BUSINESS PROCESSES MODELLING AND AUTOMATION IN THE BANKING SECTOR: A CASE STUDY

MARA NIKOLAIDOU[†], DIMOSTHENIS ANAGNOSTOPOULOS[*]

APHRODITE TSALGATIDOU[†]

[†]*Department of Informatics, University of Athens*
Panepistimiopolis 15784, Athens, Greece
tel: +(301)-7275614
fax: +(301)7275214

[*] *Harokopion University of Athens*
El. Venizelou Str, 17671 Athens, Greece
tel: +(301)-9549171
fax: +(301)-9514579

email: {mara}{dimosthe}{afrodite}@di.uoa.gr

Abstract: The banking sector is a competitive environment, where business process re-engineering is constantly needed. Business process modelling and automation are effective tools towards this direction, improving the performance of business activities and enabling enterprise-wide monitoring and coordination. In this paper, we present a case study of modelling and automating business processes in the Loan Monitoring Department of a medium-sized Bank. Loan monitoring is a typical banking activity, which includes business processes concerning loan approval, collection of delinquent loans and initiation of appropriate legal claims. These processes are often performed in cooperation with external business partners, such as legal firms and brokers, have collaborative properties and are considered to be of dynamic nature. Their efficiency strongly depends on human operator experience and subjective criteria. The loan monitoring policy employed is a significant factor for determining profits. Thus, relevant business processes should always be monitored, evaluated and refined. Business process modelling was conducted using the Modified Petri-Net (MPN) model, which allows the description of ad-hoc and collaborative business activities. Business process automation was performed using Lotus Domino/Notes groupware platform, since widely known workflow management systems do not provide the means for the description of such activities. The direct mapping and support of MPN main entities within Notes environment ensured the accurate and complete implementation of all business processes and reduced significantly programming cost. Loan Management System is the integrated environment build to support loan monitoring activities. Our experience and the potential of the business process modelling and automation approach are also presented.

Keywords: business process modelling and automation, petri-nets, banking applications, workflow support

1. INTRODUCTION

Integrating and managing complex organisations and their information systems requires understanding, partitioning and simplification of their complexity. Business modelling supports these requirements by providing means for describing process-oriented systems and decomposing them into manageable parts. Business processes (BPs) are collections of activities with a common objective, such as fulfilling a business contract or satisfying a customer need. Business process definition (i.e. a description of a BP at a high conceptual level necessary for process understanding, evaluation and redesign) requires a well-defined *model*, that provides a set of concepts appropriate to describe BPs [Zelm et al, 1995]. The model should be rich enough and enable process validation (e.g. by simulation or static analysis) to decide whether the process definition accurately represents the system under study. BP modelling is a significant tool for re-engineering organisational procedures and it is usually followed by BP automation, aiming at improving business process performance and enabling organisation-wide monitoring and coordination. An automated BP is referred to as a *workflow*, while a *Workflow Management System (WMS)* is software used for its coordination and control [Mohan et al, 2000]. WMSs also provide a set of interfaces to users and applications involved in the workflow progress. For efficient workflow development, one should start with defining and understanding business processes (BP modelling), before specifying and implementing the corresponding workflow applications (BP automation). Provision of generic and flexible modelling methods is thus required both at BP modelling and BP automation levels. Several methods have been suggested for BP modelling, most of which are based on textual programmable languages or graphical notations, such as dataflow diagrams, state transition diagrams, Petri-Nets and related notations. Combination of different BP modelling methods has also been examined to give new, enhanced approaches [Abeysinghe and Phalp, 1996].

Four types of business processes are usually studied in the literature [Alonso and Mohan, 1997]. These are: production, administrative, ad-hoc and collaborative. Administrative and production BPs refer to bureaucratic procedures that include well-defined steps and are controlled by a set of well-known rules. Such processes can be easily described by conventional modelling tools and are usually automated using a WMS [Hollingsworth, 1995]. Ad-hoc processes are similar to administrative processes, except for the fact that they deal with unique or loosely defined conditions, which are not easily modelled, and can not be efficiently supported by current WMSs. Collaborative processes are characterised by the number of participants involved and the synchronisation needed, and are handled more effectively using groupware technology.

In this paper, we present our experience on business process modelling and automation of the Loan Monitoring Department (LMD) of a medium sized, private Bank. Loan Monitoring includes business processes concerning loan approval, loan collection and loan case assignment, which are considered as collaborative and dynamic. Business process modelling was conducted using the Modified Petri-Net (MPN) model [Tsalgatidou et al, 1996], which is an extension of Coloured Petri-Nets and allows the description of dynamic activities. To automate business processes modelled with MPN, we decided to extend the capabilities of Lotus Domino/Notes groupware product [Lotus Co, 2000]. The MPN modelling environment and the WFS developed for LMD, named Loan Management System (LMS), use the same repositories maintained within Lotus Domino platform, enabling the direct mapping of all entities defined during BP modelling within the WFS.

The rest of the paper is organised as follows: Section 2 provides an overview of loan monitoring processes and their specific characteristics. Section 3 describes the MPN modelling approach and its integration with the workflow support environment. Section 4 provides an analytical description of the MPN model for the delinquent loan collection process. In section 5, we discuss process automation using Lotus Domino platform. Our experiences concerning the system deployment over the last two years and its impact on LMD operation are presented in Section 6. Conclusions reside in Section 7.

2. BUSINESS PROCESS DESCRIPTION

Bank of Athens is a medium sized, private bank. The assignment of corporate loans to healthy, small companies, with relatively loose terms was one of its most profiTable activities. The bank decided to offer other loan types as well and, after three years, it became clear that, as business grew, profits deviated significantly from the expected ones.

LMD is responsible for the approval and monitoring of loans. The main objectives of the department include approving loans, collecting delinquent loan instalments and initiating legal claims against non-credible customers. LMD is in constant collaboration with the Credit Office, all local branches of the Bank as well as external business partners (e.g. legal firms). Department employees handle three main loan types: corporate, housing and consuming. Monitoring and approval of each loan type is performed according to a different policy and data maintained for each category differ.

LMD is divided into two sections: The first for handling consuming and housing loans and the second for handling corporate loans. Each section has its own Manager and consists of groups, formed by no more than ten employees including Group Leaders. All employees, including Managers, handle loan cases, while under specific circumstances a consuming or housing loan case may be also assigned to an employee working in the corporate loan section.

Although the account management system produced detailed daily reports for loan monitoring, there was no information system support for LMD. Due to the enormous amount of data, classification and processing was far from efficient, resulting in poor coordination and low productivity. Furthermore, data of dynamic nature concerning the history of a delinquent loan case, e.g. delayed payments and payment enforcement actions, was not maintained. Both the re-engineering and WMS support of the LMD were thus approved.

The first step towards automating LMD operation was the identification and complete specification of the business processes supported. After carefully reviewing the organisational scheme and interviewing employees, we concluded that three main business processes are supported:

- Loan Approval
- Delinquent Loan Collection
- Loan Case Assignment (for both approval and collection)

Loan approval process is initiated by the Credit Office at local branches, which serve as the entry point for any loan approval request. The approval or rejection of the request must be announced within a period of 24 hours for consuming loans and 3 to 7 days for housing and corporate loans. Without information system support, the approval of consuming loans was performed by the LMD employee at the specific branch, based on locally available data. The requests for housing and corporate loans and the relevant certificates were photocopied and forwarded to LMD. While LMD is responsible for loan approval, contract establishment and customer account updates are performed by the Credit Office.

Delinquent loan collection is the most important activity of LMD, initiated whenever the payment of a loan instalment is delayed. In practice, this was not feasible due to the large number of delinquent loans, especially consuming ones. LMD had no mechanism to detect delinquent loans and assign them to a collector in a daily basis. In addition, loan assignment could not be performed dynamically according to evolving criteria, concerning either loan groups or a specific loan case. Moreover, section and group managers were not in position to modify the criteria for delinquent loan case assignment to the collectors supervised by them.

The requirement to assign the loan to a collector that had already handled it in the past was also not satisfied, as delinquent loan case history was only kept in paper files. If the collector was an external partner, all files had to be copied and sent via mail. This was one of the reasons external partners handled only a small portion of permanently delinquent loans.

In spite of the three loan categories, we concluded that loan handling was performed according to a set of common rules, which are then customised on the basis of predefined parameters, such as loan category, amount owed, etc. However, the employee approving or monitoring the delinquent loan case makes decisions according to subjective criteria. Lack of knowledge of the loan case history and customer information affected the collector's judgement.

Loan monitoring activities described above can be characterised as collaborative processes, since many participants are involved in the completion of a single step. The delinquent loan collection procedure is an ad-hoc BP, since the steps to be accomplished are not well established and exceptions may occur. Furthermore, each loan case may be reassigned to another collector at any time and collection policies may be changed while handling a case. Thus, collection activities evolve in a dynamic way [Casati et al, 1998].

3. BUSINESS PROCESS MODELLING

Numerous modelling methodologies, such as IDEF0 [Marca and McGowan, 1993] and RADs [Ould, 1992], provide the means to understand the behaviour of static systems [Starke, 1994]. Production and administrative activities usually fall in this category. BP modelling is also used to tackle the problem of changing or evolving systems [Phalp, 1998]. Ad-hoc processes can be viewed as an evolving system. The business model used to depict such systems should be flexible enough to facilitate the accurate description of business activities. BP modelling approaches based on extending Petri-Net functionality [Murata, 1989] can provide better solutions for this kind of problem [Oberweis, 1996], [Tsalgatidou et al, 1996], since they focus on depicting the relationship between activities and resources rather than the relationship between activities. BP models based on Petri-Nets can be simulated using discrete event simulation [Rajala and Savolainen, 1996].

Advantages of simulation-based validation of business processes have been well recognised in [Nidumolu et al, 1998] and [Hlupic and Robinson, 1998]. Some of them are: modelling of cyclic and stochastic behaviour and complex rule-based interactions between activities and resources, visualisation of process steps, development of accurate models through comparing them with mathematical data and enhanced confidence in predictions regarding performance impacts of changes to the process [Gladwin, 1994]. Although the benefits of simulation are not yet widely exploited by the business engineering community, there is a great potential offered, as indicated in [Hlupic and Robinson, 1998], especially when dealing with dynamic BPs.

As described in the previous section, loan monitoring processes are both ad-hoc and collaborative, thus of dynamic nature. Figure 1 depicts the steps accomplished for implementing a workflow system to support LMD operation. After gathering and evaluating information concerning the department operation and policies, a BP model was constructed, evaluated and then mapped into the workflow environment.

The BP model must fulfil the following requirements:

- Enable the accurate description of ad-hoc processes
- Facilitate the evaluation of BPs through simulation
- Support the direct mapping of entities into the workflow environment to minimise implementation cost.

The modelling formalism adopted is the Multi-level Modified Petri-Net (MPN) [Tsalgatidou et al, 1996], which is an extension of Coloured Petri-Nets

[Jensen, 1992]. The formal and executable nature of MPN models enables the employment of simulation techniques for validation purposes.

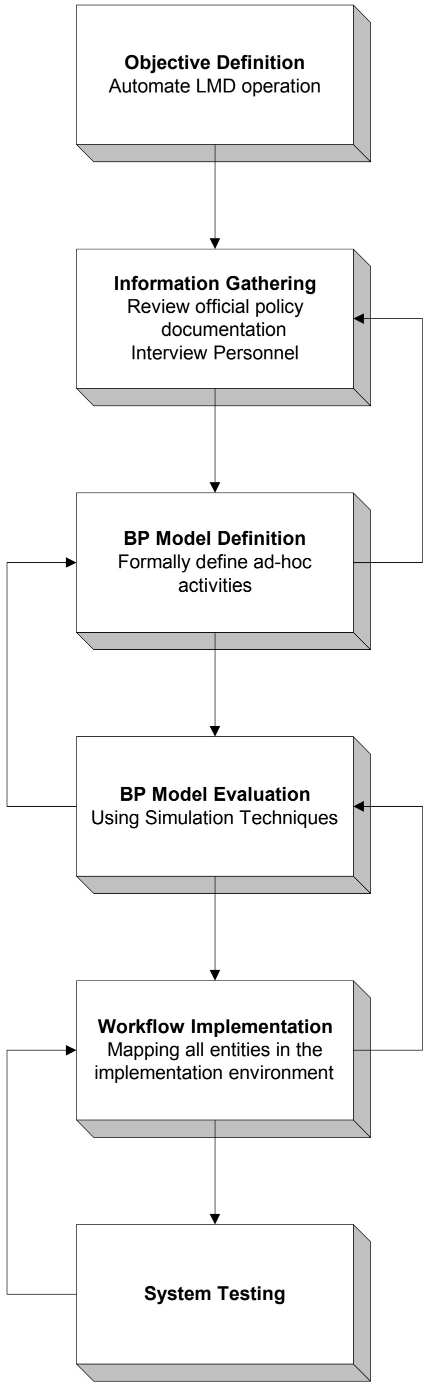

Figure 1: BP Modelling and Automation Steps

A Petri-Net consists of *places* and *transitions* between them. *Arcs* are used to denote relations between places and transitions. A transition is performed whenever all its input places are filled with *tokens*. When a transition is completed, output places are filled with tokens. Transitions depict processes and their components as activities and tasks, respectively, while places represent resources [Jensen, 1992]. An overview of other modelling approaches based on Petri-Nets and their comparison with MPN is included in [Tsalgatidou, 1996].

MPN is used for modelling BPs at various levels of abstraction. Transition decomposition depicts the decomposition of a BP to its activities, sub-activities and tasks and demonstrates the control and data flows between the different organisational units involved in the BP.

MPN also facilitates the description of an organisational model. Places in the Petri-Net can be inscribed with organisational entities, such as actors and roles, making the integration among organisation models and process models smooth and tightly coupled. Since places can also be inscribed with resource and control objects, a desirable integration between control flow, data flow and the organisation model is attained. All entities inscribed in Petri-Net places are stored as objects within the MPN Repository.

MPN facilitates the representation of ad-hoc BPs, as each activity is not connected with others, i.e. it does not follow nor is followed by another activity, as in IDEF0, and there is no activity ordering. Activities can be initiated whenever all input places are occupied by the appropriate token, i.e. whenever the necessary resources and participants are available. Upon completion, each activity provides tokens to its output places, i.e. releases the resources needed for the activation of another activity. Thus, both static and dynamic processes are described uniformly. It also provides a clear, visual representation of the activity steps executed with the collaboration of many actors, facilitating the description of cooperative processes.

Although MPN model can be simulated using discrete event simulation, this feature was not implemented within this project due to time constrains. The main objective of the project was to have an operational version of the system as soon as possible, before the full scale re-engineering of LMD operation. A discrete event simulator for the evaluation of BPs described according to the MPN model is currently under development.

In the case of LMD, FlowMark WMS [IBM, 1999] and Lotus Domino/Notes groupware product [Lotus Co., 2000] were examined for BP automation, as the Bank already had a contract with IBM for information system support. FlowMark is a high-end product supporting modelling of well-defined business processes, usually representing administrative procedures characterised by transactional features [Reinwald and Mohan, 1996].

FlowMark is not flexible enough to efficiently describe collaborative or ad-hoc process functionality. Lotus Domino/Notes platform provides the basic infrastructure to coordinate, manage and monitor activities and share data in a distributed environment. It also facilitates constraint definition, provides a secure communication mechanism and expands easily. While less demanding than FlowMark, it provides an integrated object-based database management mechanism and simple programming tools.

To support the direct mapping of BP modelling entities within the workflow implementation environment, we decided to integrate MPN within Notes/Domino platform. The repository needed to support the MPN model is implemented as a Notes database and it is used during both BP modelling and workflow operation. Workflow definition is performed using the MPN model, as MPN constructs are directly mapped into Notes workflow entities. In this way, workflow implementation cost is minimised, since the developer only adds code segments in the preconstructed Notes structures.

The MPN model and the repository architecture are presented in the following paragraphs. Workflow implementation concepts are described in section 5.

3.1. Modified Petri-Net Model

A BP model should encapsulate information related to: (a) *activities*, (b) *resources* assigned to activities, i.e. objects necessary for the execution of activities, such as actors, documents, data, etc, (c) *control* of a BP which describes 'when' and 'which' activity is executed, (d) the *flow* of data in the process and (e) the *organisational structure* which consists of organisational units, people, roles, competence, etc [Tsalgatidou & Junginger, 1995]. These entities must be therefore mapped within MPN. The formal definition of MPN model is given in [Tsalgatidou et al, 1996]. In this paper, we discuss MPN structure and the functionality of its components.

The MPN modelling approach is based on the following principles:

The overall *MPN model* represents a specific BP and consists of different *SubMPNs* depicting the decomposition of the BP into a more detailed level.

Activities, that may either be simple tasks or further decomposed, are modelled as transitions. If the activity is further decomposed, a lower level MPN is used for its description. A *script* is related with each activity and represents the set of steps to be carried out during its execution. Scripts can be described using a high level language and are particularly useful in the case of simple tasks that are not further decomposed.

Control information, *resources* and *performers* required for the execution of activities are modelled as objects inscribing the MPN places. Control objects are either signals (representing messages among activities) or events (representing occurrences of incidents) and enable the representation of control flows within the process. Resource objects are data objects used by the process, such as invoices, and enable the representation of data flows and data modification within the process limits. Resource objects are maintained in MPN *Repository*. Each resource is identified by properties and can be either simple or complex. The organisational structure supported is also maintained in the Repository in the hierarchical form presented in Figure 2.

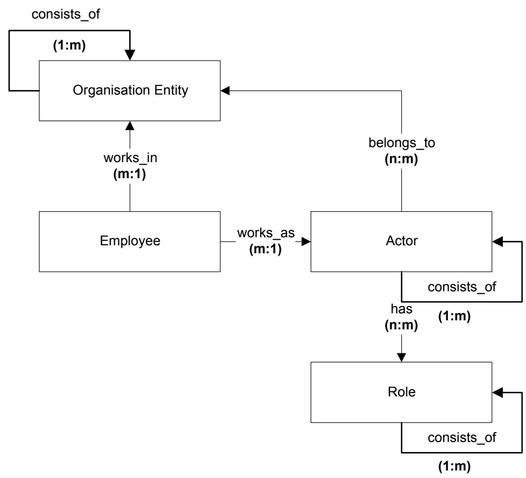

Figure 2: Organisation Structure Model

Actors represent a position profile, e.g. manager or programmer, assigned to a specific employee within the organisation. More than one employee may be associated with one Actor. Roles group sets of duties and responsibilities assigned to a specific actor. Roles can be described in terms of other roles. Roles, Actors or Employees are required at the input places of a given transition, where the presence of specific operators (performers) is essential for its enactment.

MPN constructs are presented in Figure 3. Red places indicate performers, blue ones indicate resources and green ones indicate control. Control places represent control flow. A control place has more than one outputs, while the decision concerning which output will be activated may depend on deterministic or random rules (e.g. employee decision). Control places are particularly useful when describing semi-constructed or ad-hoc BPs. Places filled with colour indicate the existence of tokens. Grey rectangles indicate activities further decomposed by a SubMPN model. In this case, performers and input/output resources can be redefined, but not altered in a more detailed level of description.

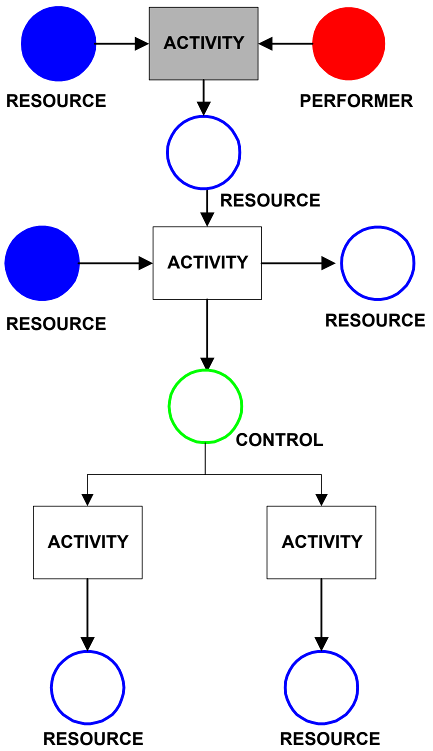

Figure 3: MPN Model Representation

3.2. Integrating MPN within Notes Platform

According to Domino/Notes architecture, all entities defined within Notes platform are stored as *Notes Objects*. Different kinds of object classes can be defined using *Notes Forms. Notes Templates* are used to implement a basic functionality. Based on these templates, it is possible to construct *Notes Databases* supporting the characteristics of specific applications. Both templates and databases are constructed using Notes programming tools.

In order to construct MPN models, a graphical interface was developed using Java. The GUI module, named MPN editor, communicates with *Repository* database stored within Domino Server. The same database is used to maintain activity description and the activity decomposition scheme. The Repository Database is based upon *Control Template* (CT) that is used for the definition of main MPN entities. The authors suggest the following mapping of MPN entities within Lotus Notes, as depicted in Table 1.

MPN Entity	Notes Representation
Employee	Notes User and Employee form
Actor	Actor form
Role	Role form
Organisational entity	Organisational entity form
Resource	Resource form
BP definition (new MPN model)	BP form
Activity definition	Activity form
Task definition	Supported Tasks form

Table 1: Mapping of MPN entities within Notes Platform

The definition of the organisational structure is performed through the Organisation Entity form, where the user defines the name of organisational entities, their level and the supervising entity. Actors are represented through the corresponding form. Constraints, such as the number of employees acting as an Actor, can also be defined. For example, each department can only have one manager. Roles are represented by the Role form. All the roles an actor may obtain appear in the corresponding field within the Actor form. The template provided to support this overall functionality is presented in Figure 4.

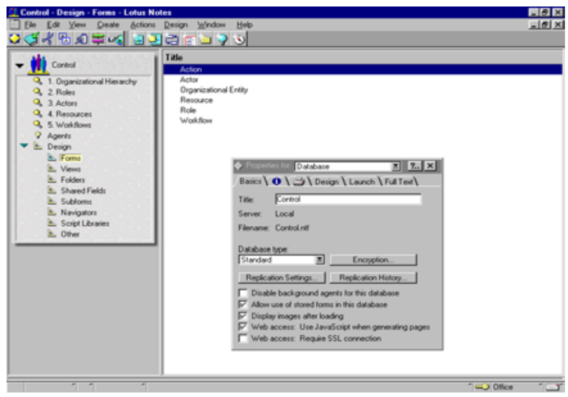

Figure 4: Control Template View

Resources are also defined using the Resource form and are maintained in the Resource Notes Database. The corresponding form and view are automatically created for each resource defined. Forms are left black so that the developer can customise them during workflow development.

Finally, the definition of BPs, BP activities and tasks (simple activities) is respectively supported using BP, Activity and Supported Tasks forms. Only the Supported Tasks form is used during workflow implementation. As an example of the structures created within CT, the formal description of the object class corresponding to “Activity” form is presented in Table 2. Activity_Script is the code executed upon activation.

```
Class Activity
      {
      Name:                 String;
      Parent_Activity:      String;
      SubActivities:        Activity[];
      Activity_Script:      Script[];
      Guards:               Script[];
      Input:                String[];
      Output:               String[];
      Performer:            String[];
      }
```

Table 2: Class Definition within Notes

Guards contain the conditions for triggering the activity and are realised as an array of script sentences that calculates to a Boolean value. Input/Output contain input and output places, respectively.

The overall architecture supporting BP modelling and automation is depicted in Figure 5.

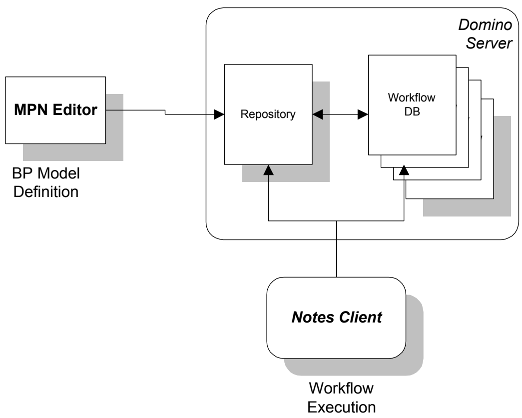

Figure 5: BP Modelling and Automation Environment

4. LOAN MONITORING PROCESS MODELLING

We discuss the *Delinquent Loan Collection* process, since it combines both ad-hoc and collaborative properties. Delinquent loan collection may be performed by any LMD employee. As Actors, we have defined all LMD employee positions: *Department Manager, Section Manager, Group Leader* and *Collector*. *Collector* role describes the task of collecting delinquent loans. Other roles are *Administrator, Case Assignor, Supervisor* etc. Collector is a simple role obtained by all employees, while Case Assignor or Supervisor roles are only obtained by specific managers.

New delinquent loan cases and loan payments are downloaded from the account management system on a daily basis. When a loan case is assigned to a specific collector, it appears in his/her daily diary. Performing loan collection actions results in the modification of the loan status. Reminder data for the loan case are also maintained. There is a predefined set of actions initiated by collectors, some of which are:

- Client contact without success
- Client contact with success/ no payment scheduled
- Client contact with success/ payment scheduled
- Sending a letter to the client
- Defining an auction
- Defining a new payment settlement

The loan case is handled by LMD until delinquent loan instalments are paid. The generic MPN model depicting delinquent loan collection is depicted in Figure 6.

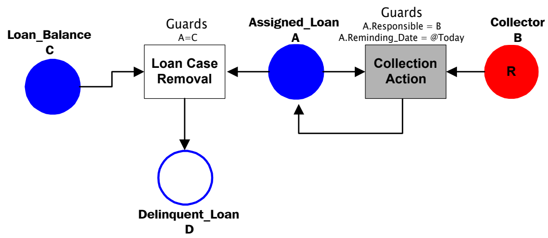

Figure 6: Delinquent Loan Collection Model (level 0)

Loan collection process consists of two main activities: *Collection Action* and *Loan Case Removal*. The first one is further analysed by another MNP, while the second one is a simple task. *Loan Case Removal* has no performer place as input and it is performed automatically. For *Collection Action* activity, which has such an input, the symbol "R" within the place indicates that a role has been defined as the performer, more specifically the *Collector* Role. Guards specify transition constraints. The *Collection Action* activity may be initiated when the collector responsible for the specific loan case is available (A.Responsible=B) and a reminder is set in his/her daily diary (A.Reminding_Date=@Today). To simplify guard definition, places are numbered using capital letters or numbers.

At the more detailed level, collection actions are distinguished into three categories: simple actions completed by the collector, actions requiring the approval of his/her supervisor and actions for which sending a letter is necessary. Determining the proper action is based partially on LMD policy and partially on the collector's strategy and experience. The sequence of actions is thus not predetermined and this is depicted in the MPN model using a control place, as indicated in Figure 7.

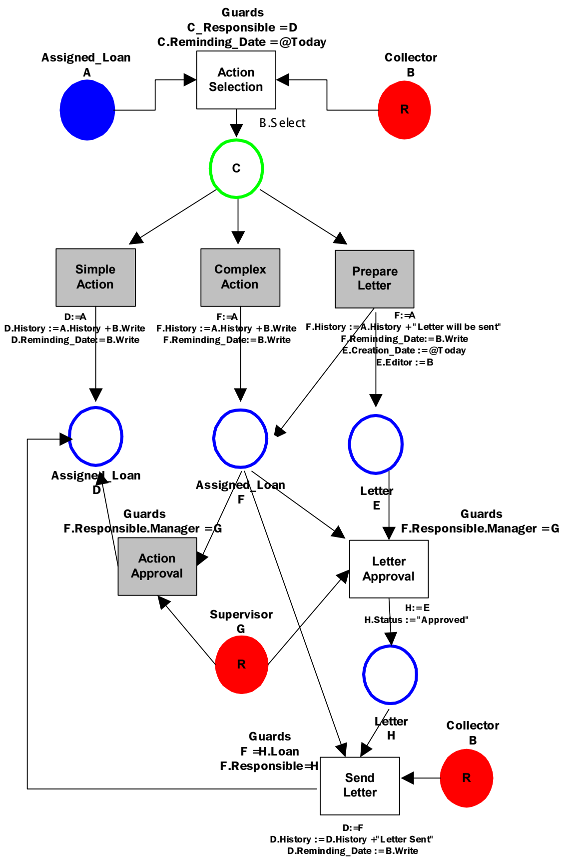

Figure 7: Collection Action Model (level 1)

The in-depth description of the *Collection Action* activity is beyond the purpose of this paper. However, one should note some points of interest. Input and output places of the MPN must be the same as the input and output places of *Collection Action* activity presented in Figure 6. Although other resources, such as *Letter*, are produced during this activity, they are not indicated as output. In the MPN model presented in Figure 7, the *Supervisor* role is also depicted as a performer. This is allowed, as the *Supervisor* role also includes *Collector* role. As indicated in *Action Approval* activity, guards are not static and enable the representation of BPs with dynamic evolution. Although it is not mandatory, scripts can be defined to describe activity functionality (scripts are depicted under activities). Scripts are useful especially when describing tasks, as they are automatically transformed into code within the workflow environment. An example concerning the *Send Letter* task is presented in the next section.

Using the MPN model presented in Figure 7, one can simulate the execution of the business processes and reach conclusions concerning the behaviour exhibited by specific collectors, as well as the effectiveness of various strategies and the indication of potential inefficiencies and bottlenecks.

5. BUSINESS PROCESS AUTOMATION

Workflow execution (e.g. BP automation) is performed using Lotus Domino/Notes platform. Loan Monitoring System (LMS) supports the operation of LMD by enabling the description, enactment and monitoring of workflows corresponding to the supported BPs. LMS provides flexibility during business process description, allowing the modification of a workflow while it is running, and is adapted according to the scenarios defined by authorised actors. It also provides tools for monitoring workflow evolution and actor productivity. Although the system is oriented towards bank specifications, it is not proprietary and can be customised to support any organisation with similar structure and requirements.

Workflow description is based upon MPN BP models. Repository database, build upon Control Template (CT) is used to obtain entity definition, while workflow databases corresponding to different BPs are constructed using the *Workflow Template* (WT). Mapping MPN main entities within CT is crucial to ensure the complete description of activities. The developer only fills the code segments needed to describe activity functionality, as similar structures are provided and the conditional invocation of activities is pre-endured. If direct mapping were not feasible, process description would have to be also performed at the implementation level, without ensuring that consistency is maintained throughout the modelling process.

Regarding WMS architecture, the CT is used for the maintenance of the organisational structure and resource description [Bussler, 1999], while WT is used for the construction, execution and management of the actual workflows. It is controlled by parameters defined in CT. Each workflow corresponding to a BP is characterised by its state [Kappel et al, 1995]. Depending on the state status, specific tasks, modelled as supported *Notes Actions*, can be initiated.

Using the BP form in CT, a new workflow and a set of valid states are defined. The actions corresponding to simple activities, the initial state of the workflow for the activation of each action and the resulting state are recorded in the Supported Tasks form. Actions can be performed automatically (system participant) or manually (human participant). Action types are also defined in the Supported Task form. Authorised actors are allowed to add new states and modify the relationship between states and actions, resulting in modifying the workflow evolution.

5.1. Workflow Template

The Workflow Template includes a basic workflow form, called Main Workflow form, depicting the progress of a business process. This form includes *State* field to depict workflow state, which is handled by the system. For each task, the Workflow template automatically creates the proper Notes Agents in the database. The developer must add the necessary code for the agents using the programming tools provided by Notes. These include a simple proprietary language, which is an object-oriented extension of Basic and Java, called *LotusScript*. Agents correspond to manual or automated activities. In the first case, the corresponding button for agent activation is automatically added in Main Workflow form. The button is visible only when the Workflow State permits the activation of the corresponding action. In the second case, the agent is properly scheduled for execution by Lotus Notes Client or Domino Server. The developer has to extend the functionality of the Main Workflow form to support specific application needs. Workflow Template also includes a basic Notes view, called Main View, providing access to all active workflow instances created using Main Workflow form. Table 3 summarises the mapping of MPN entities within WT.

The script corresponding to *Send Letter* task and the corresponding agent automatically extracted from the script description are presented in Table 4. The agent code is not fully execuTable, as the developer is responsible for adding the desired functionality.

MPN Entity	Notes Representation
BP implementation	Main Workflow form
Task script	Agents
Actor's Work List	Main view

Table 3: Mapping of MPN Entities within WT

Collection Action Workflow form is depicted in Figure 8. Buttons enable accessing resource information, such as customer personal data, loan history and the instalment payment timeTable. All previous actions concerning this particular case are also accessible. Through the Select Keyword Menu, the Collector selects an action and initiates the corresponding activity.

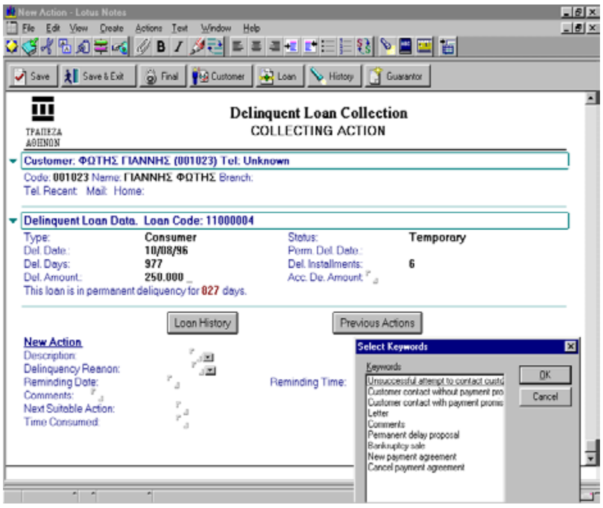

Figure 8: Collection Action Workflow Form

Task Description		**Agent Code**
`Name:`	`Send Letter`	`CreateAgent SendLetter`
`Parent_Activity:`	`Collection Action`	
`SubActivities:`	`NULL`	
`Input:`	`[F Assigned Loan as Loan Resource; H Letter as Letter Resource; B Collector as Performer]`	`{IN AssignedLoan:Loan; IN Letter:Letter;`
`Output:`	`[D Assigned Loan as Loan Resource]`	`OUT AssignedLoan:Loan}`
`Guards:`	`[F =H.Loan; F.Responsible=B]`	

`Activity_Script:`	`[D:=F; D.History := D.History +"Letter Sent"; D.Reminding_Date := B.Write]`	`Begin`
		`AssignedLoan.History:=AssignedLoan.History+ "Letter Sent";`
		`Write(AssignedLoan.Reminding_Date);`
		`End;`

Table 4: Agent Automated Creation

6. EXPERIENCE AND POTENTIAL OFFERED

LMS was operational after a development period of seven months and two months of testing. It enabled the formal description and management of all business processes performed within LMD. Gathering and evaluating information concerning the department operation and policies in a systematic way was one of the major contributions. Business process analysis proved to be time consuming due to the complexity encountered in gathering collector experience. The description of all business processes within MPN and the direct mapping between BP models and workflow implementation contributed significantly to the effectiveness of the overall approach.

Users within LMD have access to the system using a switched 10 Mbps Ethernet network. Branch users are connected to the system via private WAN connections with a rate between 64Kbps and 1Mbps. An overview of LMS is presented in Figure 9.

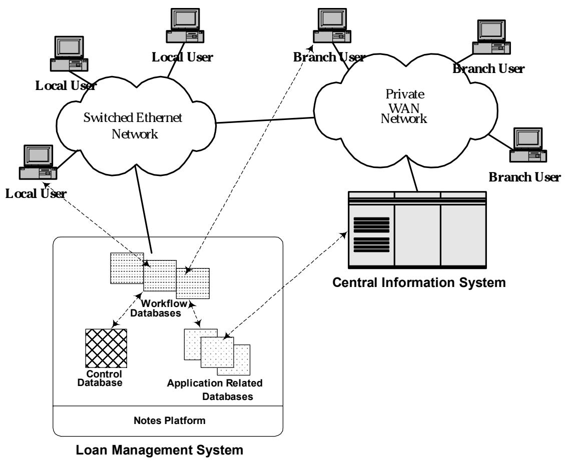

Figure 9: LMS Architecture

LMS was first introduced for managing consuming loans. This was due to the simplicity of consuming loan delinquent cases compared to the corresponding cases of corporate loans. After the successful employment of the system for a test period of 3 months, LMS functionality expanded to handle all loan types. The system is currently used to monitor an average of 10,000 delinquent loan cases. An average of 600 new cases is added on a daily basis and an average of 1,400 transactions per day is executed by all users. LMS supports 15 users in LMD and 64 remote users at branches. An average of 60% of delinquent loan cases concerns one to three delinquent instalments. Less than 30% are permanently treated by the system either due to a special settlement with the customer or the initiation of legal claims against the customer. Most such cases concern corporate loans.

In the second version of the system, client replication was introduced for branch users, especially the ones having a slow connection. The Control Database was fully replicated to eliminate the communication cost for look-ups. Replication of workflow databases was partial, according to the privileges of each user. The Central System was also upgraded to include a second server handling control databases in order to balance load and improve performance. Static data concerning supplementary information, e.g. guarantor data, were moved to an external relational database system (DB2) and were retrieved upon user request. This decision minimised data download time and improved on-line performance, as the cost of Notes database search proved to be more expensive for this type of data than the corresponding cost of an external relational database based on indexed fields through ODBC.

The system was well accepted by users, both collectors and executives. As collectors no more waste time on searching paper files, they are able to monitor more cases and concentrate on improving collection policies, which results in a considerable productivity increase. Collectors working at branches can actively participate in the department operation, as they now have access to the relevant data and may consult their colleagues at LMD. Business success ratio increased by almost 25% over the last two years. One of the main contributions of the system was the active monitoring of all loan cases. Based on the delay period, the system either handles the case itself or asserts a new task to the corresponding collector's daily work list.

Management obtains productivity measurements in real time, while executives are able to monitor the results of slightly changing collection policies, e.g. allowing collectors to establish settlements without further approval, and consequently tune the overall process. This resulted in the progressive re-engineering of LMD. The productivity of this unit increased by 40% in the last quarter of the first year and by another 19% within the second year.

6.1. Virtual Enterprise Support

After a year of full operation, the Bank decided to examine the possibility to assign the collection of permanent delinquent loans to brokers, in order to reduce operational cost. Brokers collect the delinquent instalments on behalf of the Bank and initiate legal claims in cooperation with legal firms. Seven legal firms and ten brokers are currently connected to LMS. Although the Bank is responsible for controlling business processes, external business partners are responsible for their partial execution, as indicated in Figure 10. In this context, business processes are performed within the limits of a virtual enterprise consisting of all cooperating organisations [Alonso, 1999].

Although the Bank maintains its own private network, external business partners are connected to LMS via the Internet, which is a relatively inexpensive solution. The expansion of workflow management over the Internet introduced the following additional requirements, which were appropriately handled:

- Data Security: The Bank handles private data that must be transferred securely. In addition, each partner must have access only to the data needed to accomplish a specific task.

- Performance: System performance must be efficient for all users, although operating in a distributed environment.

Every partner (legal firm or broker) signing a contract with the Bank is obliged to install a Lotus Domino Server communicating with the LMS Server over the Internet. In this External LMS Server, replicas are kept for all LMS databases. For security reasons, administrators of LMS are also responsible for these databases. External Servers are synchronised with LMS Server on a daily basis, using the replication mechanism provided by Notes platform at the field level. The replication protocol uses encryption at both the network and the application level to ensure security during data transfer.

Modifications of the existing application were minimal: Brokers were incorporated as Collectors in the Organisational Structure of the LMD, but no special privileges were assigned to them and no modification of the collection policy was required. Only Loan Case Assignment criteria were modified to include the new collectors.

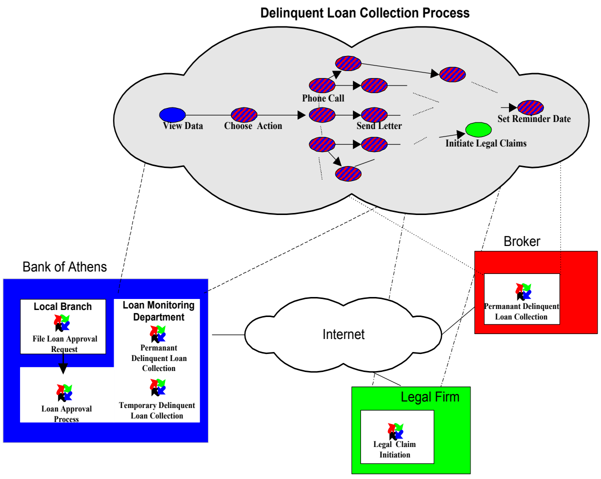

Figure 10: Loan Monitoring over the Internet

7. CONCLUSIONS

Recent research studies for collection activities in the banking sector indicated the lack of efficient information system support. This is due to the complexity and the dynamic nature of business processes. Business process automation provides significant results only if all activities, independent of their nature, are fully described in a formal way. MPN BP modelling method allowed the complete description of dynamic business activities. Direct mapping of MPN main entities within Notes environment ensures the transition of all business activities within the workflow environment and provides the means for constantly monitoring their efficiency.

LMS is an integrated environment supporting the management and execution of dynamically evolving business processes. LMS contributed to the effective re-organisation of the Loan Monitoring Department and facilitated constant monitoring and refinement of business processes and the on-line evaluation of new collection policies.

REFERENCES

Abeysinghe G. and Phalp K. 1997, "Combining Process Modelling Methods". In *Information and Software Technology*, Elsevier Science, Holland, Vol 39, pp 107-124.

Alonso G. and Mohan C. 1997, "Workflow management systems: The next generation of distributed processing tools". In *Advanced Transaction Models and Architectures*, chapter 1, Kluwer Academic Publishers, UK, pp. 35-62.

Alonso G., Fiedler U., Hagen C., Lazcano A., Schuldt H. and Weiler N. 1999, "Wise: Business to Business E-Commerce". In *Proc. RIDE-VE99 9th International Workshop on Research Issues on Data Engineering (*Sydney, Australia, March), IEEE, CA, USA, pp 132-139.

Bussler C. J. 1999, "Enterprise-Wide Workflow Management". In *IEEE Concurrency*, Vol 7, No 3, IEEE, CA, USA, pp 32-43.

Casati F., Ceri S., Pernici B. and Pozzi G. 1998, "Workflow Evolution". In *Data & Knowledge Engineering*, Elsevier Science, Holland, Vol 24, pp 211-238.

Cho Y., Kim J. and Kim S. 1998, "Role-Based to Business Process Simulation Modelling and Analysis". In *Computer and Industrial Engineering*, Pergamon Press, UK, Vol 35, pp 343-346.

Gladwin B. and Tumay K. 1994, “Modelling Business Processes with Simulation Tools”. In *Proc. WSC94 Winter Simulation Conf.*(Lake Buena Vista, FL, December), SCS, San Diego, CA, USA, pp 114-121.

Hlupic V. and Robinson S. 1998, “Business Process Modelling and Analysis Using Discrete Event Simulation”. In *Proc. WSC98 Winter Simulation Conf.* (Washington DC, USA, December) SCS, San Diego, CA, USA, pp 1363-1369.

Hollingsworth D. 1995, “Workflow Management Coalition – The Workflow Reference Model”, DN TC00-1003, WfMC Publications, UK.

IBM Co. 1999, “Functional Assessment of IBM MQSeries FlowMark 3.1”, White Paper

Jensen K. 1992, “Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use”, Volume 1, Basic Concepts - Monographs in Theoretical Computer Science, Springer-Verlag.

Kappel G., Lang P., Rauch-Scott S. and Retschitzegger W. 1995, “Workflow management based on objects, rules and roles”. In *Data Engineering Bulletin*, IEEE, CA, USA, Vol 18 , No 1 , pp 11-18.

Lotus Co. 2000, “Lotus Domino Architecture”, Technical Paper

Marca D. A. and McGowan C. L. 1993, “IDEF0/SADT Business Process and Enterprise Modelling”, Eclectic Solutions Corporation

Mohan C., Barber R, Watts S. et al 2000, “ Evolution of Groupware for Business Applications: A Database Perspective on Lotus Domino/Notes ”. In *Proc. VLDB2000 - 26th International Conference on Very Large Databases* (Cairo, Egypt, September), VLDB Publications, USA, pp 684-687.

Murata T. 1989, “Petri Nets: Properties, Analysis and Applications”. In *Proceedings of the IEEE*, IEEE, CA, USA, Vol 77, No 4, pp 541-580.

Nidumolu S., Menon N. and Zeigler B. 1998, “Object-Oriented Business Process Modelling and Simulation: A Discrete Event System Specification Framework”. In *Simulation Practice and Theory*, Elsevier Science, Holland, Vol 6, pp 533-571.

Oberweis A. 1996, “An Integrated Approach for the Specification of Processes and Related Complex Structured Objects in Business Applications”. In *Decision Support Systems*, Elsevier Science, Holland, Vol 17, pp 31-53.

Ould A. M. 1995, “Business Processes Modelling and Analysis for Re-engineering and Improvement”, John Wiley, UK.

Phalp K. 1998, “The CAP Framework for Business Process Modelling”. In *Information and Software Technology*, Elsevier Science, Holland, Vol 40, pp 731-744.

Rajala M. and T. Savolainen 1996, “A framework for customer oriented business process modelling”. In *Computer Integrated Manufacturing Systems*, Elsevier Science, Holland, Vol. 9, pp 127-135.

Reinwald B. and Mohan C. 1996, “Structured Workflow Management with Lotus Notes Release 4”. In *Proc. 41th IEEE CompCon* (Santa Clara – CA, USA, April), IEEE, CA, USA, pp 451-457.

Starke, G. 1994, “Business Models and their Description”. *In Proc. CON94 Workflow Management - Challenges, Paradigms and Products* (Wien, Austria, March), Springer Verlag, Germany, pp. 134-147.

Tsalgatidou, A. and Junginger, S. 1995, “Modelling in the Re-engineering Process”. In *ACM SIGOIS Bulletin*: Special Issue on Business Process Reengineering, ACM Press, USA, Vol 16, No 1, pp 17-24.

Tsalgatidou A, Louridas P., Fesakis G. and Schizas Th. 1996, “Multilevel Petri Nets for Modelling and Simulating Organizational Dynamic Behaviour”. In *Simulation & Gaming*, Special Issue on Simulation of Information Systems, Sage Publications, Vol 27, No 4, pp 484-506.

Zelm M., Vernadat F. and Kosanke K. 1995, “The CIMOSA Business Modelling Process”. In *Computers in Industry*, Elsevier Science, Holland, Vol 27, pp 123-142.

Zobel R.N. and Lee K.H. 1992, "A Graphical User Interface Based Simulation System for Mixed Application Areas". In *Proc. SCSC 92 Summer Computer Simulation Conf.* (Reno, USA, July) SCSI, San Diego, CA, USA. pp 62-66.