Exploring Generic Identity Management Processes

An approach to model the generic Identity and Access Management Process "approve-request"

by

Arslan Brömme	Andreas Netzer	Dr. Horst Walther
ConSecur GmbH,	iC Compas GmbH & Co. KG,	NIFIS e.V.,
Meppen, Germany	Pfaffenhofen, Germany	Frankfurt, Germany
broemme@consecur.de	*netzer@ic-compas.de*	*horst.walther@nifis.org*

Position paper on behalf of the NIFIS competence centre Identity Management (GenericIAM).

Presented at

GenericIAM 7th Quarterly Meeting,

29th June 2007,

Munich, Germany

Validated by

Dr. Dieter Coldewey (ConSecur GmbH)
Dr. Angelika Steinacker (CSC Deutschland Solutions GmbH),
Markus Vogel (KPMG Deutsche Treuhand-Gesellschaft)

Abstract:

In this paper we present the NIFIS approach for the development of generic identity and access management (GenericIAM) processes based on a variant of a state transition model (coloured Petri nets). By considering the interactions and state transitions of the fundamental objects involved in the Identity Management and the generic subject acting on them we were able to generate an adequate abstract model of the first GenericIAM core process “approve-request”, which is presented here for further discussion. Our approach intends to complement the bottom-up modelling approach of factoring out generic patterns out of an empirical base of flow oriented diagrams for IAM processes, which have been delivered for this purpose.

1 Introduction

The definition of processes for the Identity & Access Management (IAM)[1] causes major effort according to our experience and the reports of the main analysts.

Although most corporations regard their processes as unique and individually tailored, a core set of standard processes remains remarkably stable over the majority of examples. Obviously considerable similarities between the processes of different corporations exist.

This situation raises the questions: Why do we always start with a blank sheet of paper? Why "reinvent the wheel" again and again? Shouldn't we instead focus our efforts on the obvious differences and use the common set of standard processes "off the shelf"?

The NIFIS[2] initiative "GenericIAM" (Generic processes for the Identity & Access Management) was set up with the mission to extract a generic IAM process model from existing IAM processes implemented in major corporations.

However we found that even for the most experienced process modelling expert's abstraction and documentation of generic commonalities from enterprise specific solutions following a bottom-up approach turned out to be remarkably difficult.

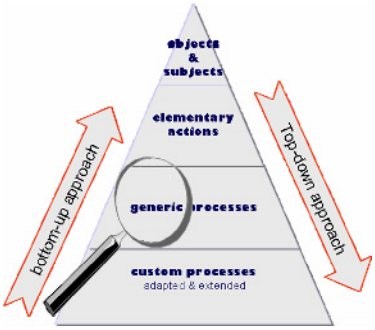


figure 1

Based on the assumption, that the IAM processes of an enterprise could completely be described by the actions of a limited and manageable number of subjects (actors) on an equally limited number of objects (figure 1), we herewith try to derive a generic model following a seven-step top-down approach.

Hereby in a first step identify the fundamental objects, which are involved in IAM processes. Seconded we detect the derived objects, which describe the relationships of the fundamental objects. Third we identify the subjects (actors) who operate on the objects. In a fourth step we name the elementary actions which express the actions of the subjects on the objects, the interactions ob the objects or which perform object state transitions. In a fifth step business events are detected as triggers for processes and sixth essential processes are formed by combining the elementary actions to net of lows resulting an a meaningful result in business terms. In a seventh and last optional step the essential processes are complemented by physical actions (check-, translation- and transport-steps) in order to cope with imperfections of existing implementations.

The intention of this documentation is to demonstrate, how the top-down- and the bottom-up approach combine seamlessly to a self-contained and consistent model.

2 Context

Our move towards a more standardised approach to set-up organisational processes fits nicely in two major trends to be observed in the general management context:

- business driven identity management

[1] Identity and access management combines processes, technologies, and policies to manage digital identities and specify how digital identities are used to access resources.

[2] Nationale Initiative für Informations- und Internet-Sicherheit (NIFIS e.V., http://www.nifis.de/)

- industrialisation of services

Although the necessity for some kind of identity and access management reaches far back it is regarded as a coherent and consistent discipline only recently [Windley, 2005]. As computers were used in the past by specialists only, the IAM tasks were delegated to technical administrators. Since computer usage has become the mainstream toolset for any business, identity management tasks received acceptance as genuine management responsibility [Stuart, 1999] – yet with a strong technical component.

The second trend has two major drivers: at first, the enterprises need to prove compliance with some regulatory requirements (e.g. Sarbanes Oxley Act[3] and the upcoming "EuroSOX"[4]), at second, the necessity to meet the challenges of global competition. Both drivers result in a more industrial perception of the enterprises as formal systems. By applying standard governance models (e.g. CobiT[5]), best practice models (e.g. ITIL[6]) or generic process models (e.g. GenericIAM) it is expected to reduce costs through standardisation and simultaneously ease the job of proving compliance while focusing on the core competencies of the business.

3 Approach

In this chapter we try to identify the fundamental objects, which are involved in IAM processes and the derived objects, which describe the relationships of the fundamental objects. Next we identify the subjects (actors) who operate on the objects. The intention is to detect the elementary activities by this se-up which express the actions of the subjects on the objects, the interactions ob the objects or which perform object state transitions.

3.1 Identity

The fundamental concept of identity management is the digital identity. In this context digital identity is defined as a minimal set of information (attributes) necessary to unambiguously identify an individual or a technical object.

When an individual enters the enterprise ecosystem the first time, its digital identity is created (figure 2), regardless whether it is a "user" of the enterprises resources not. Being a user indicates a specific relationship already: the usage of resources.

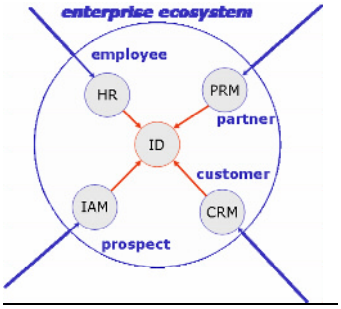


Figure 2

[3] The **Sarbanes-Oxley Act of 2002** is a United States federal law passed to enhance corporate transparency and responsibility [USA SOX, 2002].

[4] Directive 2006/43/EC of the European parliament and of the council of 17 May 2006 on statutory audits of annual accounts and consolidated accounts, amending Council Directives 78/660/EEC and 83/349/EEC and repealing Council Directive 84/253/EEC [EU DIR 2006/43/EC, 2006].

[5] The **Control Objectives for Information and related Technology** (**CobiT**) is a set of best practices for information technology management created by the Information Systems Audit and Control Association (ISACA), and the IT Governance Institute (ITGI) in 1992 [ITGI COBIT 4.1, 2007].

[6] The **Information Technology Infrastructure Library** (**ITIL®**) is a framework of best practice approaches intended to facilitate the delivery of high quality information technology (IT) services [OGC ITIL 2, 2005; OGC ITIL 3, 2007].

3.2 Objects of the corporation

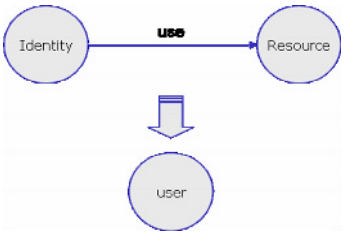


Figure 3

The digital identity is the individual's digital sibling. The lifetime of the corporation's interest in the individual determines the lifetime of its digital identity.

In an enterprise context identities normally "use" resources. They do so by performing operations. The relation "use" may carry attributes. It can be well expressed as a derived object: the user (figure 3).

If the right to use a resource of an organisation is granted to the identity, it has at least one relationship to the organisation. There are many specialisations of this relationship. Examples of the relationship are an employees' contract, the freelancer's contract, a partner- or a customer-contract. Obviously more than one such relationship may exist at the same time.

This attributed relationship can be expressed as an agreement or contract. For the purpose of organising the business it is useful to reduce the contract to the business role (or more of them). Using the generic Type-Instance-modelling pattern the instance (contract, role) defines incarnation details of the type (contract type, role type).

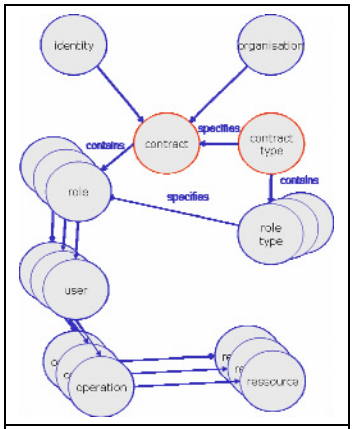


Figure 4

The operations an Identity may perform on the organisation's resources are defined in several intermediate organisational constructs (figure 4). The may be considered as the fine structure of the identities relationship to the organisation: A contract defines the total relationship, whereas the role represents the entitlements and the planned behaviour. The role hence represents a fraction of the contract and the contract may contain several roles.

For each role planned actions, the operations on objects are defined. In fact all non-ad hoc operations need to be predefined for each role. In a properly organised corporation roles rather than individuals operate on resources [Ferraiolo and Kuhn, 1992].

3.3 Subjects

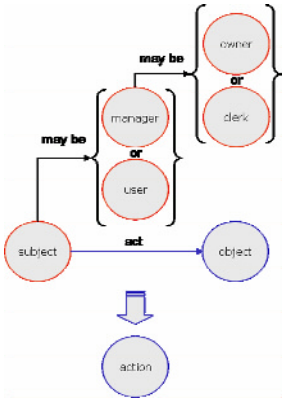


Figure 5

Next to the fundamental static objects in workflows we define subjects (actors) which act on these objects. Subjects may be managers or users. Users may be object owners or custodians or the time in case of time-triggered events. Owners are responsible for the objects. Custodians act on behalf of owners with a delegated and usually limited set of rights. Hence owners delegate rights to custodians. Typical custodians are system administrators (figure 5).

Subjects may act on their own intention or they may react on a request.

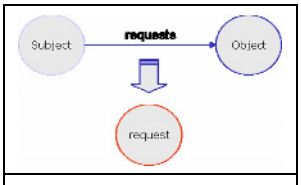


Figure 6

The subject's initial action triggers an event and instantiates the process. To given an example: the most important process instantiation is known as a request. Reactions often are approvals of requests. The request is a transient object. It is the central workflow object. It can be considered as the instantiation of a process type. The request is created by an

event when a subject requests access to an object, when time has come to re-validate a role / privilege or when the defined response period has been passed without an action (escalation).

3.4 Owners

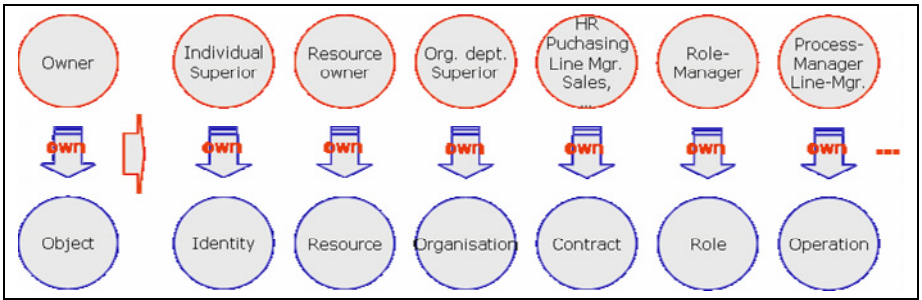


Figure 7

To identify the acting subjects we introduce the concept of object ownership. Each object has an owner. The ownership and its notation differ from one organisation to another. Hereby apparently a high complexity is introduced as the result of customising of a simple model.

We consider the owners of the objects to be the logical acting subjects to act on their objects.

3.5 Events

There are events created by a subject and time triggered events. Generally events can be understood as the occurrence of a change of an object's state.

3.6 Activities

Activities are elementary operations. By breaking down complex processes into smaller elements they can be identified when an atomic step is found, i.e. a step performed by one single subject at one location at a specific time. Elementary activities are identified by state transitions.

Example:

The approval of a request by one single approver is an elementary activity: it changes the status of the object 'request' from 'created' to 'approved'.

3.7 Processes

Processes are made out of one or several activities [Keller, Nüttgens, Scheer 1992]. They are triggered by an event and lead to a result meaningful to a subject. It is important to distinguish between the process type (the class or definition) and the process instantiation (incarnation, actual).

There are operational processes and managerial processes. There are only three – ideally one-step – operational IAM processes: *identification*, *authentication* and *authorisation*. The managerial processes are made up of three groups: *administrative processes*, *audit processes* and *change processes*[7]. The administrative processes represent the "lion's share" of all IAM processes. Its most prominent representative is the "***approve request***" process.

[7] While administrative processes maintain the parameters of the operational process level, audit processes monitor the operational all processes and raise alerts in case of conflicts with the corporation's policies & guidelines. Change processes are those which change the entire organisational set-up.

4 Notation

4.1 Basic notation

To express the processes in a specific notation, we are faced with the conflicting requirements that they need to be formally provable and understandable at the same time.

We have identified coloured Petri nets as a powerful and concise formalism for the description of complex and asynchronous IAM processes. Here coloured Petri nets are used to describe the states and transitions between states by a recursively structured generic process, consisting of similar, replicated basic state transitions.

Furthermore coloured Petri nets allow defining terminating states. Thus they are complete with respect to the requirement of termination, which is known from algorithms. Coloured Petri nets can be described in a precise mathematical fashion and can be used to prove certain state transition and invariants, and can be transformed for the usage in (automated) proofs of theorems.

4.2 Enhanced notation

The notation we use here covers a simple coloured Petri net with states, transitions, weighted edges between states and transitions, weighted edges between transitions and states, and a single colour for all markings[8].

We have enhanced the notation by adding general first order logics predicates as additional firing conditions and time-out triggering for the transitions (indicated by <pre-conditions>), labels and owners for different instantiations (indicated by <role.inst.>;<owner>) of the transitions, and (in braces) a cardinality as an additional condition showing the number of instantiated transitions (indicated by {n}) which must fire for further processing.

Graphically we symbolize a state by a circle, edges by lines with arrows showing the direction and numbers showing the weights, transitions by boxes with rounded corners and first order logic predicates filled in, labels, identifiers of owners, and cardinals in braces for the number of instantiated transitions to fire, and abstractions by boxes with sharp corners which can contain complete coloured Petri nets in a recursive fashion.

[8] For further information on (coloured) Petri nets, please refer to [Reisig, 1986; Jensen, 1997].

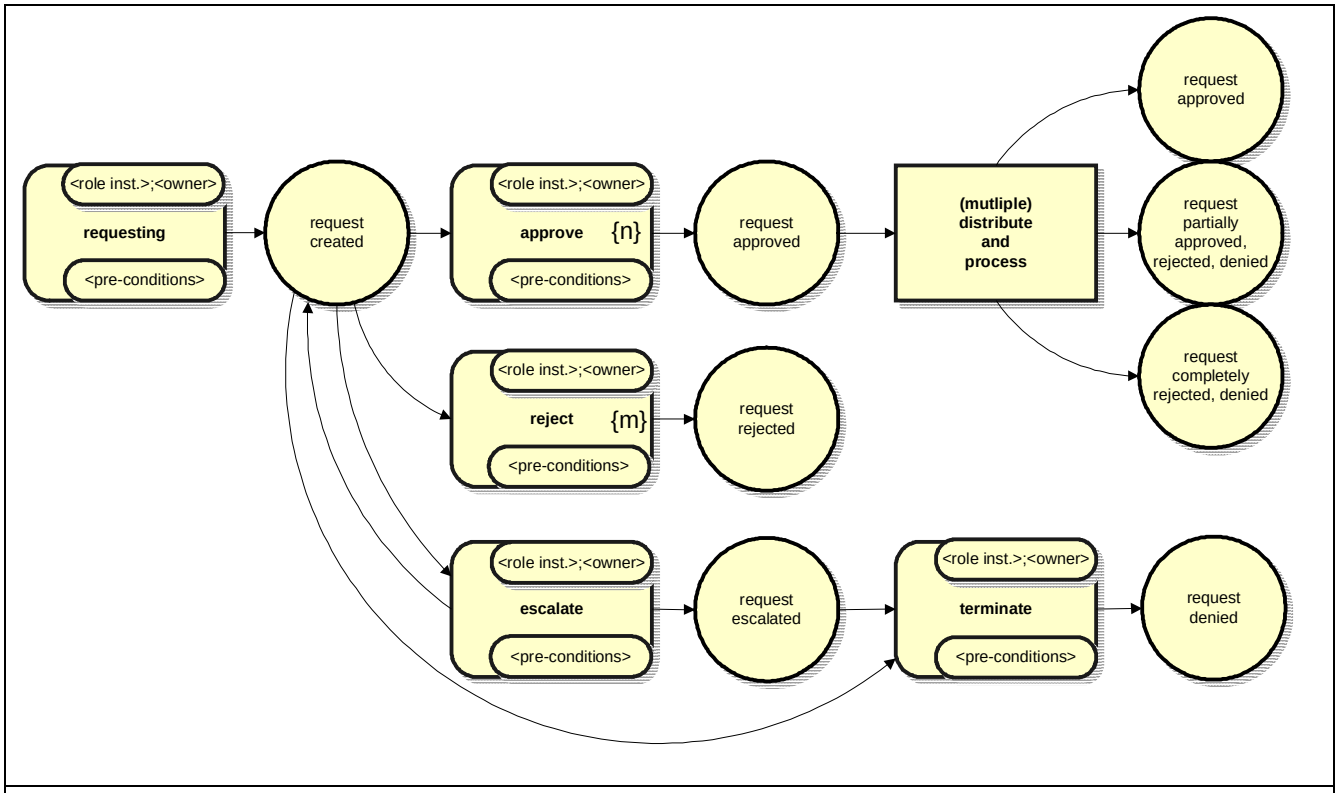


Figure 8: An Example process: GenericIAM approve-request

This process is started by generating a request within the transition “requesting”. After this initial activity, the overall process is in the state “request created”. From this state several possibilities exist for subsequent transitions to fire depending on the pre-conditions.

In the case of a reject the transition “reject” fires and the process terminates in the state “request rejected”.

Going back to the state of “request created” another possibility is an escalation due to the pre-condition of timeout because no other transition has fired within a time frame defined in the pre-conditions. If this happens the transition “escalate” fires and the process is in the state of “request escalated”.

By reaching the pre-condition for terminating the overall process the transition “terminate” deletes any pending resubmitted request and terminates. The process reaches the state “request denied”.

Going back again to the state “request created” the third case can be examined. If all “{n}” instances of a certain addressed role have approved a request the transition “approve” fires and the state “request approved” is reached.

From this state an abstract transition “(multiple) distributing and processing” is executed containing the GenericIAM approve-request process in a recursive manner and possibly multiple instantiations for subsequent detailed system approvals.

The three possible resulting states after execution of the abstract transition “(multiple) distribute and process” are “request approved”, “request partially approved, rejected, denied”, and “request completely rejected, denied”.

Within the abstract transition the single states of “request rejected” and “request denied” are monitored by transitions to propagate this information within the network for the overall process states of “request partially approved, rejected, denied” and “request completely rejected, denied”.

After this general description of the GenericIAM approve-request process the particular case of an escalation is given as a further example for a possible process execution. In figure xxx an employee requests a SAP role at his supervisor who is the holder of the role SAP supervisor to approve this specific request. Due to the absence of the supervisor a team member who is the delegate of the supervisor and holder of the role SAP supervisor, too, should decide about this requests. The delegate is not present within a given time frame, so the chief of the team, who holds additionally the role as a SAP supervisor, should decide as the last available instance in this hierarchy about the request. Instead of approving the request the chief decides to terminate this request for some reason and the escalation ends here. The employee has the chance to resubmit his request.

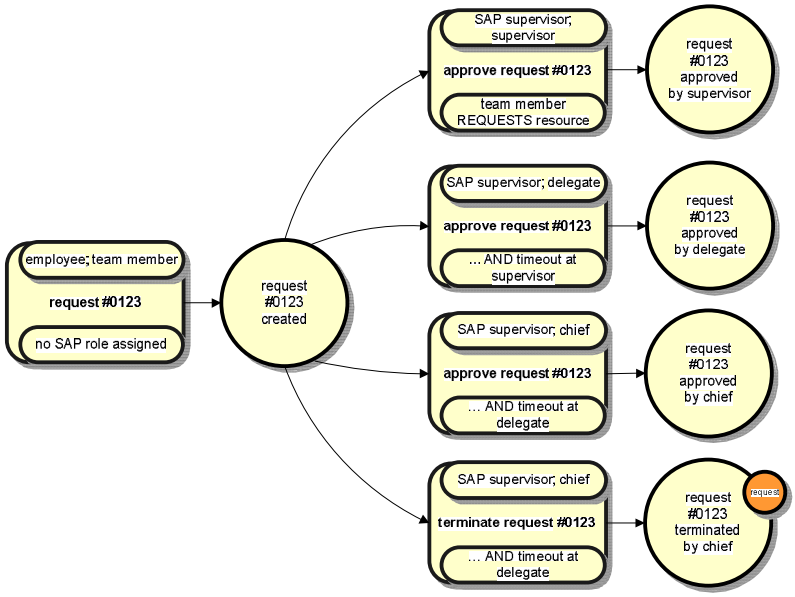


Figure 9: An instance of this example is given in section 5.

5 Example

The following example of a simple real live process shows how an implemented instance of the generic approval-request-process could look like.

The context of the following example is a classical sitation for a consultant, who has to do a specific work inside a company. For his work he needs some specific SAP-Tools, which are running under Windows, so he needs also to logon to Windows. For the internal project specific work he needs additionally access to the existing groupware-system inside the company for mailing and calendering.
The real provisioning process will only start, if the the request is completely approved. If the request is only partial approved it will finally (at the end) rejcected. After such a rejection (the request is normaly closed and the requestor is informed. This step is not shown to make the grafic more readable.

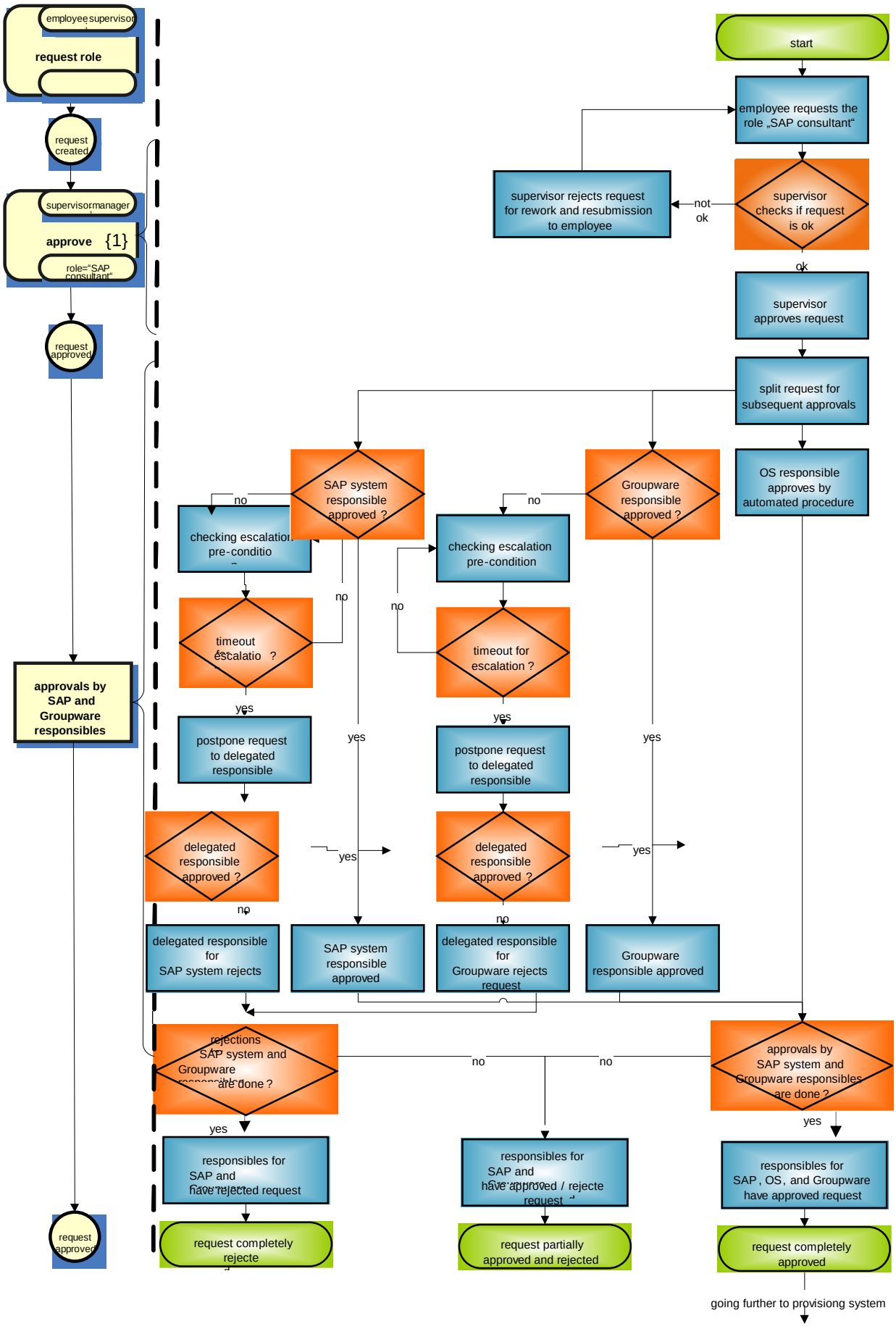
Scenario (description of the most important facts for the example scenario):

- A role “SAP Consultant” exists which comprises access rights in an SAP-System, some permissions in a specific groupware system and different access rights to access the Operating System.
- Based on the internal corporate policies the responsibilities for the SAP system and the groupware system are located in different departments.

- The request for all necessary roles must first be approved by the employee, which is responsible for the external consultant – the supervisor of the consultant. In our example the internal security policy also defines, that the responsible persons for SAP and the groupware system must additional approve such a request
- Based on the general security policy, which grants the access to the operating systee in such a case without any aproval, no manual approval is necessary and all will be done automatically.
- For SAP-system access an escalation flow is defined, so it is granted that the request will be handled in a proper time period. The situation is the same for the groupware system.
- Only if all approvals are granted, the request is forwarded to the provisioning system, to implement all the necessary access rights on the relevant IT systems.

Important comments for the following process, to explain specific parts:

- The usage of both “employee” und “superior” in the petri net is an allowed extension, which we used here.
- To simplify the picture of the classical process description (right part), we have not added the possibility for a rejection through the supervisor or the other responsible persons. This is only to make the picture more readable.
- For the escalation process the situation is the same as for the rejection. It is not containd in the right picture to be able to show the process on one page. The escalation (if nobody is reaction) would be triggerd through a timing event (which is also not put into the picture to simplify it)
- In this exampe constrains like separations of duties, etc. are not contained.
- In out example escalation is only used for handling timeouts. We have here no content specific escaltion – we have in such a case only a rejection.


employee supervisor
request role
request created
supervisor manager
approve {1}
role="SAP consultant"
request approved
approvals by SAP and Groupware responsibles
request approved
start
employee requests the role „SAP consultant“
supervisor checks if request is ok
not ok
supervisor rejects request for rework and resubmission to employee
ok
supervisor approves request
split request for subsequent approvals
OS responsible approves by automated procedure
SAP system responsible approved ?
no
checking escalation pre-conditio
timeout for escalatio ?
yes
postpone request to delegated responsible
delegated responsible approved ?
yes
no
delegated responsible for SAP system rejects
checking escalation pre-condition
timeout for escalation ?
yes
postpone request to delegated responsible
delegated responsible approved ?
yes
no
delegated responsible for Groupware rejects request
Groupware responsible approved ?
yes
Groupware responsible approved
SAP system responsible approved
rejections SAP system and Groupware are done ?
yes
responsibles for SAP and have rejected request
request completely rejecte
no
responsibles for SAP and have approved / rejecte request
request partially approved and rejected
approvals by SAP system and Groupware responsibles are done ?
yes
responsibles for SAP , OS, and Groupware have approved request
request completely approved
going further to provisiong system

1

6 Transfer the Petri net to a workflow model

It is important for the understanding of the generic model to have a clear guidance how to transform a Petri net, which is a state oriented notation, into a flow oriented model. This is necessary because the tools for the implementation of identity management solutions, which are available on the market, require flow oriented notations. Hence the models need to be created using the tools themselves.

Due to this situation mapping of Petri nets to a notation as shown in the example pertains to the overall processing of the generic IAM model, too.

7 Formats and tools

The decision to build the generic model using Petri nets allows taking advantage of the powerful tools and defined exchange formats, which were developed over the last years in the academic world as well as in commercial institutions. General it can be said that Petri-Nets can be used for technical implementations as well as for business implementations.

They allow building models of identity management processes and simulating the overall process with any kind of parameters through their built-in simulation features.

7.1 Overview on Petri net simulation tools

A wide range of such simulation tools exist. An overview of the existing tools is given by [Heitmann and Moldt, 2007] and [Mcleish, 2007]

We are currently evaluating several of them to decide which one is best suited for our purpose of modelling and simulating the defined processes. The result of the evaluation will be published shortly.

7.2 The Petri net interchange format

Caused by the huge amount of tools and the broad range of applications of Petri nets (e.g. business processing, state machines for technical flows, etc.) there is a strong move towards a generic interchange format can be observed. This discussion is mainly driven by the academic sector [Kindler, 2004].

The Petri Net Mark-up Language (PNML) is one of the proposals of an XML-based interchange format for Petri nets. A distinctive feature of PNML is its built-in flexibility: It supports the general features of all types of Petri nets and it includes the possibility to map the specific features of the several specific Petri net types. The specific features are defined in a separate Petri Net Type Definition (PNTD) for each kind Petri net [Jüngel, Ekkart and Weber, 2000].

Detailed information can be found e.g. at [Weber, 2006].

At the moment some research is done which description language (like PLNM) is the best to be used for generic iam usage.

8 Outlook

The approach presented in this paper has the potential to cover all possible processes of the essential process models. It is therefore on a sufficiently high level of abstraction.

In order to provide tangible benefits in the daily life of any person responsible for the introduction, change or management of Identity management processes it has to be …

- enriched by activities / sub-processes dealing with physical actions,
- customised in order to replace the generic names of the objects and subjects by context specific denominators,
- specified by deciding, which actions are determined implicitly, e.g. driven by policy, and which ones are triggered by requests and result in a workflow.
- transformed into a standardised and directly executable format, e.g. BPEL 2.0 or WS-BPEL Extension for People[9].

In the essential model only those processes are documented, which perform a transformation meaningful in terms of the business. They do not contain any physical sub-processes or process components (activities).

Physical activities are those performing transportation, translation or checking for exceptions (e.g. errors). An example of a typical physical activity is the provisioning of systems through specific interface connectors (adaptors).

The generic subjects, the objects they are acting on, and even the actions, they are performing are expected to be named differently in different corporations. They have to be replaced by the names commonly used in this specific context. When the generic processes are projected from the essential level to a customer specific real enterprise level, this additional information has to be introduced by the actual customer.

As it is in the corporation's very interest to automate as many process steps as possible, many necessary decisions may be determined by corporate policies either directly or via a role model. Only the remaining ad-hoc decisions need to be explicitly handled by the workflow. Hence the policy driven decisions have to be marked in the model.

The model includes the ability to generate executable code in an appropriate. The process how to derive BPEL-code or proprietary format code from the generic model and the issues of methodology attached to it will be explored in one of the next papers.

9 References

[EU DIR 2006/43/EC]
Directive 2006/43/EC of the European Parliament and of the Council of 17 May 2006 on statutory audits of annual accounts and consolidated accounts, amending Council Directives 78/660/EEC and 83/349/EEC and repealing Council Directive 84/253/EEC, THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION, 2006

[Ferraiolo and Kuhn 2007]
David F. Ferraiolo (Autor), D. Richard Kuhn, Ramaswamy Chandramouli, Role-Based Access Control, Artech House Computing Library

[Heitmann and Moldt 2007]
Frank Heitmann, Daniel Moldt (publishers), Petri Nets Tools Database Quick Overview, http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html, June, 2007

[ISO/IEC 15909-2, 2005]
Software and Systems Engineering - High-level Petri Nets. Part 2: Transfer Format. International Standard ISO/IEC 15909-2, Working Draft Version 0.9.0, June, 2005

[9] http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/

[ITGI COBIT 4.1, 2007]
Control Objectives for Information and related Technology (COBIT) 4.1, IT Governance Institute (ITGI), 2007

[Jensen, 1997]
Kurt Jensen, Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use, Volume 1, 2nd ed., Springer-Verlag Berlin, Heidelberg, New York, 1997

[Jüngel, Ekkart and Weber, 2000]
Matthias Jüngel, Ekkart Kindler and Michael Weber, Towards a Generic Interchange Format for Petri Nets, Humboldt University of Berlin, 2000

[Keller and Nüttgens, Scheer 1992]
Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). Erschienen in der Reihe: Veröffentlichungen des Instituts für Wirtschaftsinformatik. Scheer, A.-W. (Hrsg.). Heft 89, Saarbrücken 1992, http://www.iwi.uni-sb.de/Download/iwihefte/heft89.pdf, 1992

[Kindler, 2004]
Using the Petri Net Markup Language for Exchanging Business Processes, University of Paderborn, Computer Science Department, 2004

[Mcleish, 2007]
Kevin Mcleish, Petri Nets, http://www.cse.fau.edu/~maria/COURSES/CEN4010-SE/C10/10-7.html, 2007

[OGC ITIL 2, 2005]
IT Infrastructure Library (ITIL) Version 2, library of eleven books, Office of Government Commerce (OGC), United Kingdom, issue years of single books between 1999 and 2006, 2006

[OGC ITIL 3, 2007]
IT Infrastructure Library (ITIL) Version 3, library of five books, Office of Government Commerce (OGC), United Kingdom, 2007

[Reisig, 1986]
Wolfgang Reisig, Petrinetze - Eine Einführung, 2. überarbeitete Auflage, Springer-Verlag, Heidelberg, 1986

[Stuart, 1999]
Helen Stuart, Corporate Communications: An International Journal, Volume: 4 Issue: 4 Page: 200 - 207 DOI: 10.1108/13563289910299328, MCB UP Ltd., 1999

[USA SOX, 2002]
One Hundred Seventh Congress of the United States of America at the second session, Begun and held at the City of Washington on Wednesday, the twenty-third day of January, two thousand and two, An Act to protect investors by improving the accuracy and reliability of corporate disclosures made pursuant to the securities laws, and for other purposes, H. R. 3763, 2002

[Weber, 2006]
Michael Weber, 2006-01-10, http://www2.informatik.hu-berlin.de/top/pnml/about.html, 2006

[Windley, 2005]
Phillip Windley, Digital Identity, O'Reilly Media, Inc., 1st ed., 2005