Metamodel Based Framework for Informative System Development - ISTechnology

Jānis Iljins and Ģirts Karnītis

University of Latvia, Raina 19, Riga, Latvia
janis.iljins@di.lv

Abstract. The paper describes methodology and framework for model based informative system development and maintenance ISTechnology (IST). The methodology and implementation of the IST, as well as the domain specific business process modelling language created for IST and its implementation within the framework is discussed. Using IST several financial informative systems are developed for different banks in Latvia. Analysis of the time spent for the development and maintenance of informative systems used in real exploitation shows: usage of the metamodel based framework minimizes effort needed for the development and maintenance of informative systems.

Keywords: ISTechnology, Metamodel, Informative system development, Model-driven software development, Domain-specific business process modeling

1 Introduction

The paper deals with the problem of how to effectively implement and maintain an informative system in a financial institution (for example, a bank or an investment fund) characterized by a changing business environment and complex business processes. We created a system design, development and maintenance method, in which developer creates a model for implementation and maintenance, characterized by the fact that high level system design is stored in the system repository. Based on the method we developed the Information Technology Development and Maintenance methodology and framework ISTechnology (IST). On the basis of IST we have created a number of systems which are introduced in several financial institutions.

The following chapters provide a more detailed description of the IST and its assessment. The first chapter describes the similar solutions and their similarity and differences with IST. The second chapter describes the IST metamodel. The third and fourth chapters contains IST implementation details - the IST framework description and the IST domain specific business process description language. The fifth chapter describes the evaluation of the use of IST by analyzing the amount of time spent on

Lupeikiene A., Matulevičius R., Vasilecas O. (eds.):
Baltic DB&IS 2018 Joint Proceedings of the Conference Forum and Doctoral Consortium.

maintenance and the number of changes that have been received for systems that are actually developed using IST.

2 The Current Situation

IST ideas have been tested in a series of systems [1] and have proven their livelihoods. This chapter describes similar model based solutions and their differences from IST.

Most widely known solution is Model Driven Architecture (MDA). MDA is based on models and their transformations [2]. The MDA consists of Platform Independent Model (PIM), Platform Specific Model (PSM) and Code and defines how they will be derived from each other. In the MDA approach, code generation has become an inevitable step [3]. One of the major shortcomings of MDA is the failure to provide non-functional requirements (such as performance or usability requirements) [4].

Compared to MDA, IST offers a different approach, which can be termed a "broken" approach to modeling systems. An MDA-like method describes the system's main structure, the main user interface, and system specific modules that are not modeled, but are developed using traditional programming methods. System specific modules are called from IST framework.

The "broken" approach to the modeling and the systems created by it eliminates the main disadvantages of the classic MDA:

- The system is modular - it consists of many small items, which are controlled by one central item - the functional shell. In its turn, the functional shell operates according to the system model - its functionality is modeled.
- The metamodel used for modeling is small enough and simple to be clear and understandable.
- Usability is at the same level as it is in traditional systems.

The next class of similar systems is the workflow systems. A lot of different workflow management systems exists, such as YAWL [5], ADEPT2 [6], BPEL [7], JIRA [8], and KiSSFLOW [9].

Virtually all workflow management systems offer the ability to define configurable workflow scenarios and operate in accordance with defined scenarios, which is to interpret the defined scenarios. Workflow management is also one of several IST functions. Consequently, IST can be considered to some extent as a workflow management system.

Document management systems have to be mentioned as another similar type of system. Document management systems offer configurable access to business documents, provide communication between employees. Wide comparison of document management systems is in [10]. Document management systems share common features with IST, such as business process configuration and user interface configuration. However, there is also a significant difference between IST and document management systems. Document management systems have predefined business objects, from which the central is "Document". Systems are primarily intended for storing documents and controlling their versions. At the same time, IST provides business processes for arbitrary processing of business objects in a database defined by the system's developers.

The BPEL [11] language is created for configurable business process management. An interpreter for BPEL scripts is maintained [12]. An essential prerequisite for using BPEL is the provision of system functionality with the help of web services. There are several tools that provide execution of the process according to the BPEL. The most significant differences between IST and BPEL:

- BPEL does not include user interface modeling, which is a major component of the IST model.
- IST is designed for the development of one organization's information system and its business processes, but typical BPEL solutions are designed for the integration of several information systems - the business process for the exchange of information between informational systems.

Most similar to IST are those methods in which the metamodel based system model is an integral part of the developed information system and the model determines the functionality of the system. Such solutions are very rare, for example [13].

Similarity with IST also exists with systems based on the domain-specific business process description language BiLingva [14]. In these systems, the business process description in the BiLingva language is defined and the system executes it - the model is interpreted. At the same time, there are differences between IST and BiLingva systems:

- If BiLingva's business process description is at the heart of system functionality, IST offers to configure not only the business process but also the user interface - the availability of business data and the availability of operations with data.
- BiLingva focuses on state-of-the-art decision making and transition of an object from one status to another. Both of these factors are of little importance in the description of the business process in the IST. At the same time, the language of the IST business process offers complex transitions of the status of objects.
- The IST system is modular. This is not true for BiLingva systems, which consists of one large module - the interpreter of the business process.

3 The IST Metamodel

The IST metamodel is shown in the Fig. 1. For each class, only the essential attributes and methods that are most important to understand the IST work are given.

Class “Task” describes the tasks. Classes "Organization", "Department" and "Employee" allow you to store information about an organization (for example, a bank) and its departments. Information about the employees of the organization is used both as a contact database and as a database of system users who have the right to connect to the system. Classes "Object class", "Event" and "Event details" ensures the operation of the system as well as logging of database data changes. Log fails contain information about opened screen forms as well as changed data values before changes.

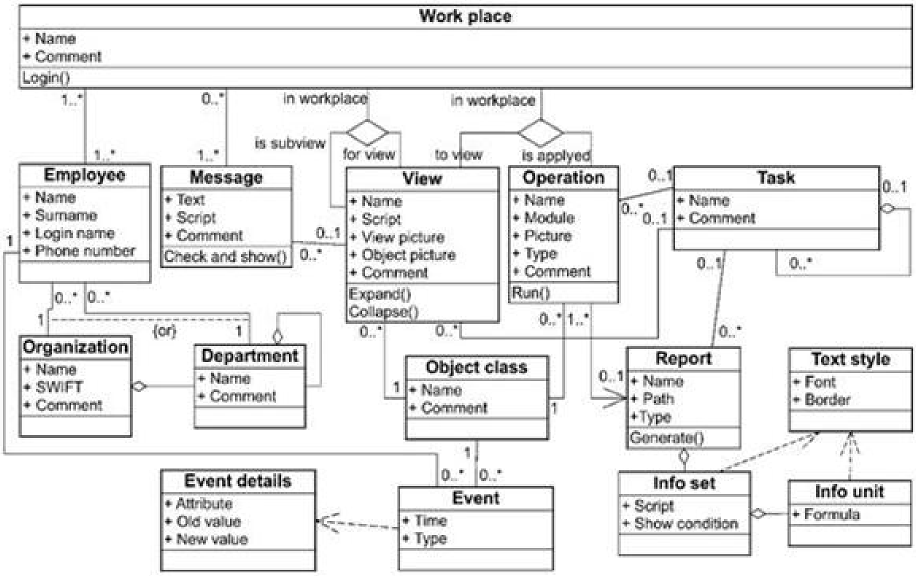


Fig. 1. UML class diagram of IST metamodel

Core of IST and main user interface is described in classes “Work place”, “View”, “Operation” and “Message”. Those classes contain information necessary to run system functional shell – system design interpreter, which interprets system design encoded in metamodel. Class "Work Place" defines a "workplace" that consists of objects that a user can see at the particular workplace and the activities that can be performed on those objects. Class "View" defines business objects available to the user at the specific workplace. Views can be defined as hierarchical structure for each level defining filters and sorting facilities available. Class "Operation" describes the modules that created outside IST and are invoked from IST and perform business functions with selected objects. Class "Message" describes system alerts and reminders for changes in database data and for user actions.

The reports required in the informational system are defined in the classes "Report", "Info set", "Info unit" and "Text style". Class “Report” contains main definition of report, class “Info set” contains tables that are used in report, Class “Text Style” contains report visual look.

4 Domain Specific Business Process Language

It was decided not to use the classic BPMN [15] in the definition and automation of IST business processes, as IST requires a much narrower and more specific process description that can be understood not only by IST, but also by end-users of the systems as well. Therefore, a special IST business process modeling language was developed.

The business process model depicts the status of business objects as well as business processes that change the status of business objects. State transitions are defined, indicating the processes that make this transition.

Fig. 2 shows example of business process – one of the simplest but typical of the Securities Recording System (VUS) business processes - the process of processing securities transactions on behalf of the client. The business process starts with entering the transaction application and executing the transaction on the market. Thereafter, a registration is made of the transaction application, which is approved for settlement.

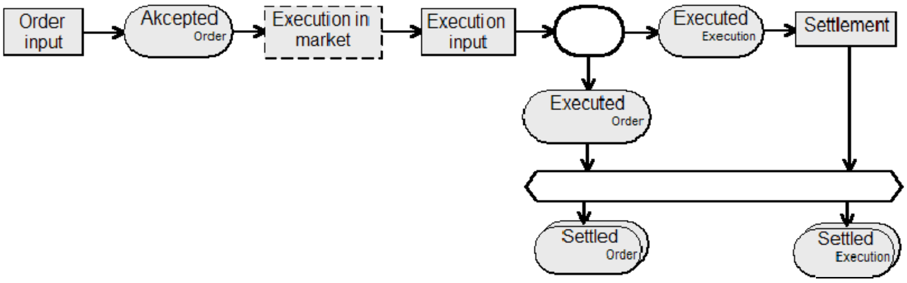
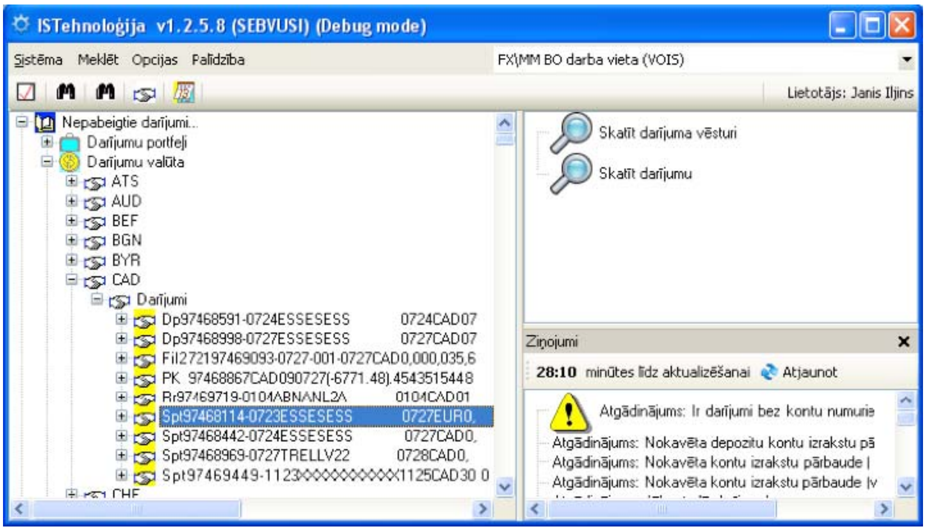


Fig. 2. An example of a domain specific business process for an IST

5 Implementation

The metamodel described in the previous chapter is used by the main program of IST framework – IST functional shell, which interprets the system design description (model). The functional shell controls user access to the database. The functional shell also provides access of a specific workplace available to a specific user. The windows display a views tree – views, objects underneath the views, which may have subviews under them. Selecting a particular object or view displays the operations that can handle the selected object or view. After a certain time interval, messages in the message window are renewed. The main window of the IST functional shell, as seen by the system user, is shown in Fig. 3.

IST views form a hierarchy. Each view creates a new level of hierarchy in which the information is selected from the DB. Query for view:



```
Nepabeigtie darījumi...
  Darījumu portfeļi
  Darījumu valūta
    ATS
    AUD
    BEF
    BGN
    BYR
    CAD
      Darījumi
        Dp97468591-0724ESSESESS          0724CAD07
        Dp97468998-0727ESSESESS          0727CAD07
        Fil272197469093-0727-001-0727CAD0,000,035,6
        PK  97468867CAD090727(-6771.48).4543515448
        Rr97469719-0104ABNANL2A          0104CAD01
        Spt97468114-0723ESSESESS          0727EURO,
        Spt97468442-0724ESSESESS          0727CAD0,
        Spt97468969-0727TRELLV22         0728CAD0,
        Spt97469449-1123XXXXXXXXXX1125CAD30 0
    CHF
```

```
28:10 minūtes līdz aktualizēšanai
Atgādinājums: Ir darījumi bez kontu numuri
Atgādinājums: Nokavēta depozītu kontu izrakstu pā
Atgādinājums: Nokavēta kontu izrakstu pārbaude |
Atgādinājums: Nokavēta kontu izrakstu pārbaude |v
```

Fig. 3. Main window of IST framework functional program

```
SELECT DISTINCT C.CURR_ID, V.CURR_NAME
FROM VOP_CURR C, VP_TRANS T
WHERE C.CURR_ID = T.CURR_ID_TRANS
ORDER BY 2
```

Query returns the list of transaction currency shown in the picture Fig. 3. It has a subview that contains transactions for each currency. Query for subview:

```
SELECT TRANS_ID, TRANS_NAME
FROM VP_TRANS
WHERE CURR_ID_TRANS = :id[1]
```

In query, parameter: `id[<level>]` contains an identifier in the upper level view for the user-selected value.

Operations are attached to the view (for example, to add a new item) or object (for example, edit, delete, print). Operation is carried out by calling the functions from particular system-specific module created especially for that system. Operation definition contains a reference to the DLL file and the name of the function that is called. The call is technically implemented as a call to the DLL function.

The operation may also be a creation of a report. IST framework includes the DIREPO tool, which, based on the reporting module's report definition, is able to create a report in popular document formats such as Microsoft Word (docx format), Microsoft Excel (xlsx format), PFD.

Business process models are developed with the tool Grade 2 developed by LUMII [16].

The business process description contains too little information for IST configuration automatically created from model to include all the configuration options possible. There is a lack of information about the hierarchy of views, which is widely used in IST workplace configurations. Consequently, the automatic IST configuration can be used to:

- Create a DEMO version of the system immediately after business processes analysis,
- Create initial configuration according the original business process in cases when users can not define configuration requirements.

The basic idea behind generating IST configuration is the transformation of models from business process diagrams to IST configurations according the configuration metamodel. A transformation tool has been developed that can convert defined diagrams to database queries (SQL scripts) that make up the IST configuration.

6 Evaluation of Effectiveness of Methodology and Framework

Based on IST, 4 large information systems have been developed, each of which is implemented in a number of institutions:

- Currency operations information system – implemented in 2 institutions;
- Securities accounting system – implemented in 6 institutions;
- Fund management system – implemented in 4 institutions;
- Pension fund system – implemented in 2 institutions.

The experience of using IST has enabled experts to develop guidelines for the assessment of labor productivity based on the ideas of [17]. The expert assessment suggests that the creation of new functions requiring just configuring the IST model requires about 3 times less workload than the new functions for which a new application needs to be developed.

Architecture of IST framework made it easier to migrate to another environment (The first version of the IST was created in Centura environment and was later migrated to .NET):

- IST is common to several systems that require migration. When migrating multiple systems, the IST code had to be re-written only once.
- IST based systems have a modular structure. The system consists of many small applications. Consequently, the migration of the system could be done gradually in parts. During the transition period, both platforms can be used in parallel. Some apps work in a new environment, some still in the old one.
- Because IST components are reused in different contexts, by overriding one component, several business functions are transferred to the new platform. For example, when overwriting 15% of applications in the VOIS system, the new platform already operates 43% of the system functions.
- The IST metamodel is a platform-independent description of system functionality. By moving to a different platform platform-specific metamodel interpreters, the metamodel and the system functionality description did not have to be changed.

Since September 2003, database of requests for changes has been carried out in the change request system for the systems developed and put into operation using the IST. During this period, 9429 changes requests were processed and completed. The study [18] analyzed their content and the time devoted to them. The listed change requests were analyzed based on what was changed in the system due to the request:

- IST applications - Improved or enhanced IST functional shell or other IST applications that support its operation, as well as business model input or system configuration.
- Configurable components - reconfigured multi use components or added a new operation that uses an existing component. In other words, the request for change has been implemented by introducing changes to the business model.
- Reports - reconfigured existing reports or added new ones.
- Specific applications - specific applications developed for a specific system - modified algorithm or application program code.
- Changes to the data - the change request did not require any changes to the system. Instead, the database data has been corrected or only end user consultation was needed.

Fig. 4 shows the percentage of requests for change requests and the percentage breakdown of time by these categories. The left diagram shows the distribution of the number of requests, but the right diagram shows time spent.

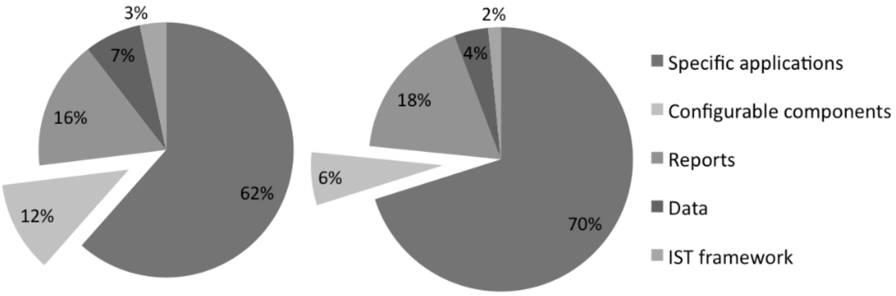


Fig. 4. The number of requests for change and the time distribution

The diagrams show that implementation of business model changes in the business model requires about two times less time than the implementation of changes in specific requirements.

7 Conclusions

The "broken" approach used by IST allows you to combine both the flexibility of model-based applications with the benefits of traditional apps - usability, security, and more. The IST method provides the following benefits:

- IST has its own metamodel, which allows you to define a system model or configuration that defines the functionality of the system being developed. Consequently, part of the system functionality described as data that is stored in the database and can be easily changed even without the change of system software. This feature allows you to adapt the system to a changing business environment and changing user requirements.
- IST is a model interpreter. IST fully runs without generating code.
- The so called "Broken" approach to business functionality development. The "broken" approach does not fully model the functionality of the whole system to the executable code, but only the superstructure of the system. Modeling does not include specific and precise low level requirements for business functionality. It is developed using traditional programming methods (such as Microsoft .NET environment, C # programming language) and embedded in business specific system items that the system calls according to the model. Consequently, the IST metamodel and functionality description is simple and transparent. At the same time, non-functional requirements for complex business specific features, such as usability, are analogous to traditionally developed systems.
- IST systems are modular. They are made up of IST framework and a series of independent business functionality modules. The IST framework consists of an IST functional shell and IST service modules. The IST framework calls business functionality modules and IST service modules. Business functionality modules are independent of each other, they rarely call each other. Only they all use one shared database. Such a system design considerably facilitates changes to the system and regress testing, because changes in the new version is made only in some modules whose replacement does not affect the functionality of other modules.

The informative system developed with the aforementioned features requires less development and maintenance time than traditional systems or systems without these features. The results of these studies show that using IST saves system design and maintenance time.

References

1. Iljins J.: Design interpretation principles in development and usage of informative systems. In: Scientific papers, University of Latvia. 2004; 669:99-106.
2. Kleppe A., Warmer J., Bast W.: MDA Explained: The Model Driven Architecture™: Practice and Promise, Addison Wesley, 2003, ISBN: 0-321-19442-X.
3. Lajmi A., Ziane M., Ziadi T., Cauvin S.: A multi-view model-driven approach for packaging software components. In: Proceedings of the 2010 ACM Symposium on Applied Computing 2010 Mar 22 (pp. 2297-2304). ACM.
4. Ormeño YI, Panach JI.: Mapping study about usability requirements elicitation. In: Advanced Information Systems Engineering 2013 Jun 17 (pp. 672-687). Springer Berlin Heidelberg.
5. MacCaull W., Rabbi F. Nova Workflow: A workflow management tool targeting health services delivery. In: International Symposium on Foundations of Health Informatics Engineering and Systems 2011 Aug 29 (pp. 75-92). Springer Berlin Heidelberg.

6. Dadam P., Reichert M., Rinderle SB, Jurisch M., Acker H., Göser K., Kreher U., Lauer M.: ADEPT2-Next Generation Process Management Technology. (2007).
7. Curbera F., Goland Y., Klein J., Leymann F., Weerawarana S.: Business Process Execution Language for Web Services, version 1.1. (2003).
8. JIRA homepage, https://www.atlassian.com/software/jira
9. Bicevska Z., Bicevskis J., Karnitis G.: Models of event driven systems. In: International Baltic Conference on Databases and Information Systems 2016 Jul 4 (pp. 83-98). Springer International Publishing.
10. Rama J., Bishop J.: A survey and comparison of CSCW groupware applications. In: Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries 2006 Oct 9 (pp. 198-205). South African Institute for Computer Scientists and Information Technologists.
11. Mendling J.: Business process execution language for web service (bpel). In: EMISA Forum 2006 Aug (Vol. 26, No. 2, pp. 5-8).
12. Christos K., Costas V., Panayiotis G.: Enhancing BPEL scenarios with dynamic relevance-based exception handling. In: IEEE International Conference on Web Services (ICWS 2007) 2007 Jul 9 (pp. 751-758). IEEE.
13. Tegegne, Abel A.: A Model Driven Application Framework for Managed Processes. Diss. University of Ottawa (Canada), 2010.
14. Ceriņa-Bērziņa J., Bičevskis J., Karnītis Ģ.: Information systems development based on visual Domain Specific Language BiLingva. In: IFIP Central and East European Conference on Software Engineering Techniques 2009 Oct 12 (pp. 124-135). Springer, Berlin, Heidelberg.
15. Owen M., Raj J.: BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard. 2003. http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf
16. Sproģis A., Bārzdiņš J.: Specification, configuration and implementation of DSL tool. In: Frontiers in Artificial Intelligence and Applications, vol. 249, IOS Press, p. 330-343, 2013 (SCOPUS)
17. Jorgensen M. Practical guidelines for expert-judgment-based software effort estimation. Software IEEE, Volume 22, Issue 3, pages 57-63. (2005)
18. Iljins J.: Metamodel based approach to IS development and lessons learned. In: Proceedings of the Eighth International Baltic Conference Baltic DB&IS 2008, 2008 (Vol. 167, p. 178). Tallinn.