Overdraft Agreement Processing Based on XML and its Implementation in Bank Information System

Igor O. Marin, Sergey V. Orekhov

National Technical University "Kharkov Polytechnic Institute"
Frunze str. 21, Ukraine, 61002, Kharkov
osv@kpi.kharkov.ua

Abstract: In this paper we present a brief overview of Bank Information System (BIS) market situation in Ukraine. An architectural approach for BIS is proposed. The approach demonstrates how to build the BIS application for credit department of a bank. The paper presents a first step of such system implementation, namely: subsystem for overdraft agreement processing. The subsystem uses XML to describe agreement statements. XML content is processed using XSLT to implement credit commission servicing.

Keywords: Enterprise Information System, Bank Information System, XML, XSLT.

1. Introduction

In recent years, substantial changes have been affecting the economy of Ukraine, giving rise to new trends in information systems development and application. More and more organizations perceive that an information system serving as the artificial nervous system for them is essential for successful management, marketing and resource planning. The most progressive organizations in this direction are financial institutions and banks, which need such systems to provide effective and high-quality services and to deal with huge and continuously growing amounts of data.

Due to steady tightening of law restrictions and prohibitions on use of non-licensed software in Ukraine, banks will change over from self-developed modules of information systems implemented by their own IT departments using non-licensed tools and DBMS to standardized and web-oriented information systems, provided by specialist IT companies. Nevertheless this will lead to impoverishment of the bank services market. So, developers of bank information systems should provide highly flexible interfaces, allowing to develop different add-ons for those IS parts, accountable for providing services and document treatment.

In fact, good idea for a company (BIS developer) is to have an IS skeleton template, liable for legally regulated and slowly changing part of the banks' activities and individually customizable and easily upgradeable modules for other segments of banks' market activities. In this case, it would be cheap and less time-consuming operation for such

company to "tune" their IS for a bank of almost any scope of operations. ERP systems advanced further in this direction and can serve as example of information systems, tunable for wide range of enterprise structures and functions.

2. Proposed solution

The concept, proposed by our project team is to define a stable backbone of information system, providing common functions specified by banking license (which is in turn defined by banking law, and thus is changed very slowly). This part of the system includes data warehouse and transactional system. Actually, for Ukrainian banking systems, there are regulations, defined by National Bank of Ukraine, concerning their interface with NBU, account card, report systems and interaccount items for various operations [Law].

The regulations do not specify implementation of the systems, so we propose to orient the implementations at the legally defined interface. The particular legislative base changes constantly, especially when interaccount items concerned, so we define flexibility points in the system and use distributed architecture, isolating employees interface from legally defined business operations.

The other factor, designating use of flexibility (or connection) points is marketing and constant ambition to provide new services to customers. We faced this factor while developing agreement-processing system for credit department of the bank. The requirement, set before our group, consisted in providing presentation of the documents, allowing usage of conditions, not known at the moment of system design.

The conceptual scheme of proposed system architecture can be seen in the Figure 1. As shown, internal data warehouses are linked and interoperate using flexibility points. Such decision is conditioned by the fact that these warehouses can be implemented using various technologies and by different vendors, what usually does not provide compatibility. Target servers also interact with other parts of the system via flexibility points. The physical deployment of information system, functioning now in the bank, is presented in Figure 2.
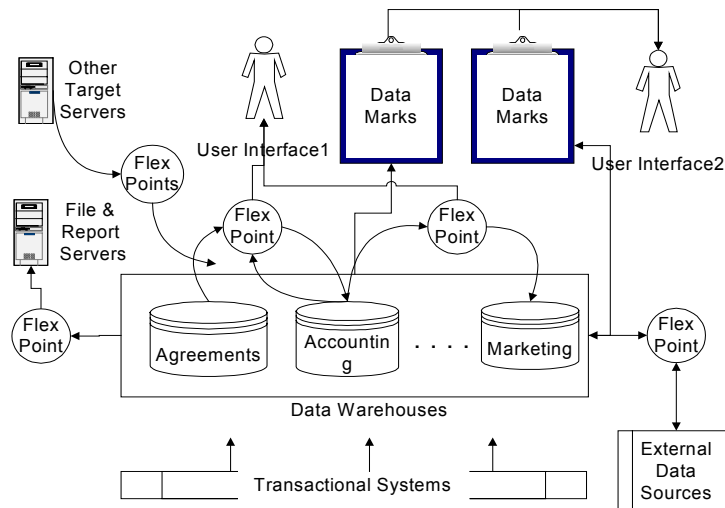
Figure 1. Architecture of proposed IT solution

Target servers are shown in the upper row: 1) Domain controller – maintains domain naming and access rights 2) Backup domain server is a reserve server and stores backup information 3-5) Western Union, Reuters and Liga-Online servers are vendor supplied servers, using which is an essential clause in agreements with services suppliers 6) Cardmaster server is responsible for processing plastic cards transactions 7) File server – stores essential files, common documents etc. 8) Data server – server with installed DBMS storing accounting, transactions DB etc. 9) Proxy Server serves for external internet activities 10) Local Web server encapsulates connection points between hetero-geneous system clusters and provides other services 11) NBU server performs all com-munications with National Bank of Ukraine, is also responsible for reporting.

Using Web server to deploy connection (flexibility) points allows easy distribution among bank affiliates, connected to the main office.
Having this physical scheme, one of the basic parts of proposed architecture for over-draft agreements processing was developed.

2.1 Overdraft agreements processing subsystem

Within the framework of our research project, task to develop credit department agree-ments storage and processing subsystem as a part of business management system was formulated. At the first stage we have developed a pluggable add-on to the existing in-formation system of the Factorial Bank (Kharkov, Ukraine), which stores overdraft agreements. The basic functionality of the developed module is credit data processing and daily commission computation depending on credit agreement conditions in force. We believe that adaptation of the subsystem to process other credit and deposit agree-ments with monthly commission and interest rates computation will not take much ef-

forts from the developers. This statement is based on the fact that employees of any bank try to develop some kind of uniform agreement template (or several templates) for all credit and deposit agreements to simplify their work with processing and commission computation. The information system of the bank consists of several objective oriented modules, distributed among servers (see in Figures 1-2).

Proposed subsystem divides into two parts:
1) Agreement storage and processing part
2) Commission Computation part

The add-on is, in fact, a flexibility point between agreements database, transactions log warehouse and account card. The working principle of the add-on can be seen in Figure 3. Agreements are stored in XML format along with their changes history and processed by the first part of the module to elicit actual agreements terms and conditions. "Accounting" element represents accounting warehouse, from which account balances histories are retrieved by commission computation module to compute daily charges. The charges information is then reported to credit department employees and, if approved, is enforced in the accounting warehouse [S01].

We defined a document specification for overdraft agreement, which includes, so called, growth points. This means that when
marketing department develops new commission rates conditions or new criteria for assessing credit rates, these conditions or criteria can be dynamically added to the operational information. The rules for computing percent rates should be specified in formal way to allow automated processing.
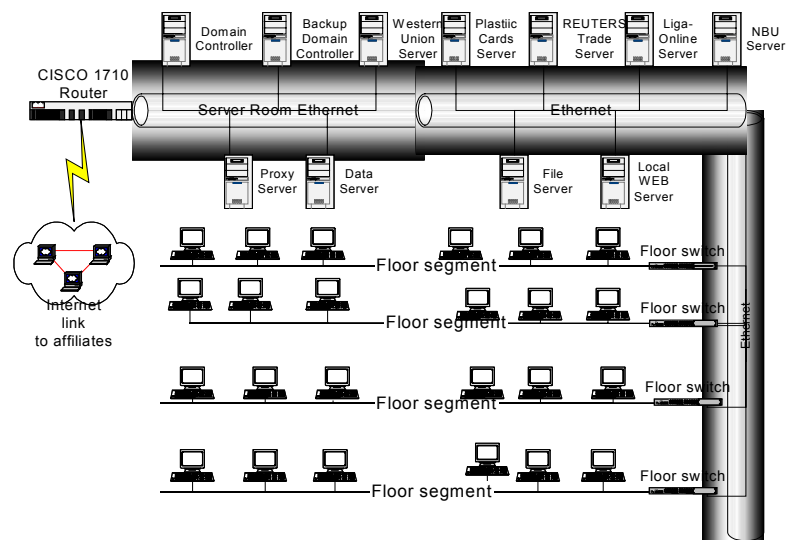

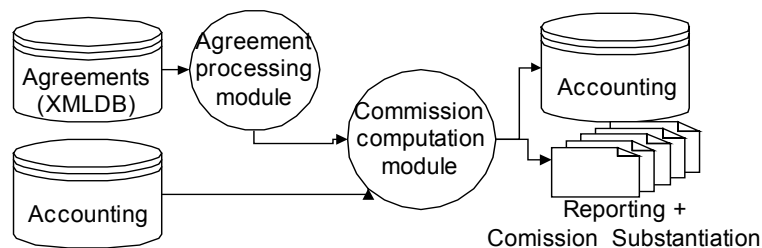Figure 2. Physical implementation of proposed IT solution

Figure 3. Functionality of BIS component for overdraft agreements processing

2.2 Agreement storage and processing

Our BIS component analyzes the credit agreements as structured documents with legal template, defining common agreement conditions and business part, defining commission rates and changing rules for the latter. For today, the changing rules, occurring in contracts with bank customers can be of 2 formally different types:

a) Rates changes after some days passed with debit balance more than particular amount and within particular credit limit, defined by the agreement. The debit balance is registered on the overdraft account of the contractor. This means that contractor is using definite amount of credit resources of the bank and should be charged corresponding commission.

b) Rates changes after some defined date if the agreement is not prolonged. This is a computation rule for arrears debt. If contractor did not stop using overdraft loan after agreement duration had expired. This rule can also be used if the bank expects rise or slide of money demand on the market to increase or reduce normal commission rate for using credit resources. For this purpose, there is a parameter indicating date from, which new rate should be applied.

Because the agreement contains structured text, it should be naturally to use markup inside the agreement text. So we decided to use XML as markup language for agreement texts and to define extensible document type for the agreement.
The structure of the agreement document is the following:

- **Overdtaft** {
(*)Contract
        { Contractor, (+)Limit_Group
                {(*)General_Condition
/*This is a sample point of growth. New conditions can be derived from General conditions*/
                        {(?)Rate_Condition
                                {NR_Day_Count
/*New percent rate results from uninterrupted usage of credit resources below the limit set*/
                                        | NR_Date
/*New percent rate results from passing particular date*/

```
                                    } } },
        (*)Addit_Agrmt
                {…
        /*here we define how conditions can be changed in additional agreements to
the main contract*/
                }
        } }
```

Attributes and text nodes are omitted in the above description.

The points of growth are General_Condition and Rate_Condition elements, within which respectively new common conditions of the agreement and new conditions influencing commission rate can be set.

Limit_Group element represents grouped conditions for particular overdraft limit. That is we can have different rates and dates policy for loans below UAH 200K, below UAH 500K, below 1M etc.

We also developed the possibility to store the agreement along with history of changes of credit conditions. Because in real life changes are legally fixed using additional or supplementary agreements to the main contract, we designed the formal specification of such supplements so that any condition of the main agreement or any supplement, preceding (by the date) the given one can be changed or removed. New conditions or limit groups of conditions can be added, changed or deleted.

Addit_Agrmt element in the DTD represents particular additional agreement, containing changes in contract conditions. We treat changing conditions as removing old one and setting new one, following C.J. Date paradigm in database operations (UPDATE = DELETE + INSERT). The Addit_Agrmt element structure is following:

```
    Addit_Agrmt
        {(*)Add_lim_g
                {Limit_Group {…} },
                (*)Del_lim_g, (*),
                (*)Add_cond {General_Condition{…}},
                (*)Del_cond
        }
```

We define each supplement as a set of conditions or limit groups of conditions to be added or deleted. For this purpose we index main contract conditions and limit groups with IDs. Inside Addit_Agrmt element we use these IDs to manipulate conditions. Each supplement is also characterized with date when it goes into effect.

Such organization allows building agreement conditions for given date. The process of automated daily commission charge begins with elicitation of actual agreement conditions in effect. The elicitation is performed using a set of XSLT rules, which can be parameterized for particular contract, day or contractor. [X01]

2.3 Commission computation module

Actual conditions in effect for a particular day are conformant to the same document type as contracts without supplementary agreements. Conditions in effect and daily balance of the overdraft accounts serve as input for commission computation procedure, also implemented as transformation module of XSLT rules. There are two input text streams, corresponding balance statistics and conditions, and one output stream, the latter represents commission rates and amounts intended for charge.

Account balance statistics is formed by stored procedure in the accounting database and conforms to the following document type description:

Daily_stat {(+)Day [@DDate]{(+)Acc_data [@Number, @Saldo]}}

The output of the module contains daily commission, rate and reasons for charging the commission. Hereinafter there is the sample output, given by the program code – see Listing 1.

```
<Percents>
 <Day DDate="2003-01-08">
   <Acc_data Number="2600xxxxxxxxx1" Saldo="1000"
        Perc_rate="20.5" Commission="0,56">
        <norm_rate rate="18" />
        <ad_perc  reason="day count" param="2"
           min_saldo="150">2</ad_perc>
        <ad_perc  reason="date" param="2003-01-05"
           min_saldo="150">0.5</ad_perc>
   </Acc_data>
   <Acc_data Number="2600xxxxxxxxx2" Saldo="2000"
        Perc_rate="20" Commission="1.1">
        <norm_rate  rate="18" />
        <ad_perc  reason="day count" param="2"
           min_saldo="0">2</ad_perc>
   </Acc_data>
 </Day>
 … … … …
</Percents>
```

Listing 1. Output, given by the subsystem

There is information present for 2 accounts for one day 8 January 2003. The evaluated percent rate is 20.5% and charged commission is 0.56 UAH. Inside the Acc_data tag there is percent rate substantiation. It is stated that there is normal commission rate of 18% and two additional charges. These are 2% for two days of resources usage and 0.5 percent commission advance agreed in contract and enforced from 5 January 2003. The additional charges are valid for balances more than 150 UAH.

Both developed components are deployed at the Web-server, allowing access from distributed users. The access is performed by middleware components, run in the application server container.

2.4 Requirements and testing

The developed subsystem is deployed at the Apache Web Server. It requires Java-compatible XML parser and XSLT transformer. The XSL transformer, actually used is Saxon 6.5.2 based on the Xerces parser. The transformation is performed on the Athlon 2100+ processor, 256 MB RAM server. At the bank, where the subsystem is deployed, there are 35 overdraft credit agreements opened for today. If the subsystem is adopted to process conventional credit agreements, it would have 300-400 credit agreements to process with an average of approximately 5 conditions and 3-4 additional agreements, specifying conditions modifications. Timings for the subsystem under different OS and JVMs are presented below:

OS	JVM	AgrmtCount	AvgTime
Win2kProf	MSJVM	32	9.563
Win2kProf	JDK1.3	32	3.827
WinXPProf	JDK1.3	300	29.805
WinXPProf	JDK1.3	400	39.901

We can see that the best performance is achieved using JDK 1.3 VM rather than Microsoft VM. This factor is reported as the Saxon transformer feature. We can also make a conclusion that the system is worth adopting for conventional credit agreements processing.

3. Conclusion, future and related work

3.1 Architectural approach description

We intend to design the loan department business management subsystem in the way that connection points will be deployed in the middleware servers or on the Web server. Such design reduces work when integrating new business processes and new objective oriented tools from third party vendors.

Deploying flexibility points at the Web server allows fast and easy update for document types and processing rules. The template-oriented technology allows easy and fast development and integration of new conditions, devised by marketing department.

Now people from IT department perform this update, but the work is underway, concerning integration of XSLT editor, which allows automated rules writing following the user. This will give loan department staff the opportunity to directly integrate new loan and commission schemes. We aim to implement other parts of the system in the same direction and propose to research the opportunity of developing abstract environments for programming business rules.

3.2 Related and future work

We have found partially related work in different aspects of our project, but no fully cognate research was found by our research group.

Our data representation concept of describing commercial documents as structured pieces of information independent from the representation is not new. We found a set of Document Type Definitions, named cXML (for Commercial XML, **http://www.cxml.org**), which describe uniform structure of bills, invoices, purchase orders and basic features of contracts. The cXML, however, is mainly intended to be a communication protocol (request-response), rather than to be a comprehensive data storage structure.

In the area of distributed system implementation, there are commercially available products of similar banking information implementation, for example SmartVista processing solution from Banking Production Center (**http://www.smartvista.biz/pdf/bookSV.pdf**).

Cognate XML-based document processing research was found in [VRL00], related XSLT composition ideas are expressed in [ES01].

To describe our future work, we can state that we aim at building distributed credit department business management subsystem, which can be plugged into the enterprise system with help of XML and Web-based connection points. Next steps in our work will be extending agreement-processing module functionality to process all credit department agreements. We also propose to adopt the agreement processing system to process constantly changing regulations, alterations and amendments to which are sent almost daily from National Bank of Ukraine.

Strategic goal is to develop interface conception for enterprise system modules. This lets enterprise system be highly customizable and simplifies integration, modification and scaling.

References

[S01]   Spenser Pall. XML. Design and Implementation. Moscow: Lori, 2001. – 509 p.

[X01]   World Wide Web Consortium (W3C) XSLT Specification
            (http://www.w3.org/TR/xslt20/ http://www.w3.org/TR/xslt)

[VRL00] L. Villard, C. Roisin and N. Layada, "A XML-based multimedia document processing model for content adaptation", Digital Documents and Electronic Publishing (DDEP00), LNCS, 2000.

[ES01] J.Eder, W.Strametz;"Composition of XML-transformations"; technical report ISYS-01-01, University of Klagenfurt, 2001.

[Ba94] M. U. Babichev et al "Banking", Moscow: Economics, 1994

[Law]   Banking Regulations Digest (www.bank.gov.ua)