Design of Visual Repository, Constraint and Process Modeling Tool based on Eclipse Plug-ins

Rushiraj Heshi
Department of Computer Science and Engineering
Walchand College of Engineering, Sangli

Smriti Bhandari
Department of Computer Science and Engineering
Walchand College of Engineering, Sangli

ABSTRACT
Master Data Management requires creation of Central repository, applying constraints on Repository and designing processes to manage data. Designing of Repository, constraints on repository and business processes is very tedious and time consuming task for large Enterprise. Hence Visual Repository, constraints and Process (Workflow) modeling is the most critical step in Master Data Management. In this paper, we realize a Visual Modeling tool for implementing Repositories, Constraints and Processes based on Eclipse Plugin using GMF/EMF which follows principles of Model Driven Engineering (MDE).

General Terms
Model Driven Development/Architecture, Software Engineering, and Master Data Management.

Keywords
EMF, GMF, GEF, Repository, Constraint, Process.

1. INTRODUCTION
Currently Master Data Management Systems [1] have been widely used in Enterprise information systems. The Master Data Management is a way to unify, manage and integrate references data across the Information System of the company. Master Data is Right view of Information delivered to the right people and processes at a right time. In Enterprise Environment Master Data are stored in CRM, ERP etc. For Master Data Management systems Repositories are main component for data storage and access. All enterprise data and records, master and reference, are created, stored, and managed in repositories. With the increase in Enterprise Data, repository structure increases in size. It is difficult to design repository for enterprise with large Repository structure. There are several challenges to manage structure/properties of repositories and define relationship between repositories.

In Master Data Management environment a Repository constraint allow enterprise to specify more complex constraints on attributes. As repository size increases, it is difficult to design constraint on repository. Hence, Repository and Constraint modeling is important aspect of design issues. Workflows are integral to enterprise. Organization supports workflows to manage business process. Workflow modeling is critical step of workflow application technology. In the following text, section II describes related work done. Then section III presents backgrounds and research foundation regarding modeling tool generation. Section IV describes design of visual modeling tool for repositories, constraints and processes. Then, Section V gives conclusion of tool designed.

2. RELATED WORK
Last decade, tremendous amount of research conducted on designing of visual modeling tools regarding Master Data Management which is based on Model Driven Engineering. Repositories required for storing organization important data. Previously repositories and constraints on repositories were written in xml format. Sadiq, et al. [2] used a graphical symbolic language to describe the business processes. Vander proposed using workflow-net to describe the business process [3] based on Petri net. Knolmayer, et al. [4] proposed a workflow model based on ECA (Event-Condition-Action, ECA) rules. Davu1cu, et al. [5] described and analyzed the business process using the CTR (Concurrent Transition Logic, CTR). Fan Yu-shun, et al. [6] proposed a workflow modeling method based on coordination theory and feedback mechanism. No tool focusing on design of repositories and constraint of repositories. Design of repositories and implementation of it is very tedious task in master data management environment. Also applying rules on repositories and managing constraints on repositories is also tedious task in Master Data Management.

In this paper, proposed the design of a visual Repository, Constraint and Process modeling tool based on Eclipse plug-in. Its basic features include: repository, constraint and process object creation, editing and the import and export of the repository, Constraint and process models.

3. BACKGROUND AND RESEARCH FOUNDATION
Thorsten Arendta, et.al [7] Describes the paradigm of model-based software development which provides the efficiency and quality of software development. Especially in model-driven software development, models become primary artifacts where quality assurance of the overall software product depends on the quality assurance of involved software models. Model-Driven Engineering (MDE) [8] is the software engineering approach that creates increasingly detailed models through transformations, until code is produced. Models are defined according to a meta model, but the growing number of meta models led to a higher abstraction level to describe meta models: meta-meta models. Arguably the most widely used MDE framework is the OMG's Model-Driven Architecture approach, which includes UML (Unified Modeling Language) models, OCL (Object Constraint Language) constraints and MOF (Meta-Object Facility) meta models and meta-meta models.

EMF [9] is a code generation facility modeling framework for building tools and applications based on a structured data model. On the base of a model specification

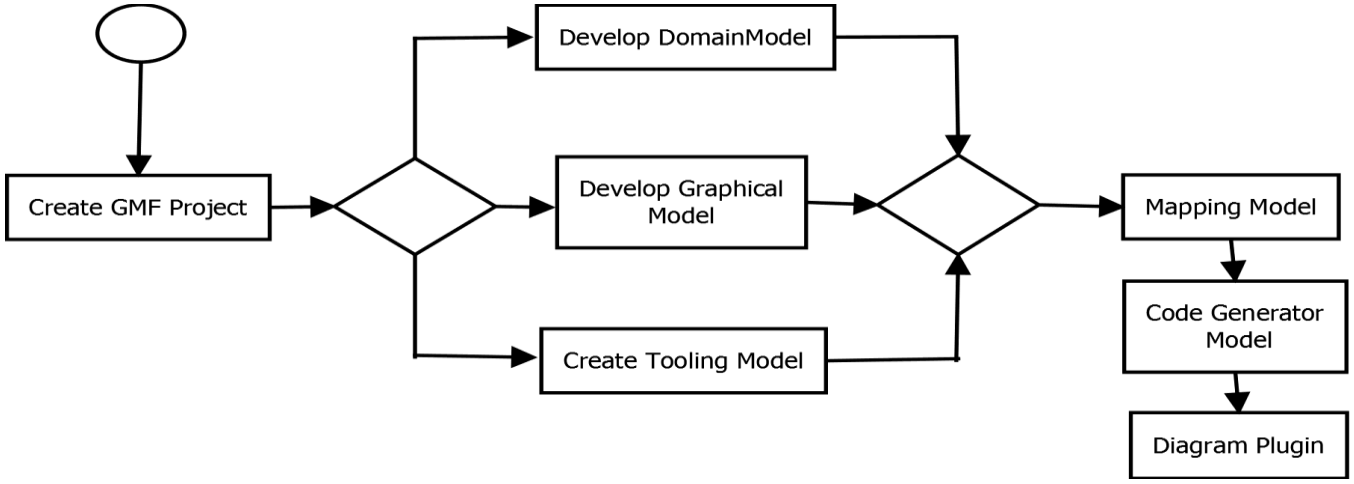

Fig 1: Graphical Modeling Framework

expressed in XMI, EMF provides runtime and tools support to produce a set of adapter classes for viewing, command-based editing of the model, Java classes for the model, and a basic editor. Models can be specified using annotated Java, XML documents, or modeling tools like Rational Rose, and then imported into EMF. EMF consists of three fundamental parts:

Ecore is the core of the EMF which includes Metada Model for describing Models. EMF provides runtime support for the models with change notification and persistence support with default XMI serialization.

EMF.Edit - The EMF.Edit includes generic classes for building editors for EMF models. EMF.edit provides reusability for building editors.

EMF.Codegen - The EMF code generation is capable of generating code to build a editor for an EMF model. The generation facility uses the JDT (Java Development Tooling) component of Eclipse. EMF codegen has three levels of code generation:

Model - Provides java interfaces and classes for all class in the model, plus factory and package (metadata) implementation class.

Adapter Classes - Generates implementation classes for editing and display.

Editor – provides structured editor that conforms to style for Eclipse EMF model editors and customization support is provided.

Graphical Editing Framework (GEF) provides a Model-View-Controller (MVC) framework for graphical editors. The GMF runtime provides a set of frameworks to assist in the development of Eclipse graphical editors using Eclipse Modeling Framework (EMF) and Graphical Editing Framework (GEF). GMF provides a separation of diagram and domain model. Although both can be persisted in a single file, the runtime provides for automatic persistence of all notational information (position of elements, color, font, and so on), requiring the Tool smith to provide only a domain model. GMF run- time provides the following:

A set of reusable diagramming components, such as action bars, connection handles, compartments, geometrical shapes, a diagramming toolbar, a set of diagramming actions, properties view, page setup and print preview, diagram export to image file, SVG support, border shapes, and system Clipboard support.

A standard notation model for storing diagram information separate from domain information A command infrastructure that bridges EMF and GEF

Extensibility options for the notation model, palette, diagram elements, layout, decorators, and domain model .A service provider infrastructure with priority and policy facilitiesFigure.1. describes main components in graphical modeling framework.

4. DESIGN OF VISUAL MODELING TOOL

Following steps are needed while Designing of visual modeling tool as described in Figure 2.

4.1 Functional Analysis of Visual Modeling Tool

4.1.1 Repository, Constraints and Process Modeling Tool

Designing of Graphical modeling tool that provides creation of Repository, applying Constraints on Repository and design of Processes for Master Data Management. Visual Modeling tool enables application developers to manage Repositories, Constraints and Process elements, and creating Repository, Constraint and Process definition by visual interface.

4.1.2 Developers Perspective towards Visual Modeling Tool

Developers working on Master Data Management can use Visual Modeling Tool and it can be extended to end users.

4.1.3 Engine Interface

Repositories, Constraints and Processes created by modeling tool are passed to engine interface to do simulation and optimization.

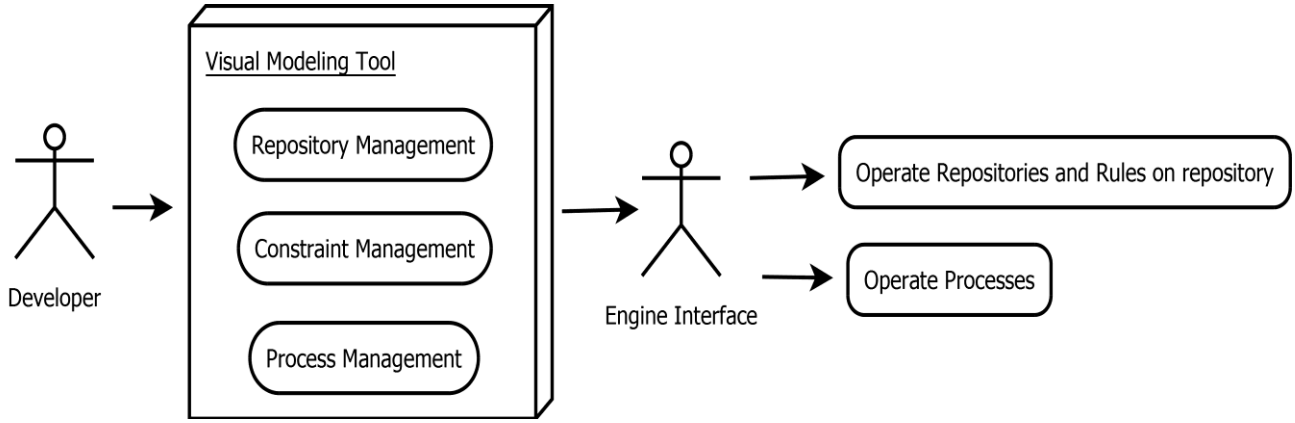

Fig 2: Case Tool for Visual Modeling Tool

In this paper, focus is on the visual modeling tool including Repository, Constraints and process object creation, editing, and the import and export features.

4.2 System Architecture

In order to make visual modeling tool able to integrate with other design modeling systems, followed design standards for Repositories, Constraints on repositories and Process definition in XPDL(XML Process Definition Language).Figure 3 shows software architecture of visual modeling tool that we researched in this paper. Modeling tool provides graphical user interface to user to visually define repositories, Constraints on repositories and processes, to save model respective storage format.

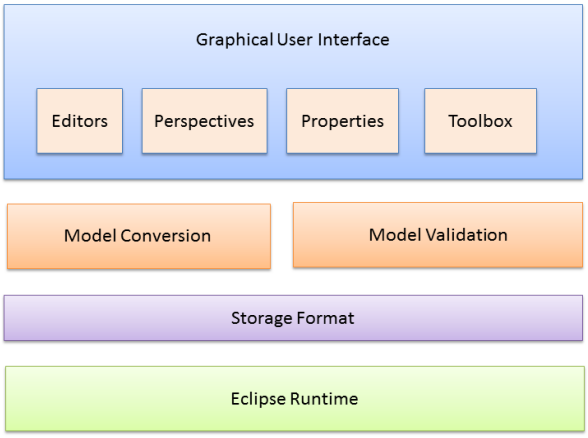

Fig 3: System Architecture of Visual Modeling Tool

Visual modeling tool supports model conversion and model validation. Eclipse runtime provides supports for running eclipse plug-in. In this paper, we mainly researched the visual Repository, constraint and process modeling tool. That is to say, we designed the graphical user interface, which is the top-level of Figure.3, and stored the model as the format we wanted in order to transfer to the next layer for model verification and validation.

4.3 Control Flow of Visual Modeling Tool

The flowchart of establishing a model when running the system is shown in Figure 4, 5 and 6. Figure.4. describes control flow of Repository Design. Repositories are designed using visual interface. After creating new Repositories Mouse events are handled and add relationship between these repositories. Properties of Repositories and Relationship defined. Figure.5. describes control flow of designing constraints on repository and set properties of constraint. Figure.6. describes flow of how processes are defined and how to set connection between processes. After establishing connection between processes properties of processes are defined. After defining processes, save process description in respective format.

In the model, view, and control layer, the entities of the visual modeling tool are defined as shown in Table 1.

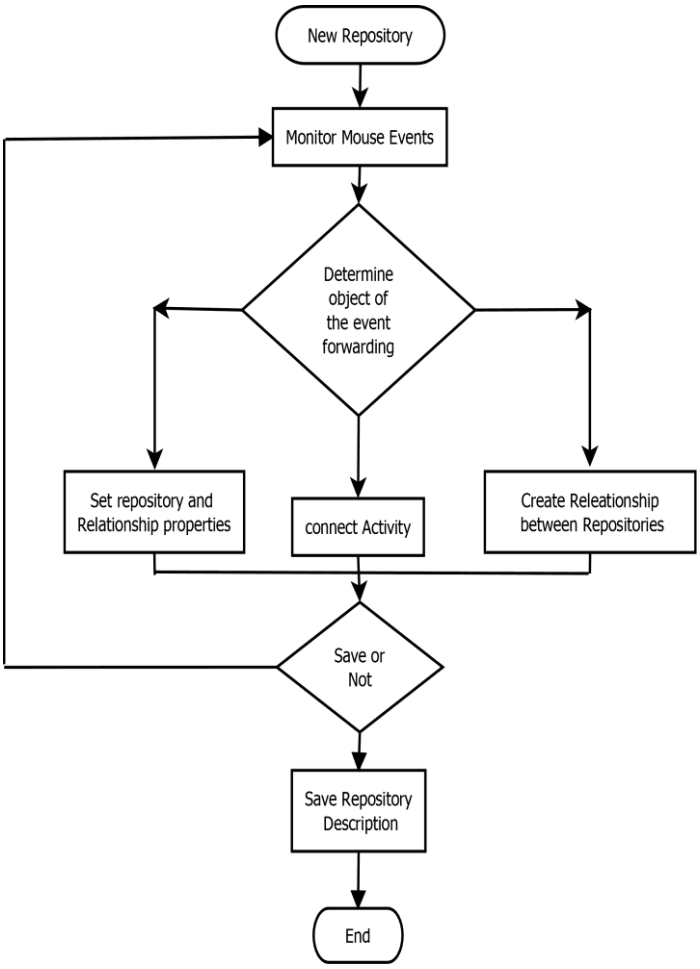

Fig 4: System operation flowchart for Repository Creation

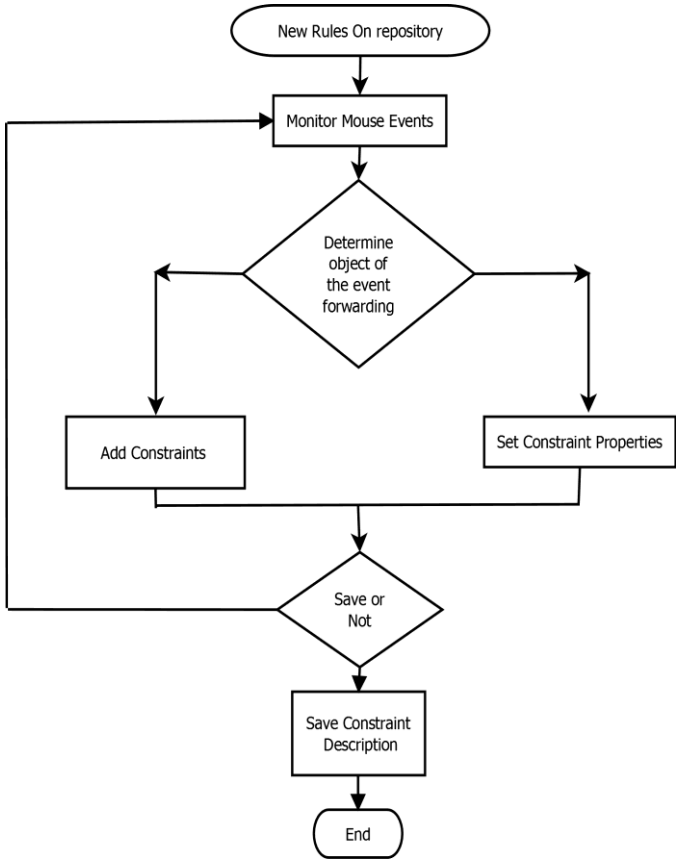

Fig.5: System operation flowchart for Constraint Design

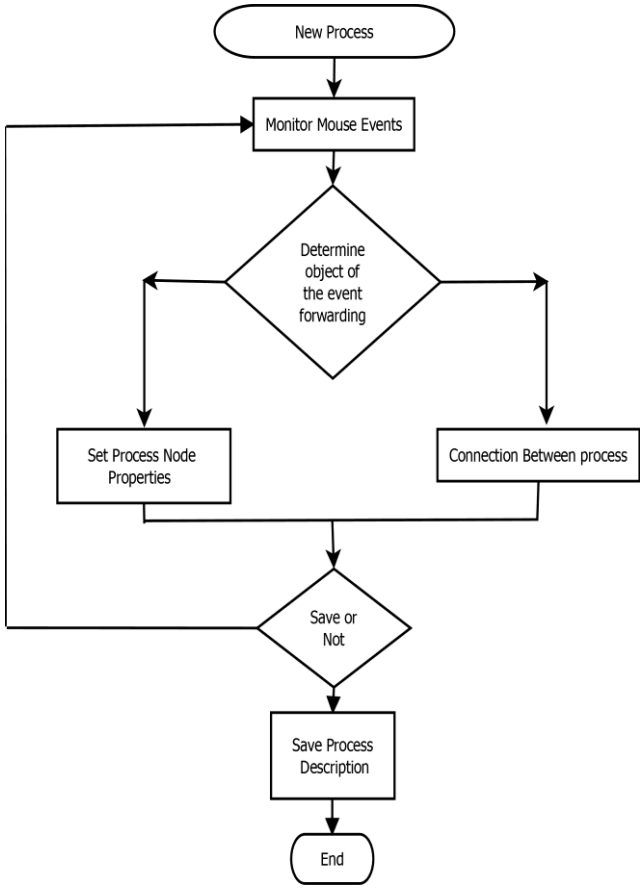

Fig.6: System operation flowchart for Process Design

Table.1. The Definition of the Entities of The Visual Modeling Tool in Model, View and Control Layers

Entity	Model	Control	View
Repository	Repository	Repository Editpart	Repository Figure
Relationship	Relationship	Relationship Editpart	Relationship Figure
Constraint	Constraint	Constraint Editpart	Constraint Figure
Activity	Activity	Activity Editpart	Activity Figure
Process	Process	Process Editpart	Process Figure
Start Node	Start	Start Editpart	Start Figure
End Node	End	End Editpart	End Figure
Transition	Transition	Transition Editpart	Transition Figure
Relationship Connector	Connector	Connector Editpart	Connector Figure

4.4 Entity Properties of Visual Modeling Tool

4.4.1 Repository Entity Properties

Repository Entity Definition contains Properties of repositories such as name, attributes and their properties etc and relationship properties such as name, attributes etc. The basic properties of repository definition are shown in Table 2.

Table.2. Basic Properties of Repository Definition

Repository Definition		
Property Name	Allowed to be Null	Description
ID_REPOSITORY	No	Repository ID
Name	No	Name of Repository
Description	Yes	Description of Repository
Repository Attributes Definition		
Name	No	Name of attributes

Type	No	Type of attribute
Length	Yes	Length of attribute
Multi Value	Yes	Multi value attribute
Partition key	Yes	Partition Key of Table
Description	Yes	Description of attribute
Relationship Definition		
Name	No	Name of Relationship
Description	Yes	Description of Relationship
Reverse_Name	No	Reverse name of Relationship
Source_repository	No	Source Repository for Relationship
Target_repository	No	Target Repository for Relationship
Multiplicity	No	Multiplicity of Relationship
Relationship Attribute		
Name	No	Name of Attribute
Type	No	Type of attribute
Length	Yes	Length of attribute
Description	Yes	Description of attribute

4.4.2 Constraint Entity Properties

Constraint Entity Definition contains properties of Constraint applied on repositories. The basic properties of constraint definition are shown in Table 3.

Table.3. Basic Properties of Constraint Definition

Constraint Repository		
Property Name	Allowed to be null	Description
Name	No	Name of Constraint
Type	No	Type of constraint used
Related Repository	No	Repository on which constraint applied
Related Relationship	Yes	Relationship on which constraint applied
Description	Yes	Description of Constraint

4.4.3 Process Entity Properties

Process definition contains all important process, activities, start node and end node. The start node definesa unique entry point of the process. The end node defines a unique export point of the process. There are multiple types of activity nodes. Process definition provides the context information for the other entities. For process definition certain activity parameters required. Table 4 describes properties of process definition.

Table.4. Basic Properties of Process Definition

Process Definition		
Property Name	Allowed to be null	Description
ID	No	Process Definition ID
Name	No	Process Name
Purpose	Yes	For which purpose process is used.
Process Definition Header		
Author	Yes	Creator of process definition
Creation time	No	Time of Process Creation
Description	Yes	Process Description
Duration	Yes	Life span of process.

4.4.4 Activity Node Entity Properties

Activity describes certain operation to be performed when process executes. Table.5 Describes properties of Activity used in Process operation.

Table.5. Basic Properties of Activity Node Definition

Activity		
Property Name	Allowed to be Null	Description
ID	No	Id of Activity
Name	No	Name of Activity
Description	Yes	Description of Activity
Action	No	Action to be performed

		by activity
Type	No	Type of activity (routing activity or process/action)
Input parameters	No	Activity Input Parameters.
Output parameters	No	Activity output parameters

4.4.5 Connection Arc Entity Properties

In a process definition, the connection arc of visual modeling tool connects the activity nodes; the connection arc describes the condition of whether the transition occurred or not during the running of process. Connection arc properties are described in table 6.

Table.6. Basic Properties of Connection Arc Definition

Connection Arc		
Property Name	Allowed to be Null	Description
ID	No	Arc Id
Name	No	Name of Arc
Description	Yes	Description of arc
Source Node	No	Id of source node
Target Node	No	Id for target node
Type_of_arc	No	Type of arc
Con_Expr	No	Condition expression

4.5 Function Design of Visual Repository, Constraint and Process Modeling Tool

While making function module decomposition functions that to be supported by visual modeling tool is as follows:

1. Graphics generation for repositories, constraint and processes.
2. Selection of graphics.
3. Moving of graphics.
4. Deleting of graphics.
5. Changing size of graphics.
6. Storing model elements based o xml.
7. Providing import/export feature of model.

5. CONCLUSION

Based on GMF technology, design of visual modeling tool described in this paper. The specific approach is to construct visual modeling tool for Eclipse plugins and master data management which will help in designing of Repositories, applying constraints on repositories and designing of processes to perform operations on data. Hence, realized how repositories should be defined, how constraints on repositories should be applied and how process should be designed through eclipse modeling framework. Further work is implementation of visual modeling tool and simulation and validation of visual modeling tool in master data management environment.

6. REFERENCES

[1] Lei Wang, Xinguo Ming, Jing You, "The Steps and Methodology of Identifying Master Data from Business Processes," *World Congress on Software Engineering*, pp. 329–333, IEEE 2009.

[2] W. Sadiq, and A. M. Orlowsk, "Analyzing process models using graph reduction techniques," *Information Systems*, vol. 25, no. 2, p. 117-134, 2000.

[3] W. M. P. Vander Aalst, "The application of Petri nets to workflow management," *The Journal of Circuits, Systems and Computers*, vol. 8, no. 1, pp. 21-66, 1998.

[4] R. Endl, G. Knolmayer, M. Pfahrer, "Modling processes and workflows by business rules,*" Proceedings of the 1 st European Workshop on Workflow and Process Management*, Zurich, Switzerland, 1998, pp. 47-56.

[5] H. Davu\cu, M. Kifer, and C. Ramakrishnan, "Logic based modeling and analysis of workflows," *Proceedings of the ACM Symposium in PODS'98, Seattle*, USA, 1998, pp. 25-33.

[6] Y. S. Fan, and C. Wu, "Research on a workflow modeling method to improve system flexibility," *Journal of Software*, vol. 13, no. 4, pp. 833-839, 2002.

[7] Thorsten Arendta, Florian Mantzb, Gabriele Taentzera, "EMF Refactor: Specification and Application of Model Refactorings within the Eclipse Modeling Framework," *Elsevier*, November, 2011.

[8] Jean Bézivin, Christian Brunette, Régis Chevrel, "Bridging generic modeling Environment and Eclipse modeling Framework," *IRINA & LINA*, 2010.

[9] Jean Bézivin, Guillaume Hillairet, Frédéric Jouault, "Bridging the MS/DSL Tools and the Eclipse Modeling Framework," *ATLAS Group*, 2010.