment prepared as a separate independent document.

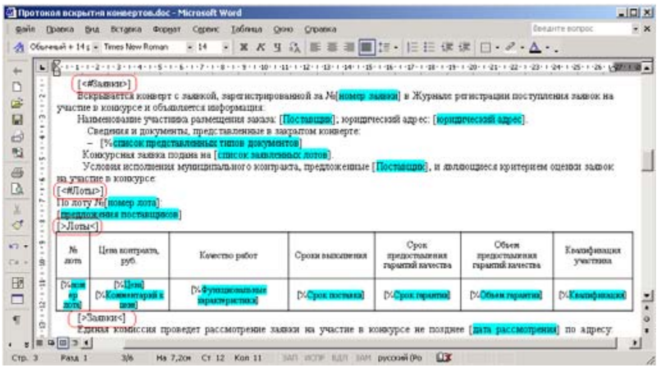
Format:

$< Sub\_document >::= [@ < Name\ of\ sub\_document > @]$



Comment: name of sub-document is indicated in square brackets with the at sign (@) after the opening bracket and before closing bracket. Name of sub-document – is a keyword that reflects the meaning of automatically added information from another created document and identifies the sub-document in visual model of document information structure. The opening tag – is an opening square bracket with the at sign; the closing tag – is the at sign with closing square bracket. Application of this type element allows to process the document fragments that are repeated in several documents (e.g. headers and footers of the protocols, the list of persons approving or signing the document etc.).

The name of variables, the name of table variables and the name of parameters that are used in markup language are specified in terms of domain. Figure 3 demonstrates the examples of document – “Protocol of procurement results” with specialised makeup language. In this figure the names of variables and the names of table variables are highlighted in colour, the opening and closing tags of optional blocks are circled by line.


Протокол вскрытия конвертов.doc - Microsoft Word
Файл Правка Вид Вставка Формат Сервис Таблица Окно Справка
Введите вопрос
Обычный + 14 Times New Roman 14
Стр. 3 Разд 1 3/6 На 7,2см Ст 12 Кол 11 ЗАП ИСПР ВДЛ ЗАМ русский (Ро

[<#Заявки>]

Вскрывается конверт с заявкой, зарегистрированной за №[номер заявки] в Журнале регистрации поступления заявок на участие в конкурсе и объявляется информация:

Наименование участника размещения заказа: [Поставщик]; юридический адрес: [юридический адрес].

Сведения и документы, представленные в закрытом конверте:

– [%список представленных типов документов]

Конкурсная заявка подана на [список заявленных лотов].

Условия исполнения муниципального контракта, предложенные [Поставщик], и являющиеся критерием оценки заявок на участие в конкурсе:

[<#Лоты>]

По лоту №[номер лота]:

[предложения поставщиков]

[>Лоты<]

№ лота	Цена контракта, руб.	Качество работ	Сроки выполнения	Срок предоставления гарантий качества	Объем предоставления гарантий качества	Квалификация участника
[%номер лота]	[%Цена] [%Комментарий к цене]	[%Функциональные характеристики]	[%Срок поставки]	[%Срок гарантии]	[%Объем гарантии]	[%Квалификация]

[>Заявки<]

Единая комиссия проведет рассмотрение заявки на участие в конкурсе не позднее [дата рассмотрения] по адресу

FIGURE III. THE FRAGMENT OF DOCUMENT WITH SPECIALIZED MAKEUP LANGUAGE

The comprehensive application of variable information elements, including the cyclic and optional blocks allows us to create documents with different structures and implement the complex logic of document generation based on data stored in relational databases.

III. Algorithm for Analysing the Document Information Structure

In order to provide the setting of document generation parameters, it has to provide a visual model of the document information structure in form of hierarchy of structural elements. The visual model should reflect the document information structure, specified by the specialised makeup language, and give a setting tool to connect the structural elements of the document and tables of relational database with a possibility of flexible adaptation to changes in the systems' functioning conditions. Figure 4 presents the algorithm for analysing the document information structure and contracting the adaptive model.

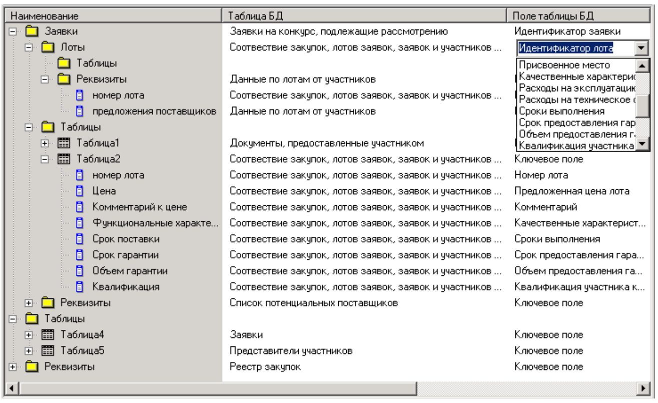
```
Algorithm DocAnalysis()
    input: D
    output: V, SD, TV, C
  begin
    Search_Variable_SubDocument (tag)
      if tag is VariableTag do
        Get_Variable (V)
      if tag is SubDocumenTag do
        Get_SubDocument (SD)
      else
      enif
    Identify_Table(tab)
      for i ← 1 to N do
        Search_TableVariable (tag)
          if tag is TableVariableTag do
            Get_TableVariable (TV)
          else
          enif
      endfor
    Search_CyclicBlock (tag)
      if tag is CyclicBlockTag do
        Get_CyclicBlock (C)
      else
      enif
  end
```

FIGURE IV. THE ALGORITHM FOR ANALYSING THE DOCUMENT INFORMATION STRUCTURE

The analysis of the document information structure is based on sequential identification of the variable information elements by parsing the text according to specialised tags. The algorithm implements two principal tasks: the first task is to identify the main structural elements that characterise the document; the second task is to identify the structural elements inserted in the cyclic block that characterise the separate fragment of the document.

At the first stage, the algorithm searches the variables and sub-documents by analysing the tags (*Search_Variable_SubDocument (tag)*). At the second stage, the algorithm identifies all tables in the document and searches the table variables (*Search_TableVariable (tag)*). Then, the algorithm searches the cyclic blocks (*Search_CyclicBlocks (tag)*) and saves the body of cyclic blocks with all inserted elements into the temporary documents for further processing. As a result, four types of variable information elements are defined: variables (*V*), sub-documents (*SD*), table variables (*TV*) and cyclic blocks (*C*). At the last stage, the algorithm forms the hierarchy of structural elements where the set of variables is combined into a section named Requisites, the set of sub-documents is combined into a section named Sub-documents and the set of tables is combined into a section named Tables. In the case of cyclic blocks, the algorithm is started again to process the temporary documents in the same way. The structural elements, identified in the body of cyclic blocks form the lower hierarchy for each cycle.

Thus, the hierarchical model reflects the document information structure and allows us to set the field and database table for each element. Figure 5 presents the visual model of the document information structure. This informational model gives the flexible setting tool to manage the document generation and data exchanging.



Наименование	Таблица БД	Поле таблицы БД
Заявки	Заявки на конкурс, подлежащие рассмотрению	Идентификатор заявки
Лоты	Соотвествие закупок, лотов заявок, заявок и участников ...	Идентификатор лота
Таблицы		
Реквизиты	Данные по лотам от участников	
номер лота	Соотвествие закупок, лотов заявок, заявок и участников ...	
предложения поставщиков	Данные по лотам от участников	
Таблицы		
Таблица1	Документы, предоставленные участником	
Таблица2	Соотвествие закупок, лотов заявок, заявок и участников ...	Ключевое поле
номер лота	Соотвествие закупок, лотов заявок, заявок и участников ...	Номер лота
Цена	Соотвествие закупок, лотов заявок, заявок и участников ...	Предложенная цена лота
Комментарий к цене	Соотвествие закупок, лотов заявок, заявок и участников ...	Комментарий
Функциональные характе...	Соотвествие закупок, лотов заявок, заявок и участников ...	Качественные характерист...
Срок поставки	Соотвествие закупок, лотов заявок, заявок и участников ...	Сроки выполнения
Срок гарантии	Соотвествие закупок, лотов заявок, заявок и участников ...	Срок предоставления гара...
Объем гарантии	Соотвествие закупок, лотов заявок, заявок и участников ...	Объем предоставления га...
Квалификация	Соотвествие закупок, лотов заявок, заявок и участников ...	Квалификация участника к...
Реквизиты	Список потенциальных поставщиков	Ключевое поле
Таблицы		
Таблица4	Заявки	Ключевое поле
Таблица5	Представители участников	Ключевое поле
Реквизиты	Реестр закупок	Ключевое поле

FIGURE V. THE VISUAL MODEL OF THE DOCUMENT INFORMATION STRUCTURE

IV. CONCLUSION

The article presents an approach to modelling the document information structure to provide the operational document creation based on database information. The author proposes the method of document structuring based on highlighting the specific elements of variable information and applying a specialised markup language. This method allows us to form the documents with different structures and implement the complex generation logic. The paper gives the algorithm for analysing the document information structure and constructing the adaptive model in form of hierarchy of structural elements. This informational model gives a flexible setting tool to manage document generation with a possibility of adaptation to changes in the systems' functioning conditions.

ACKNOWLEDGMENT

The reported study was funded by Russian Foundation for Basic Research, Government of Krasnoyarsk Territory, Krasnoyarsk Region Science and Technology Support Fund to the research project No 16-47-243020.

REFERENCES

[1] G. Hohpe, B. Wolf, “Enterprise integration patterns: designing, building, and deploying messaging solutions”, Addison-Wesley Professional: Computers, 2004, 683 p.

[2] D. Braue, “The Integration Imperative”, 2002, // http://www.cio.com.

[3] T.Penkova, A. Korobko, A. Belorusov “Development of a unified model of data representation for cross-system interaction”, Advances in Intelligent Systems Research, vol.133, 2016, pp. 48-51.

[4] "Contract system in the procurement of goods, works and services for state and municipal needs": The Federal Law N 44-FZ, 05.04.2013 (ed. from 03.07.2016).

[5] A.Soloviev “An overview of the report generation tools. Concepts and tools”, Document management. Concepts and tools, 2004, pp. 174-172.

[6] K. McKeown, K. Kukich, J. Shaw “Practical issues in automatic documentation generation”, Proc. ANLC '94, pp.7-14.

[7] R.N. Moll, M.A. Arbib, A.J. Kfoury “An introduction to formal language theory”, Springer Science & Business Media, 2012, 203 p.

[8] S.C. Reghizzi, L. Breveglieri, A. Morzenti “Formal languages and compilation”, Texts in Computer Science, Sc.Ed.: Springer, 2009, 400.