MODELING AND CONSTRUCTING A METADATA REPOSITORY FOR WFMC'S WORKFLOW MINIMUM MODEL USING OMG/MOF

André Morais Riccioppo and Juan Manuel Adán Coello
[1]*Serviço Federal de Processamento de Dados – SERPRO, andre.riccioppo@serpro.gov.br*
[2]*PUC-Campinas/UniSantos, juan@puc-campinas.edu.br*

Abstract: Workflow Management Systems have been increasingly used in companies to automate organizational processes. To allow the integration of those systems, the Workflow Management Coalition (WfMC) defined the Workflow Reference Model. The Object Management Group (OMG), looking for the standardization of workflow management systems in the context of its object-oriented architecture, had also defined a standard based on WfMC's standards, called the Workflow Management Facility (WMF). However, process definitions, that are the representation of the organizational processes, have not been standardized by OMG yet, besides its major role in the context of workflows. In this paper it is described the representation of WfMC’s model to represent process definitions, called the Minimum Metamodel, using UML diagrams. From this representation, using the dMOF tool, it is constructed a repository of metadata that represents the minimum metamodel according to OMG’s Meta Object Facility (MOF), the standard for representing metamodels in the context of OMG’s OMA architecture. Finally, it is presented an representative business example that was modeled and inserted on the repository to verify its functionality. It is argued on the paper that this approach decreases the complexity of workflow management systems and allows a more efficient use and reuse of process definitions.

Key words: Process definitions, Workflow Management Systems, Metadata Repository, WfMC Minimum Metamodel, OMG/MOF.

1. INTRODUCTION

The trend towards a global market is increasing the competition among enterprises, making critical the search for efficiency and efficacy on organizational processes management. In this context, increases the importance of workflow management systems, defined by the Workflow Management Coalition [1] [2], as "A system that completely defines, manages and executes “workflows” through the execution of software whose order of execution is driven by a computer representation of the workflow logic".

Workflows management includes: 1) business process modeling; 2) workflow specification e 3) workflow implementation using Information Technology resources.

The existence of several incompatible workflow management systems on the market lead to the creation of the Workflow Management Coalition (WfMC), an international organization involving users, vendors and analysts of workflow products, and industrial and academic researchers, with the objective of developing standards for workflow product implementation. Those specifications aim at permitting the interoperability between heterogeneous workflow products and simplify the integration of workflow applications with other Information Technology services. Those specifications are contained in the Workflow Reference Model [1], currently the most comprehensive attempt to describe and standardize the elements that constitute a workflow management system.

The WfMC identified that, at the highest level, workflow management systems provide support in three functional areas: "the Build-time functions, concerned with defining, and possibly modeling, the workflow process and its constituent activities; the Run-time control functions concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process; and the Run-time interactions with human users and IT application tools for processing the various activity steps."

The Workflow Reference Model identifies five functional interfaces between the major components of an workflow architecture and the Workflow Enactment Service: an interface with modeling and definition tools (Interface 1); an interface with client applications (Interface 2); an interface that supports the interaction with several IT applications (Interface 3), an interface that supports the interaction between heterogeneous workflow enactment services (Interface 4); and an interface with management tools (Interface 5).

Another organization that turned its attention to workflow management was the Object Management Group (OMG). Its work on the subject is reflected on the specification of a Workflow Management Facility (WMF) [3], a Common Facility based on WfMC specifications that defines the interfaces and required semantics to manipulate and execute objects that represent workflows. It specifies a set of interfaces for executing workflows. Because the WMF is part of the OMG Reference Model [10], its interfaces are represented in IDL. The WMF specifies the requirements for the interoperability between different implementations of workflow objects in a CORBA environment.

Although several aspects related to the execution of a workflow depend on the definition of process, for instance, which activities and participants will be represented on a process instance and on which conditions the activities will be executed, this subject was not addressed on the Workflow Management Facility. This happened, in part, because the *WfMC* work related to the process definition was not completed when the WMF was specified, and, in part, to limit the scope of that first specification, permitting to work it on more detail. There is ongoing work for creating specifications that define how to make those process definitions.

Currently, the WfMC specification that deals with process definitions is already concluded, and is part of Interface 1. This interface includes a metamodel that describes a process definition, a textual grammar for the interchange of process definitions, the Workflow Process Definition Language (WPDL), a set of APIs for manipulating process definition data in execution environments based on WfMC's Reference Model [4] and a schema for interchange of process definitions, the XML Process Definition Language (XPDL) [11].

On Interface 1 it is defined the Minimum Metamodel that identifies an extensible set of objects and attributes sufficient to support common process definition characteristics.

A process definition identifies a set of concepts that specify a workflow, for example a task at its decomposition, and data and control flow constructs. The set of those specification constructs supported by a workflow management system constitutes its workflow metamodel. This metamodel determines if a workflow management system will be able to model a given business process, because only the aspects that can be modeled by the standard metamodel can be instantiated during a workflow execution.

It can be concluded that it is important to have a standard form to interchange process definitions between applications that access or modify them, enabling to share tools and simplifying the exchange of process definitions between workflow execution environments.

OMG defines a Common Facility called the Meta Object Facility (MOF) [5] that provides a set of CORBA interfaces that can be used to define and manipulate metamodels, in order to improve the management and interoperability of metadata in distributed object environments in general, and in distributed development environments in particular.

The main objective of the following sections of this paper is to present the construction of a repository of metadata of process definitions done in accordance with the Minimum Metamodel and compatible with existing OMG standards. On section 2 it is presented the Meta Object Facility and the MOF Model, defined as the meta-metamodel on that specification, and discussed its relation with WfMC's Minimum Metamodel. On section 3 the Minimum Metamodel is represented using UML diagrams. From that representation, on section 4, it is constructed a repository of metadata for process definitions using the dMOF tool and is outlined an example that uses the repository. Finally, on section 5, some final remarks are made.

2. THE MOF MODEL AND THE MINIMUM METAMODEL

On [5], OMG uses a four layer framework to represent metadata. As shown on table I, the layers are described as follows:

1. User object layer, or M0, comprised of the information that we wish to describe, typically referred as data;
2. Model layer, or M1, comprised of the metadata that describes information, informally aggregated as models;
3. Metamodel layer, or M2, comprised of the descriptions (meta-metadata) that define the structure and semantics of metadata. A metamodel can be seen as a language for describing different kinds of data;
4. The meta-metamodel layer, or M3, comprised of the description of the structure and semantics of meta-metadata. The meta-metamodel can be seen as a language for defining different kinds of meta-data.

Table I show also how MOF and the concepts of workflow can be seen on the four-layer metadata architecture. Workflow instances are placed on the user object layer, process definitions on the model layer, WfMC's Minimum Metamodel on the metamodel layer, and MOF model on the meta-metamodel layer.

Table 1. Workflow concepts and MOF on the four-layer metadata framework

M3	Meta-metamodel	MOF Model
M2	Metamodel	WfMC’s Minimum Metamodel
M1	Model	Process definitions
M0	User Objects	Workflow Instances

The Meta Object Facility (MOF) specification [5] provides a set of CORBA interfaces that can be used for metamodel definition and manipulation. MOF is used to manipulated metaobjects in order to permit the integration of tools and applications, integrating metamodels from various domains. This is done by means of the definition of a simple meta-metamodel with a semantics sufficient for describing those metamodels from several domains.

The MOF metadata architecture has some features that distinguish it from earlier metamodeling architectures.

– The MOF model is object-oriented, supporting meta-modeling constructs that are aligned with UML's constructs [6].
– The MOF model is self-defining, that is it is formally defined using its own meta-modeling constructs.

The main MOF modeling concepts use terms that are common with UML, for example, a MOF Class corresponds to a UML Class. However, the correspondence is not always direct. For example, UML Associations may have many AssociationEnds, but MOF Associations must have precisely two.

MOF uses an object oriented framework that is essentially a subset of the UML core. The four main modeling concepts are: Classes, which model MOF metaobjects; Associations, which model binary relationships between metaobjects; DataTypes, which model other data (e.g. primitive types, external types, etc.); Packages, which modularize the models.

3. REPRESENTING THE MINIMUM METAMODEL USING UML

WfMC's Minimum Metamodel, shown on figure 1, identifies the entities usually used on a process definition, some attributes that describe the characteristics of those entities and the relationship between those entities. Some of the elements on the Mininum Metamodel are mandatory and some of then are optional, although usually found on workflow management systems. The basic metamodel can be extended using "extended attributes" to meet the specific needs of particular products or workflow systems. The entities System & Environmental Data and Organizational Model may be accessed by a process definition, but are not included on that definition.

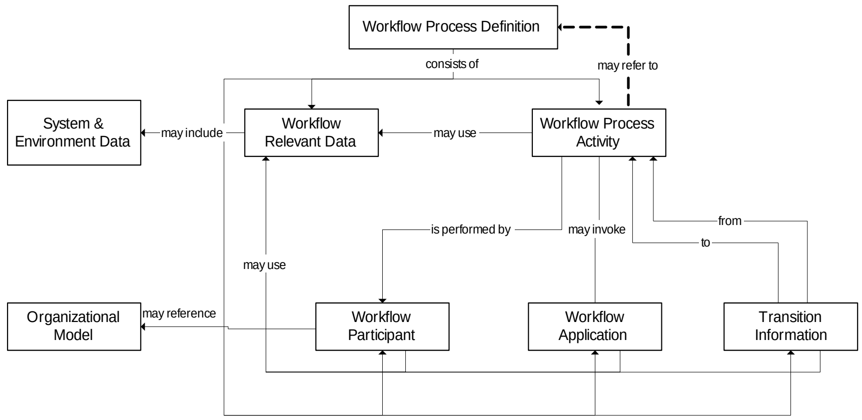

Figure 1. WfMC's Minimum Metamodel Top Level Entities

The Minimum Metamodel is described on [8] by text, diagrams and tables. In this section this metamodel will be represented using UML class diagrams. From this representation a MODL (Meta Object Definition Language) description will be used to build a metadata repository by means of dMOF, an implementation of MOF that provides tools for creating and managing MOF meta-models and generates stand-alone metadata repositories called moflets [7]. Due to space limitations, on this section, only some representative entities from de minimum model will be presented. The full set of descriptions can be found elsewhere [12].

Class Entity

All metamodel entities share the attributes identifier, name and description. To avoid the replication of those attributes on all entities, they are grouped on an abstract class called Entity, shown on figure 2.

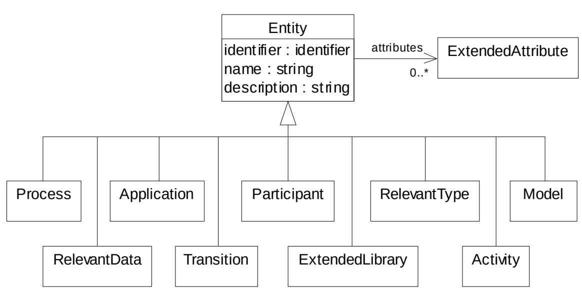

Figure 2. Class Entity

Workflow Process Definition

Class Process (fig. 3) represents a Workflow Process Definition, that defines the entities that make up a workflow: activities, transitions, applications, participants and relevant data. A process definition may reference those entities on other process definitions. A process definition may also include attributes to store important data for administration purposes, for example, author and version , for runtime control, for example, priority, and for simulation.

A Workflow Process may run as a sub-process invoked as an implementation of an Activity of type Subflow; in this case parameters may be defined as attributes of the process.

Besides the attributes above, it is also worth mentioning the following:

– Responsible: a workflow participant that is *Responsible* for this workflow process.
– Restrict_to: list of identifiers of workflow relevant data defined in the surrounding process Model definition.
– Extended library: allows the declaration of library functions and procedures.

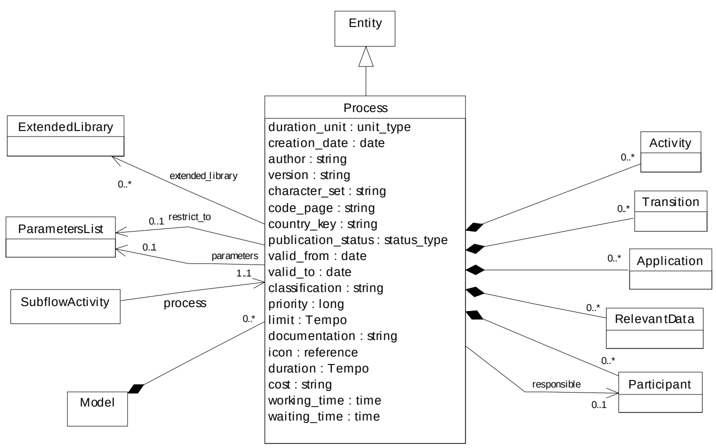

Figure 3. Class Process

Workflow Process Activity

Class Activity (fig. 4) is used to define each elementary activity that makes up a workflow process. Attributes may specify control information, implementation alternatives, Performer assignment, runtime relevant information and data used specifically in Business Process Reengineering (BPR) and simulation situations. In addition, restrictions on data access and transition evaluation (e.g. Split and Join) can be described.

Activities can be of two kinds Route or Implementation. A Route Activity is a "dummy" Activity that permits the expression of cascading transition conditions (e.g. of the type "IF condition-1 THEN TO Activity-1 ELSE IF condition-2 THEN TO Activity-2 ELSE Activity-3 ENDIF").

If an activity is not of type Route, it can be implemented in one of four ways: No, Application, Subflow and Loop. No activities are implemented by manual procedures (i.e. not supported by workflow); Application activities are implemented by one or more applications; Subflow activities are implemented by a subprocess; and Loop activities are implemented by a loop of other activities, connected by specific loop transitions.

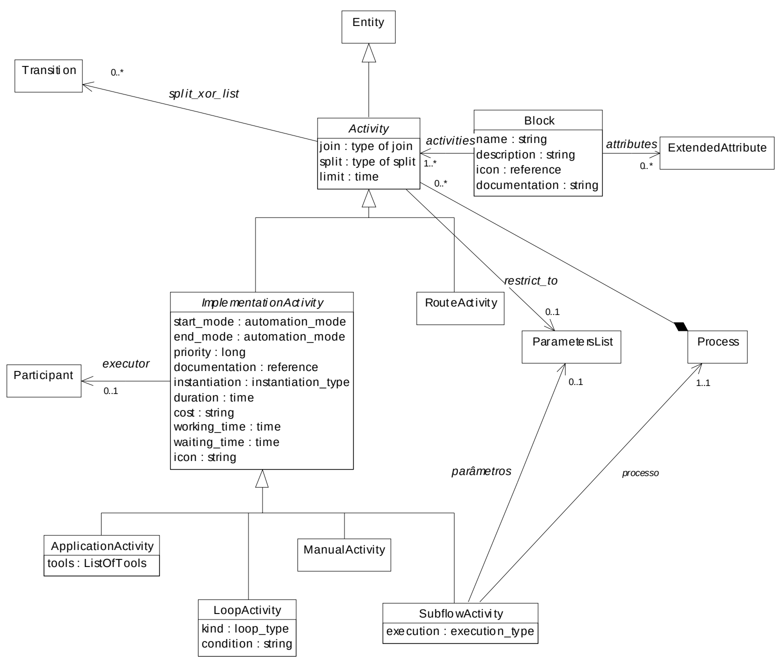

Figure 4. Class Activity

Transition Information

Activities are related to each other by control flow conditions, represented by transitions. Each transition has three elementary properties: the activity that is performed before, the activity that is performed after and the condition under which the transition is done. Class Transition, which represents Transition information is presented on figure 5.

Transitions can be of two kinds: Regular and Loop-connecting. For "regular" Transitions it is possible to define or synchronize multiple (concurrent or alternative) control threads (SPLIT, JOIN) and sequences of Transitions between Activities (cascading Transitions/conditions) and Blocking restrictions. Loop-connecting Transitions allow the expression of cycles in the Transition network. They connect the body of a Loop with the Loop Activity that is implemented by this body. Loop conditions are expressed in the loop Activity, not as Transition conditions.

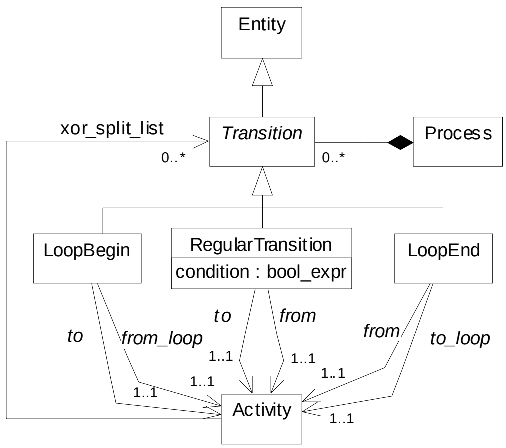

Figure 5. Class Transition

Workflow Participant Specification

Class Participant (fig. 6) permits to describe activity performers. Four participant types are defined: an organizational unit, a human, a role, and a resource. A role and a resource are used in the sense of abstract actors. During run time these abstract definitions are evaluated and assigned to concrete human(s) and/or program(s).

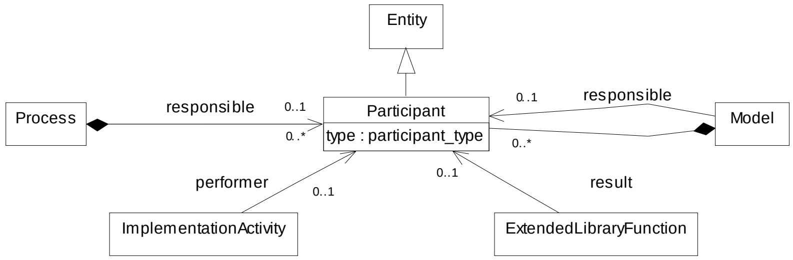

Figure 6. Class Participant

Workflow Model

The Minimum Metamodel includes various entities whose scope may be wider than a single process definition (in particular the definitions of participants, applications and relevant data). The concept of a model is introduced, which acts as a container for the grouping of common data entities, to avoid redefinition within each individual process definition. Figure 7 shows class Model, used to represent a Workflow model.

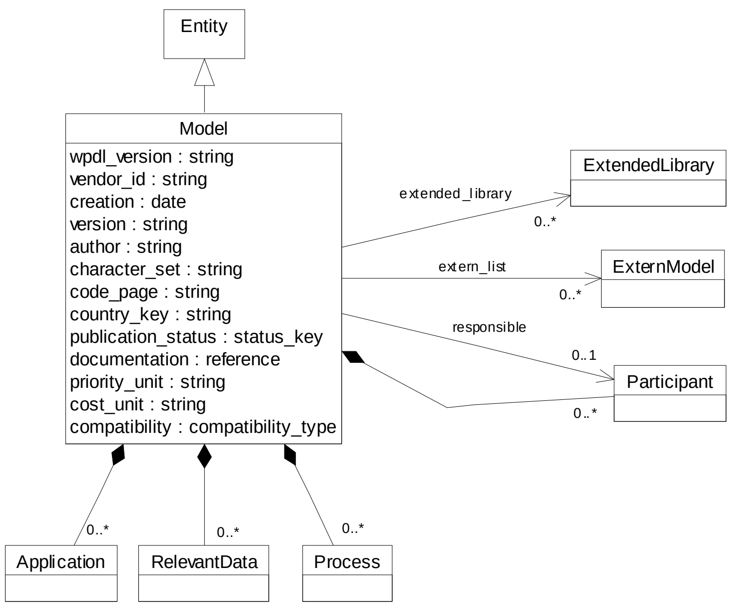

Figure 7. Class Model

4. CONSTRUCTING A REPOSITORY OF METADATA FOR THE MINIMUM METAMODEL

From the representation of the Minimum Metamodel presented on previous section, a repository of metadata for the Minimum Metamodel will be created using the dMOF tool [7].

The *dMOF* tool generates Java classes, that implement a metadata repository compatible with MOF, called a *moflet*. *dMOF* comprises:

– A repository of metamodels compatible with MOF;
– The *MODL (Meta Object Definition Language)*, that permits to specify MOF metamodels;
– A MODL compiler that loads MODL specifications of a metamodel in the repository of MOF metamodels;
– A IDL generator that produces the *CORBA IDL* compatible with MOF for MOF metamodels;
– a generator of *moflets*.

After a moflet is produced, the generated IDL can be compiled, for example using *Inprise Visibroker for Java,* in order to produce the CORBA *stubs* and *skeletons CORBA* necessary for compiling the *moflet*.

When the *moflet* is executed a new instance of the repository of metadata for the Minimum Model is instantiated, where process definitions can be inserted and queried. Once the workflow engines use this repository as a central point to query the process definitions, it becomes easier to distribute workflows between workflow management systems and to allow interoperation of workflows at execution time. With specific tools that interoperate with the repository, it becomes easier to modify the process definitions at run time, when the workflow engine supports ad-hoc workflows.

Using the Repository: The Representative Business Example

To validate and exercise the functionality of the repository, the *Representative Business Example* [8] was modeled and inserted and queried in the repository using two Java programs: MetaModelDefinition and MetaModelQuery. Program MetaModelDefinition inserts into the repository the process definitions for the *Representative Business Example*, that uses most entities on the Minimum Metamodel, and program MetaModelQuery performs queries on the repository.

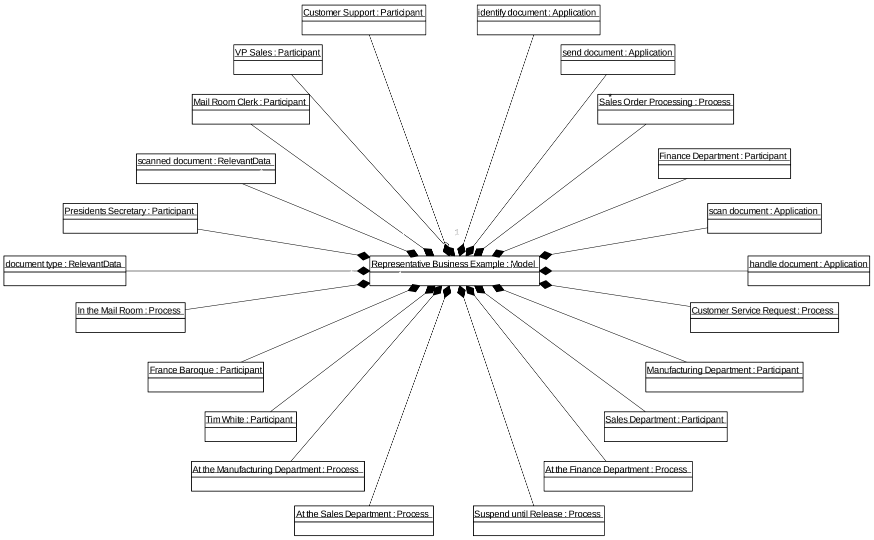

Figure 8. Model Representative Business Example

MetaModelDefinition is responsible for the instantiation of the classes defined during the creation of the repository and for defining the relationships between the instances using the interfaces generated by *dMOF*.

The *Representative Business Example* is informally defined as follows in [8]:"The FBN Sports Equipment Company, located in Luxembourg, manufactures a complete range of soccer, baseball, tennis and general athletic equipment. They only sell to European resellers, major sports outlets and North American Distributors. All sales are made by way of Purchase Orders. The company has grown from a small organization to the point where response to their customers is getting longer and longer. They have decided to employ a workflow management system in order to improve their business processes. In the existing environment all mail comes into the mail room and is then distributed. Since the company receives 80% of its sales_order activity by mail and FAX, it has been determined to have an image scanner in the mail room as one of the methods to enhance the workflow system."

The top level of the Model *Representative Business Example*, inserted on the repository by MetaModelDefinition, is shown in figure 8.

Figure 9 shows some detail of "In the Mail Room", one of the processes that comprises the Model Representative Business Example. This business process involves giving a mandate to the mail room clerk to review all mail which will be scanned into the system and then to make decisions as to which routing process the mail documents should take. Only the mail to the President or marked "personal" will be sent directly and unopened to the person.

Program MetaModelQuery makes some queries to the repository using the interfaces generated by dMOF, for example, the amount of Models and Processes instantiated on the repository, which processes are part of a given Model and which activities belong to a certain process.

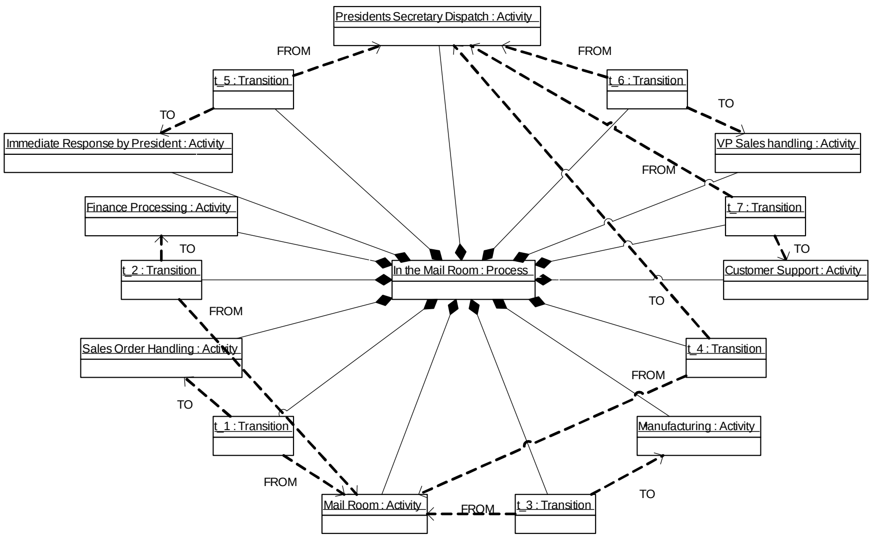

Figure 9. In the Mail Room Process

5. CONCLUDING REMARKS

On this paper it was discussed the representation of the elements from WfMC's Minimum Metamodel using UML class diagrams. From this representation, with the aid of the dMOF tool, it was constructed a repository compatible with OMG standards to store metadata related to process definitions made in accordance with the Minimum Metamodel.

To exercise the repository functionality two programs were developed. The first inserts into the repository a representative business example and, the second, queries the repository. The development and use of both programs showed good results, but evidenced the importance of having user-friendly tools for process modeling and management.

A metadata repository, as the implemented, makes easier the distribution of workflows between workflow management systems, the interoperation of workflows at execution time to support changes on process definitions at run-time in management systems that permit ad-hoc workflows.

The repository can be seen as the central element around which can be developed a rich set of tools, for instance, graphical tools for process modeling and deployment that can be integrated into a CORBA compliant services platform [9].

6. REFERENCES

[1] Workflow Management Coalition, The Workflow Reference Model, WfMC Document WfMC-TC-1003, 1995.

[2] Workflow Management Coalition, Workflow Management Coalition – Terminology & Glossary – Issue 3.0, WfMC Document WfMC-TC-1011, 1999.

[3] Object Management Group, Workflow Management Facility Specification – V. 1.2, OMG Document ftp://ftp.omg.org/pub/doc/formal/00-05-02, 2000.

[4] Workflow Management Coalition, Interface 1: Process Definition Interchange – Version 1.1, WfMC Document WfMC-TC-1016-P, 1999.

[5] Object Management Group, Meta Object Facility (MOF) Specification – Version 1.3, OMG Document ftp://ftp.omg.org/pub/doc/formal/00-04-03, 2000.

[6] Object Management Group, Unified Modeling Language (UML) – Version 1.4, OMG Document ftp://ftp.omg.org/pub/doc/formal/01-09-67, 2001.

[7] Distributed Systems Technology Centre, dMOF 1.0.1 – User Guide, http://www.dstc.edu.au/dMOF.

[8] Workflow Management Coalition, Interface 1: Process Definition Interchange Q&A and Examples, WfMC Document WfMC-TC-1016-X, 1999.

[9] Silva C., Soto J., Mendes M., Integrating a Workflow Engine and a MOF Repository to an Open Service Platform, 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises, 2002.

[10] Object Management Group, The Common Object Request Broker: Architecture e Specification, OMG Document ftp://ftp.omg.org/pub/doc/formal/01-12-35, 2001.

[11] Workflow Management Coalition, Workflow Process Definition Interface – XML Process Definition Language – Version 1.0, WfMC Document WfMC-TC-1025, 2002.

[12] Riccioppo, A. M., Implementation of a workflow process definition repository based on the Meta Object Facility, Master Dissertation (in portuguese), PUC-Campinas, june 2002.