

AMNIS - DESIGN AND IMPLEMENTATION OF AN ADAPTIVE WORKFLOW MANAGEMENT SYSTEM

Bálint Molnár¹, Zsigmond Máriás^{1,3}, Zoltán Suhajda², István Fekete¹

¹Eötvös Loránd University (ELTE) Faculty of Informatics

H-1117 Budapest, Pázmány Péter sétány 1/C

molnarba@inf.elte.hu; zmarias@inf.elte.hu; fekete.istvan@inf.elte.hu;

²Amnis Laboris Kft.

H-1106 Budapest, Gyakorló köz 3.

amnis@verecundus.hu

³LogiNet Kft.

H-1221 Budapest, Vihar u. 5/D

zmarias@loginet.hu

ABSTRACT

The experiences of introduction and operation of ERP systems have revealed that update of these software due to the constantly changing business processes demand huge resources. That is why the demand was formulated for a method that enables introducing new features in software system without any modification in program code according to the evolution of the organization. The objective of Amnis development project is to create a system with this adaptation capability using the basic idea of workflows that create documents during evaluation. In this article design and programming challenges are shown that had to be met during the development of Amnis, focusing on topics of effective data storage and queries, workflow control structures and workflow evaluation techniques.

KEYWORDS

Workflow, ERP, Business Process Management, Business Process Modeling, document-centric.

1. INTRODUCTION: THE CONCEPT OF AMNIS

Information systems recently used by organizations in the government and business sector typically consist of several built-in software modules containing a lot of predefined features. As organizations are permanently facing new challenges, their businesses processes have to change frequently, therefore the modification of these software systems are also often needed. It is well-known fact in the software technology that the development and installation costs of information systems are far exceeded by resources needed for maintenance and modification (about 30:70%).

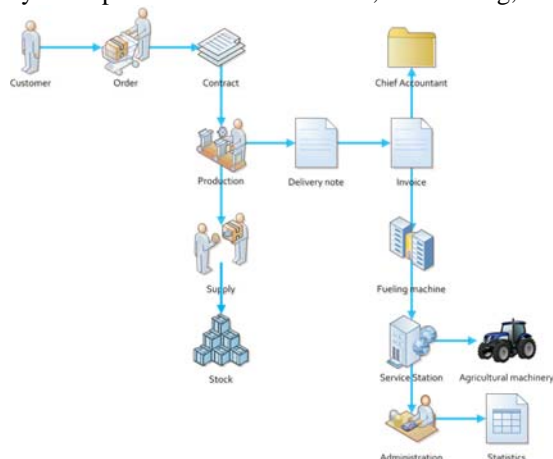
Software update should follow the changes straightway because any lagging can lead to considerable cost of the required human correction activity. In practice, usually the launch of any new activity is often synchronized with the development of the according systems and sometimes new concepts are rejected just because these developments would raise the project costs too high.

Traditional software systems allow parameterization but new parameters will amend the system only to fit to the foreseeable and known demands. In all other cases the software must be modified. A software system that enables introducing new features in operation without any modification in program code would be highly advantageous.

The information system named “Amnis” has this beneficial quality as the system can be flexible adapted according to the changes of the organization’s business processes (Van Der Aalst 2002). *Amnis* is a workflow management software system focusing on business processes as dataflow consisting of a series of documents. As an example, a business process of an enterprise can be seen in Figure 1 using *Amnis*; the “real life” procedures applied by a company can be easily modeled and transformed (Becker 2000) into a computational process. *Amnis* contains no specific business logic, but users can generate potentially any kind of activities by the *workflow components* of the system.

The application of Amnis is not limited to the profit oriented business field; it also can be used in the non-profit and public administration sector. Process orientation is crucial in the e-Government sector, as the

organizations in this field always have to comply with laws and regulations. The goals and motivations are basically different in public administration compared to business area, but the users of the information systems perform similar activities, i.e. creating, manipulating and finalizing documents.



AMNIS v0.031 (m) Demo

Screen 1

Processes

	Id	Process name	# Open	All	Activity
	612.1	EXTERNAL REFUELING RECORDING	0	0	flow.chad (start ExternalRefuelingDataSheet)
	605.1	MONTHLY SUM PER TANK	0	0	flow.chad (start MonthlySummaryTankSheet)
	594.3	IDENTITY CARD MODIFICATION	0	0	flow.chad (start IdentityCardModificationSheet)
	583.2	MONTHLY REFUELING SUMMARY PER VEHICLE	0	0	flow.chad (start VehicleRefuelingSummarySheet)
	574.3	TANK STOCK	0	0	flow.chad (start TankStockSheet)
	568.1	VEHICLE RECORDING	0	1	flow.chad (start VehicleDataSheetRecording)
	567.1	EMPLOYEES RECORDING	0	1	flow.chad (start PersonalDataSheetRecording)
	566.1	IDENTITY CARD RECORDING	0	1	flow.chad (start CardDataSheetRecording)

Using a special kind of graphical editor, a workflow graph can be created. This is not only a definition tool for business processes but it also implements an operational functionality, as the workflow steps will be performed by the *Amnis* interpreter. The workflow graph exhibits the feature of a graphical language that can be interpreted. When the processes vary then any changes can be adaptively pursued only by the modification of process graph; the software code itself does not need to be altered. The changes of the processes in the *Amnis* system are followed without programming, simply using the graphical interface.

The pilot version called *Amnis* 1.0 has been put in operation in 2009. It lacked several important workflow control structures and had performance issues: documents of greater quantity effected an increased time for database operations. Next year, in 2010 a new version of the system was launched after a significant improvement in data storage concept which resulted in enhanced performance efficiency. *Amnis* 2.0 recently runs with success at 184 companies from various fields of business (agriculture, building industry, shipping agencies etc.). The 3rd version of the software has been under development; it will contain more advanced control structures e.g. iterative and parallel flow element. The main point of novelty will be the simultaneous multithreaded execution of one single core system.

2. BUSINESS PROCESSES AND DOCUMENTS

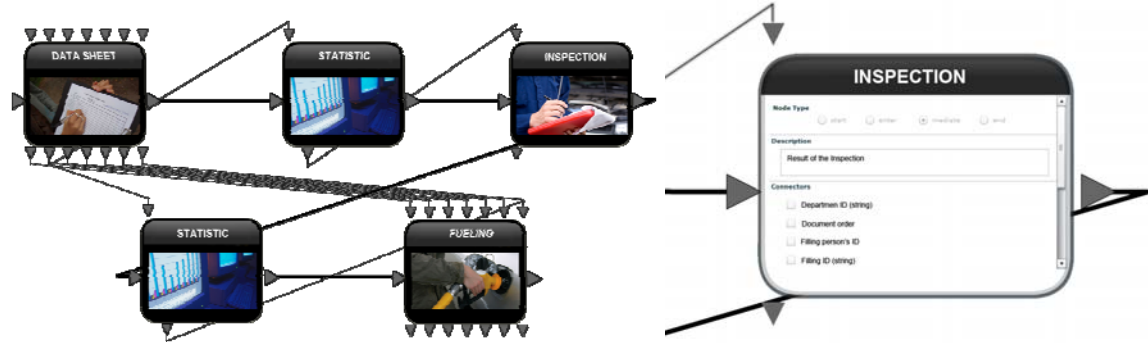
The concept of *Amnis* basically is built upon *processes*. The exact knowledge and deliberate observance of business processes are one of the basic conditions for successful operation of an organization (Khan 2004). Besides the processes the *documents* also form a basic component in the *Amnis* approach. Surprisingly a great number of companies deal with *document oriented systems*.

The *graph of the workflow processes* can be easily created and flexibly edited by the graphical interface which can be regarded as the strength of *Amnis*. The documents *flow* from nodes to nodes along the *edges* within the process graph. The activities defined within the nodes realize operations, such as create, filling in, modification, load and save etc. on the documents. In the representation by the workflow editor, the nodes of a workflow graph contain specific documents and queries to be executed on them. Based on the result of the queries new documents can be defined which will be created at the next node of the workflow graph, on the fly, on interpreting the flow graph. The edges of the workflow graph connect the fields of forms within documents situated in the nodes. The fields of forms can be understood as free and valued variables. Processes in *Amnis* can be defined by a graphical programming language tool which is called workflow editor.

Within an application the workflow manager models and implements the business logic currently effective for the company by editing the *Amnis* processes. The working processes can be created or modified without program any change or modifications in the backend of the *Amnis* system. The set of workflows defines the collection of operations in the company and displays them in a visualized form. Besides, the

hierarchical workflow system achieves the effective operations in the form of an executable program for the *Amnis* interpreter.

In the *nodes* of the process graph the typical basic activities of the enterprise can be placed. The nodes are filled in by tasks out of the list of basic activities at the organization (see: Figure 2). The resulting flowchart can be freely drilled down and up, moreover rearranged on the workflow editor user interface (see: Figure 3).



The documents may contain different number of data fields defined by the related document type schema. Zooming in to the nodes the data fields of the corresponding document type will appear (see: Figure 4.).

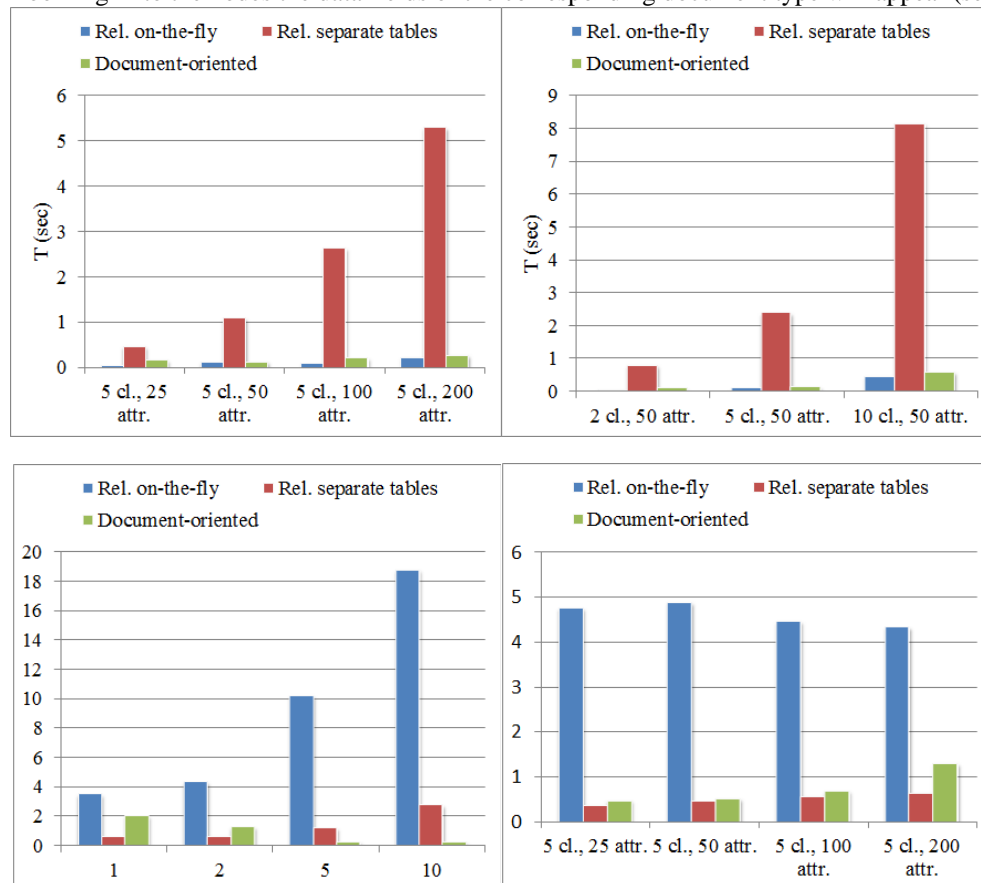


Fig. 5. Benchmark test of different data model implementations

The thick arrows define the control structure of the workflow graph, while the thin arrows show the data flow between the nodes of the processes. It means real data transmission when the flow graph is interpreted and the actual process step is evaluated. The nodes of the procedure graphs according to the execution can be classified by the next types:

Parallel: beginning from a single node the procedure can be executed on two threads in an asynchronous mode. (For example a logistics procedure: while on the first thread the purchase and delivery are prepared on the other thread the contract documents are produced parallel.)

Synchronized: it is a method to put parallel procedures into one execution thread. (For example, returning to the previous example, we have an assumption that purchase on the first thread is completed, the goods are ready to delivery but the process has to wait for the other thread for contract signing or money collection. When the obstacles to go on are eliminated then the two procedures join again in the delivery node and then the control passed on to continue in one single thread once again.)

Conditional: as the name suggests the execution of the node should be carried out in the case of a specific condition had been fulfilled. (When the employee uses his own car there is no need to fill in the cost calculation document.)

Iterative: the system executes the operation of this node multiple times according to a conditional parameter. (For example the procedure will send the summary of daily deliveries via e-mail.)

The parallel and the iterative control structures are new features being under development in *Amnis* 3.0. These features enhance the expressive power of the workflow model.

3. DATA MODEL, OBJECT-ORIENTED DATA STORAGE

In the *Amnis* system a large amount of documents that belong to different document types has to be stored in a single database. The number and type of attributes can highly vary among different kinds of documents. Storing such inhomogeneous data in a single database encumbers fast retrieval of data during querying; the fast retrieval is necessary requirement on any attribute of documents.

Amnis 1.0, as a prototype system was installed and used by several companies. After a few months of work, when a lot of documents had been created in the database, a large delay in the systems' response times was experienced. This was the main reason to develop *Amnis* 2.0 in cooperation with Eötvös Loránd Science University focusing on the database performance of the system.

Amnis documents belong to different types. Every *document category* has different set of descriptive data called document attributes. Each subtype inherits all the descriptive data of its parent document type and extends it with several additional attributes. This structure is similar to the concept of object-oriented programming, where categories correspond to *classes*, subcategories are provided with *inheritance*, and each record is an instance of a class.

Beside the maintenance of this taxonomy, document instances have to be stored and retrieved for all categories, with all the descriptive data of a certain category. Since the database stores a huge amount of documents, functionality is needed to retrieve not just single objects but a set of documents based on different filter conditions. These filter queries contain conditions based on the descriptive data, such as retrieving documents with a specific attribute value, or class conditions, such as retrieving all the objects in a specific category and its subcategories.

According to the permanent changes of business environment and processes, the *Amnis* documents are subjects to change, therefore new versions of document types are created. The queries of the documents created with the previous, deprecated document types have to be as fast as the queries for the new documents. Although 30-60 seconds response time is acceptable for a part of queries, the majority of them should provide result in 1-2 seconds response time, independently of the actual number of documents stored.

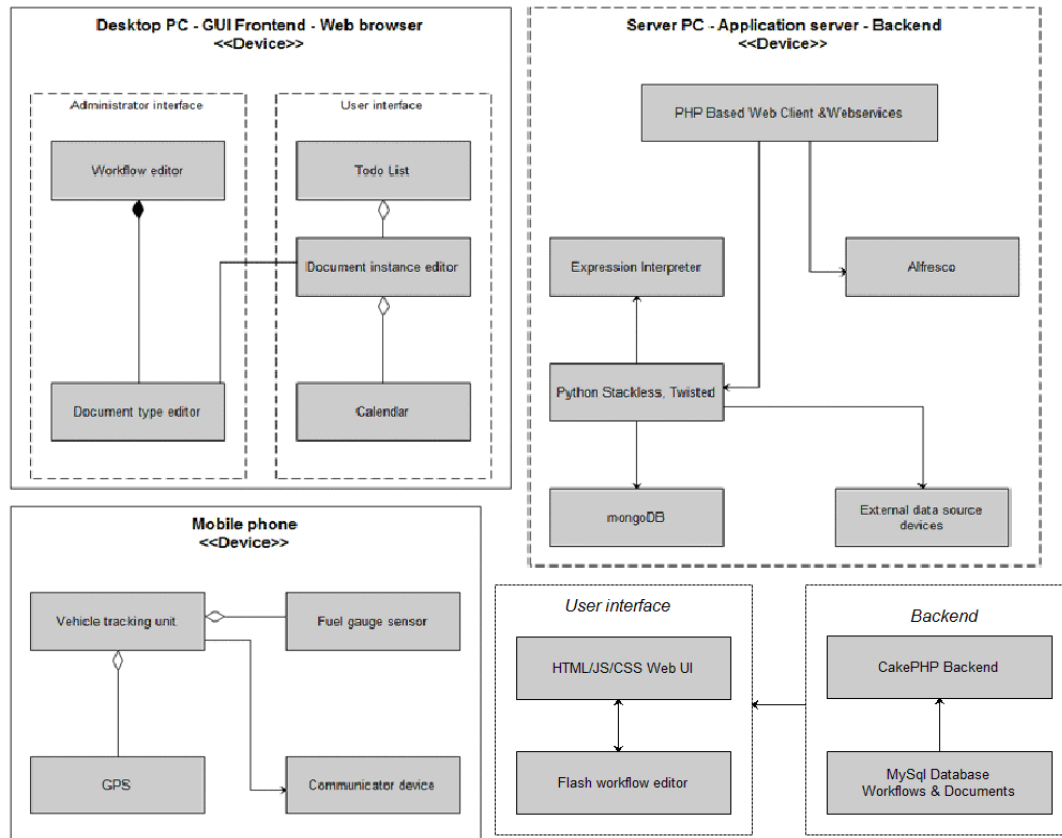
During the development of *Amnis* 2.0 several different implementation approach were studied. Finally three prototypes were implemented using the MySQL and MongoDB database engines respectively. In two cases the solution was based on relational database management system: in the first case the "entity-attribute-value" data model's slightly modified version was implemented, in the second case the data were stored in on-the-fly generated data tables. These two solutions were compared with the third, document-oriented database solution, which is a rather natural approach of the problem.

The benchmark environment and results in Figure 5 are presented in details in an article (Máriás 2012). On the figure chart no. 1, 2 and 4 present two values: the first one indicates the number of classes, the second shows the average number of attributes in the database used for the benchmark. The first two charts show the average query time of documents belonging to a certain document type, the first with growing number of document types but fixed average attribute number, the second with growing number of attributes but fixed number of classes. The second row of the figure shows the benchmark results for filtering queries. The diagram no. 3 shows the growing number of attribute filter options, while the fourth one shows the query times for fixed number of filter options with growing number of attributes. Based on the results of the

benchmark, the MongoDB based solution was implemented in *Amnis* 2.0. The result of the development was a significant enhancement of database performance.

4. SYSTEM ARCHITECTURE

The software system architecture of currently operating *Amnis* 2.0 is shown in Figure 6. The data model is implemented with a combination of relational and schema-less, document oriented database systems. In *Amnis* 2.0 workflows are stored in MySQL, while documents are stored in the open-source MongoDB database system.



Amnis 2.0 is available in SaaS (Software as a Service) business model. *Amnis* can be regarded as a web information system (Molnár 2011, Molnár 2013), as instances are installed as a web application on a web server and the application frontend is accessible by the client with a web browser software. Although the general frontend is built using the standard HTML, CSS and jQuery, the workflow editor is written in Adobe Flash. The application backend is developed in PHP, using the CakePHP framework and its MVC implementation.

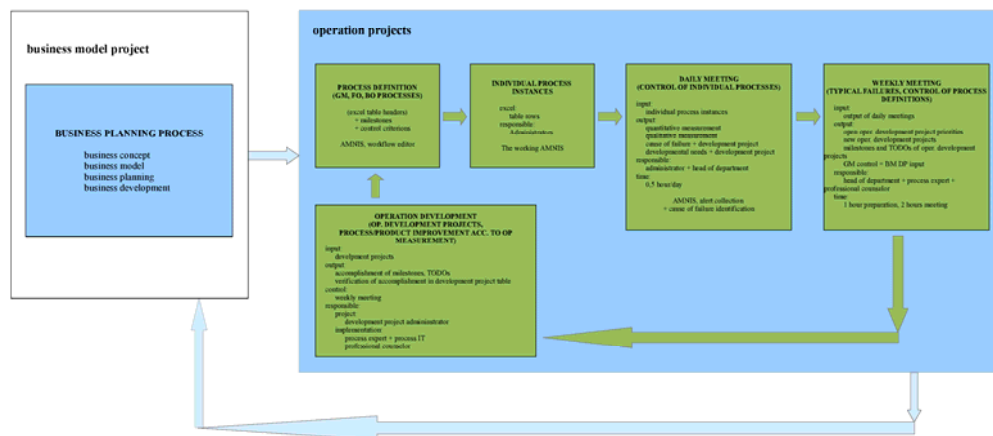
Each company as a tenant has a separate instance of the system with their own specific workflows that are built utilizing of the *Amnis* features. This separation is obtained by using virtual private servers. Above the basic features, each instance can be extended with special functions called toolkits. *Amnis* 3.0 as a new rerelease has been planned and has already been under development. Its architecture consists of various modern software-components. The picture below presents the most important logical components of the system.

The system user interface will be developed using modern web frontend technologies including HTML5, jQuery and AJAX. The interactive functions of *Amnis* 3.0 can be accessed using this frontend: general workflow, document and document type administration. This interface provides the functionalities to actually run workflow and submit user data during the interactive steps of the workflow.

The workflow editor with GUI is one of the most advanced and spectacular part of *Amnis* 3.0. This function is also web-based, using HTML5 and JavaScript and it makes possible the creation and flexible modification of the graphical representation of the workflow graph. The interface has a zoom and topological re-order function. The vertices of the graph represent the workflow steps, the directed edges show the data transfer between them. When the workflow is being executed the actual data transfer is performed.

We have designed a new, more advanced environment for the *Amnis* 3.0 version, than the previous versions. In the former versions installation of a new instance was needed for every company, which made the maintenance much harder. The new system includes an application-server, which is capable to run multiple threads simultaneously. This way the structure of the new software both from design and development viewpoint – as it is used to by these days – breaks down to the following three layers: database handling, business logic and user interface.

VRC PROCESS SCHEMA



The functions of application server are accessed from the web user interface through proxy server which is implemented in PHP. To actually run of the business workflows are performed by the standalone application server which was developed in Python programming language. Implementation of thread management that is independent from operating systems is based on Stackless and Twisted Python frameworks.

According to the research and benchmarks the inhomogeneous documents and all workflows of the system are stored in MongoDB, which is a non-SQL, document-oriented database system.

The system architecture plan of *Amnis* 3.0 already contains a part which is going to be developed in mid-term. The system will be extended with mobile communication and GIS functions which are going to be a separate subsystem. The foundations of storing the spatial data in *Amnis* are already studied (Máriás 2011); a detailed database plan is ready for development (Amnis 2014).

5. ILLUSTRATIVE EXAMPLE

As we have seen above, the concept of *Amnis* is grounded in strong theoretical background and best practices. The research and development within the field of software can be considered as experimental and empirical science. In this project, this phenomenon appears in the following form. Firstly, the leading development company (*Amnis Laboris*) started analyzing and monitoring its own business processes. The tool for investigation was the actual version of *Amnis* system. Secondly, on having beneficial experiences the company started to provide a consultancy service in business process management by exploiting the function of the available system and offering a remote access solution.

The major business of the company is the rental of mobile refueling stations. This activity is information technology intensive operation. The company offers a comprehensive and complex solution from delivering infrastructure till operating the whole system with the related services. The activities at the company and its partners can be considered as process-oriented intensively. Figure 8 contains a basic scheme for monitoring business processes. On the left hand side, there is a blue rectangle within a white box to represent a comprehensive, company-wide process. The main task of this process is to design and develop the business

processes of the enterprise. After development and implementation of processes, all changes will be carried out at this logical spot. For example, the modification of tasks for job positions and for employees, re-engineering the processes, offering new service or product either the supply or demand side from viewpoint of *Amnis* approach. The common feature of business partners on developing their own business is that when they procure the necessary complex equipments at the same time they want the whole process automate at a highly advanced level. Consequently, they acquire the required software components as well.

In practice, the business process engineering, re-engineering and reviewing at enterprises does not happen very frequently. Succeedingly, the proposals for new IT solutions appear late, and the derived requirements for development are taken into consideration only afterwards so that the changes will be implemented effectively only thereafter. In Figure 8, the first green box refers to the workflow editor that was shown in Figure 3 and 4. The process re-engineering happens in line with a controlled change management, i.e. the effective amendments take place only after the approval by the enterprise management. The modification should accurately reflect the proper operations, and the effective process re-engineering. The second green box describes daily operations of *Amnis* system. Actually, it depicts the process-oriented support for business processes of the enterprise, i.e. the document handling, the steps of inputting the fields of documents. The third green box represents the essential point of monitoring. The *Amnis* system controls and monitors phenomena that seem deviation from the normal operations, and creates signal in the form of 'alert'. The software collects these signals systematically in a daily routine. At the developing company, the system highlighted that the time between quotations and contracting was longer than that can be considered reasonable; this statement was in concert with the feeling of staff. The possible causes contained the following items:

- The contact person at the business partner lacked the necessary competence;
- The quotation and bid did not fulfill the requirements of potential business partner;
- The clerk who is responsible for relationships of a specific business partner is overloaded;
- The developing company reacts slowly and belatedly.

The experiences led to the conclusion that a specific, dedicated *matrix* is needed to specify thoroughly the interdependencies between processes and tasks in the case of mobile refueling stations enterprise-wide business process chain.

The important nodes of the workflow as e.g. fueling, compliance-control appears in the matrix; then the work-breakdown structure of tasks extended by roles and actors. For example, out of the various responsibilities some of them are dedicated to partners some of them to the service provider (in this case *Amnis*).

The elements of matrix are expanded by potential cost reduction, hardware requirements and costs. The matrix is itself a specific general document that can be handled as a form with fields provides an important analysis tool that accelerated the requirements analysis phase. Following the agile management pattern, there is weekly assessment where the actual set of alerts is evaluated; and root cause analysis can be carried out during maximum one hour. If it is necessary a defect elimination project can be commenced to define exactly the problem and improve the situation.

The result of the project is an initiative for re-engineering the actual business activities in term of flows and procedures related to information processes along with necessary re-organization steps. This approach does not focus only on the removal of defects but on the process improvement as well. The empirical data obtained from the system demonstrates that the performance of a newly defined business process can achieve 80% efficiency in spite of the initial surprising low 20-30 % performance level.

6. SUMMARY

Several modern and system paradigms appear in the new version of *Amnis*. During the development of process management tool with evolutionary approach new control structures appeared in the control flow beside the usual construction of sequence and conditional. The most important are the inclusion of the parallel execution and the loop control structure. In case of *Amnis*, the biggest challenge was to solve the database performance problem mentioned above. After reviewing several database management systems and their benchmarks, we chose the open source document based database system MongoDB.

The *Amnis* system is a valuable tool for describing business processes even when it is "empty". Parallel with the general system development, several applications got developed as examples to support different activities, such as HR management, mobile fuel stations re-filling, payroll management, vehicle tracking and logistics. These subsystems were built by simply using the tools provided by *Amnis*. At the moment 148 companies run and use *Amnis* in their everyday business operation from various fields.

In the future we are going to study the differences between *Amnis* and other systems with similar capabilities such as TIBCO iProcess Engine, Microsoft Workflow Manager. We are also going to implement the mobile communication and GIS subsystem to be able to build richer application based on *Amnis*.

In future research we plan to provide a more formal foundation for the whole system. We make use the results published in articles by one of ours authors (Molnár 2012, 2013). The document and business process model can be treated in a unified framework that may help the understanding the functionalities of *Amnis* system.

REFERENCES

- Amnis* project, developer portal: <http://docs.amnisworkflow.hu>. (Retrieved 2014)
- Becker, J., Rosemann, M., & von Uthmann, C. (2000). Guidelines of business process modeling. In Business Process Management (pp. 30-49). Springer Berlin Heidelberg.
- Khan, R. N. (2004). *Business Process Management: a practical guide*. Meghan-Kiffer Press.
- Máriás, Zs., Giachetta, R. (2011). A study of various techniques for storing inhomogeneous descriptive data of digital maps. *Periodica Polytechnica, Electrical Engineering*, 55/3-4 103–110, web: <http://www.pp.bme.hu/ee>, *Periodica Polytechnica* 2011.
- Máriás, Zs., Nikovits, T., Takács, T. , Giachetta, R. (2012). A Study of Large Amount of Inhomogeneous Data in Workflow Management Systems. *Annales Univ. Sci. Budapest., Sect. Comp.* 37 275-292.
- Molnár, B., & Benczúr, A. (2013). Facet of Modeling Web Information Systems from a Document-Centric View. *International Journal of Web Portals (IJWP)*, 5(4), 57-70.
- Molnár, B., Benczúr, A., & Tarcsi, Á. (2012). Formal approach to a web information system based on story algebra. *Singidunum Journal of Applied Sciences*, 9(2).
- Van Der Aalst, W., & Van Hee, K. M. (2004). *Workflow management: models, methods, and systems*. MIT press.