

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,
Meppen, Germany

iC Compas GmbH & Co. KG,
Pfaffenhofen, Germany

NIFIS e.V.,
Frankfurt, Germany

Arslan.broemme@aviomatik.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)
et al.

Abstract:

Here we present a top-down modelling 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 subjects 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 complements the bottom-up modelling approach of factoring out generic patterns out from an empirical base of flow oriented diagrams for IAM processes.

1 Introduction

According to our experience and the reports of the main analysts the definition of processes for the Identity & Access Management (IAM)¹ requires major effort.

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 NIFIS² 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 experts abstraction and documentation of generic commonalities from enterprise specific solutions following a bottom-up approach turned out to be remarkably difficult.

Based on the assumption that the IAM processes of an enterprise could be described completely 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. The 7 steps are...

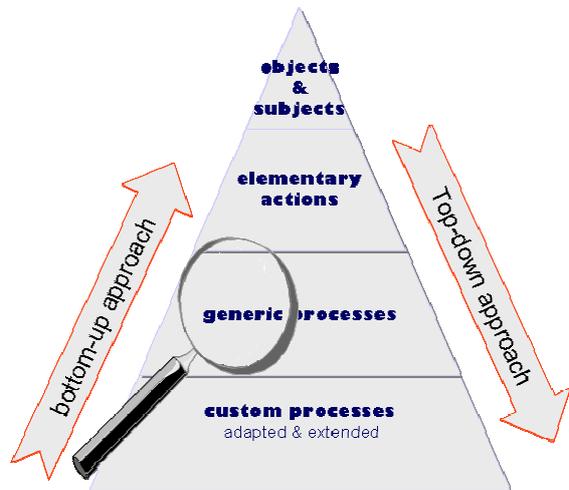


figure 1

1. Identify the fundamental objects which are involved in IAM processes.
2. Detect the derived objects which describe the relationships of the fundamental objects.
3. Identify the subjects (actors) who operate on the objects.
4. Name the elementary actions which ...
 - express the actions of the subjects on the objects,
 - express the interactions of the objects, or
 - perform object state transitions.
5. Detect business events as triggers for processes.
6. Assemble essential processes by combining the elementary actions to net of flows yielding a meaningful result in business terms.
7. Complement the essential processes by physical actions (check-, translation- and transport-steps) in order to cope with imperfections of existing implementations.

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/>)

The intention of this paper 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 of developing organisational processes fits nicely into two major trends to be observed in the general management context:

- business driven identity management
- 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, 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, enterprises need to prove compliance with some regulatory requirements (e.g. Sarbanes Oxley Act³ 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. CobiT⁵), best practice models (e.g. ITIL⁶), 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 these fundamental objects. Next we identify the subjects (actors) who operate on the objects. The intention is to detect the elementary actions by this procedure which express the actions of the subjects on the objects, the interactions of the objects, or object state transitions.

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].

3.1 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 or not. Being a user indicates a specific relationship already: the usage of resources.

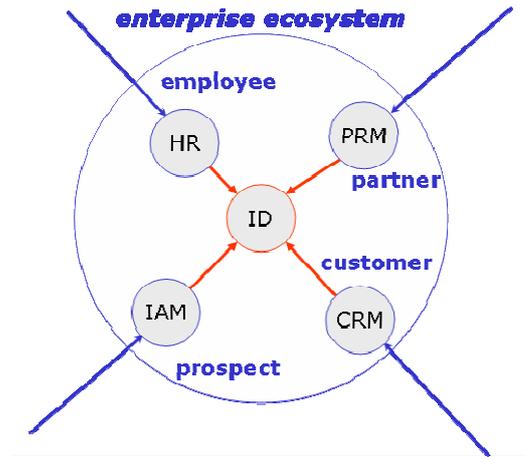


Figure 2

3.2 Objects of the corporation

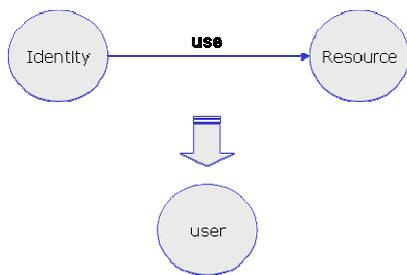


Figure 3

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 employees’ contracts, freelancers’ contracts, partner or customer contracts. 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 one (or more) business role(s). Using the generic Type-Instance-modelling pattern the instance (contract, role) defines incarnation details of the type (contract type, role type).

The operations an identity may perform on the organisation’s resources are defined in several intermediate organisational constructs (figure 4). They 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].

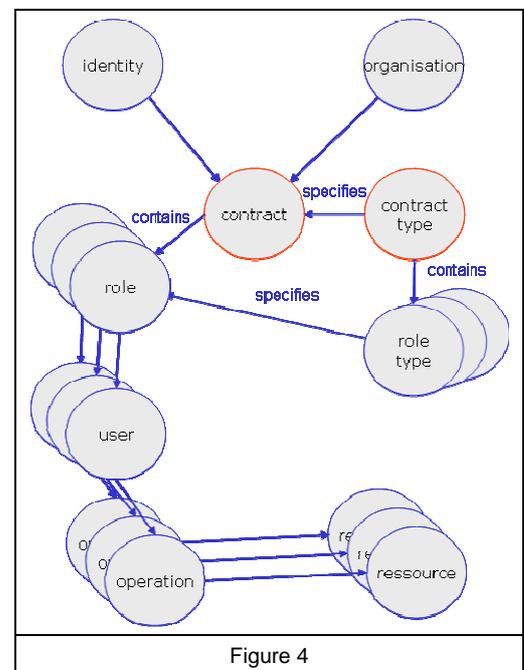


Figure 4

3.3 Subjects

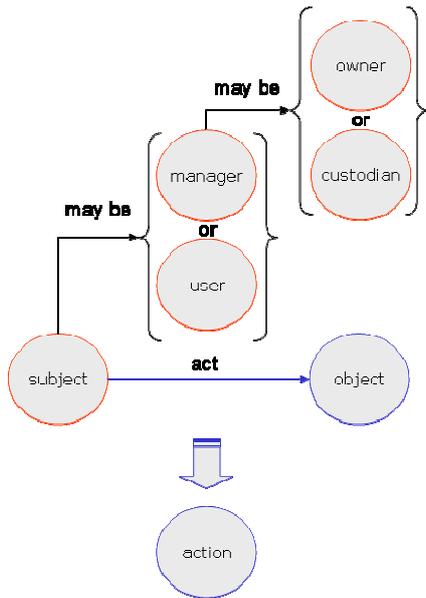


Figure 5

Next to the fundamental static objects in workflows we define subjects (actors) which act on these objects. In processes subjects (actors) act on objects. These subjects may be users or the managers of the objects. Managers are the object owners or custodians. Whereas owners are responsible for their objects, custodians act on behalf of these owners. Typical custodians are system administrators (figure 5). Hence owners delegate to the custodians.

In case of time triggered actions a policy acts on behalf of the organisation and hereby becomes the subject (driven by policy). Such Time-triggered events are very common.

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

The subject's initial action triggers an event and instantiates the process. To

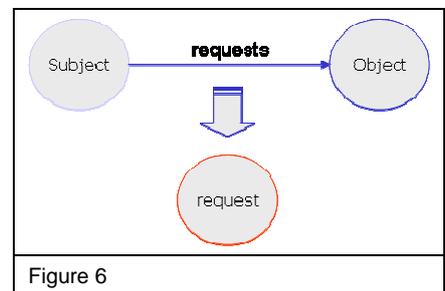


Figure 6

give 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

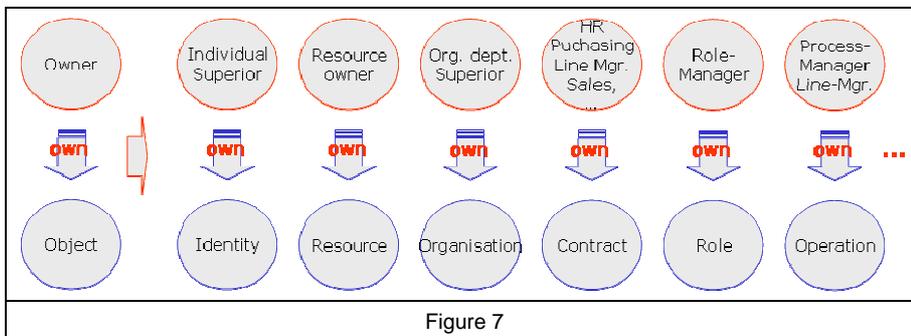


Figure 7

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.

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 Actions

Actions 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 actions are identified by state transitions.

Example:

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

3.7 Processes

Processes are made out of one or several actions [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.

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 markings⁸.

We enhanced the notation by adding general first order logic 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.

⁷ While administrative processes maintain the parameters of the operational process level, audit processes monitor 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.

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

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.

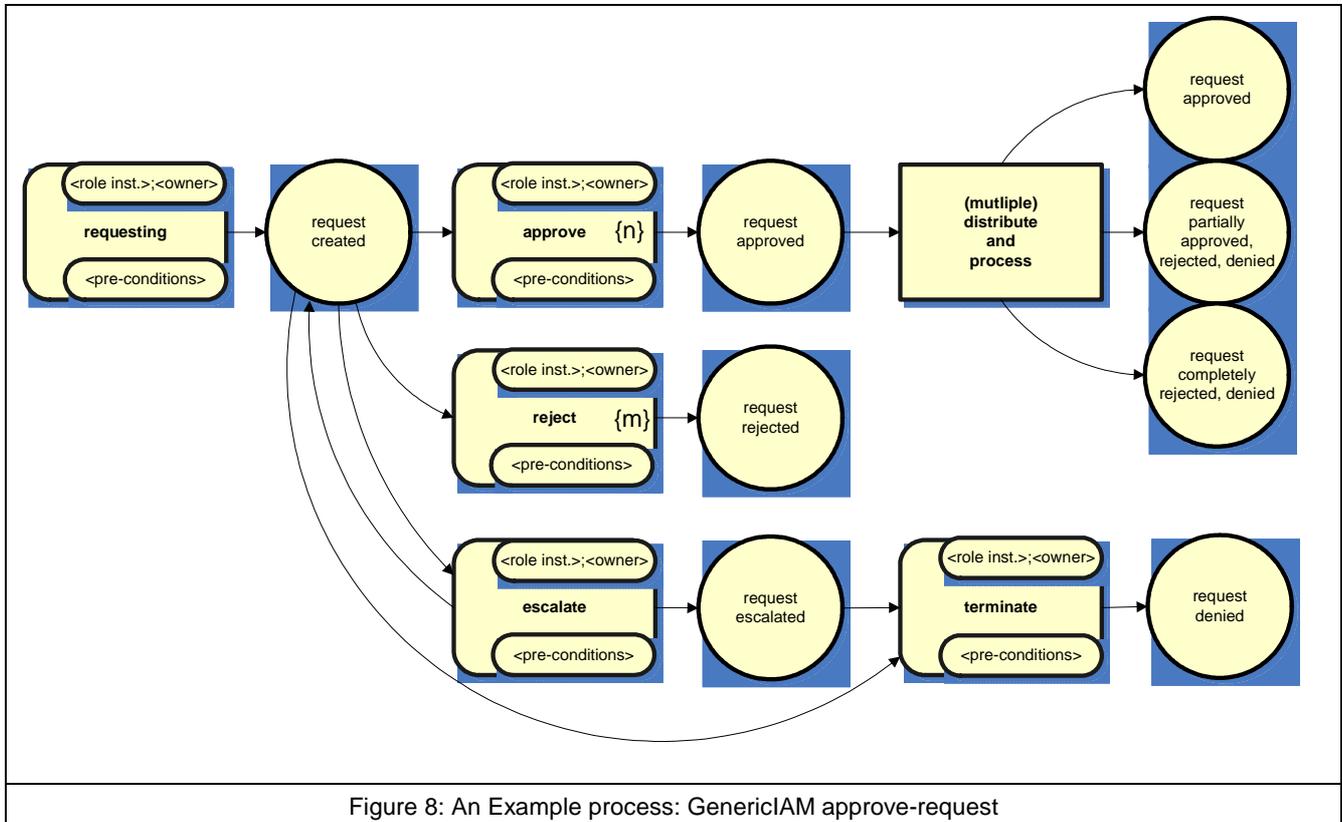


Figure 8: An Example process: GenericIAM approve-request

This process is started by generating a request within the transition “requesting”. After this initial action, the overall process is in the state “request created”. In this state several options 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”.

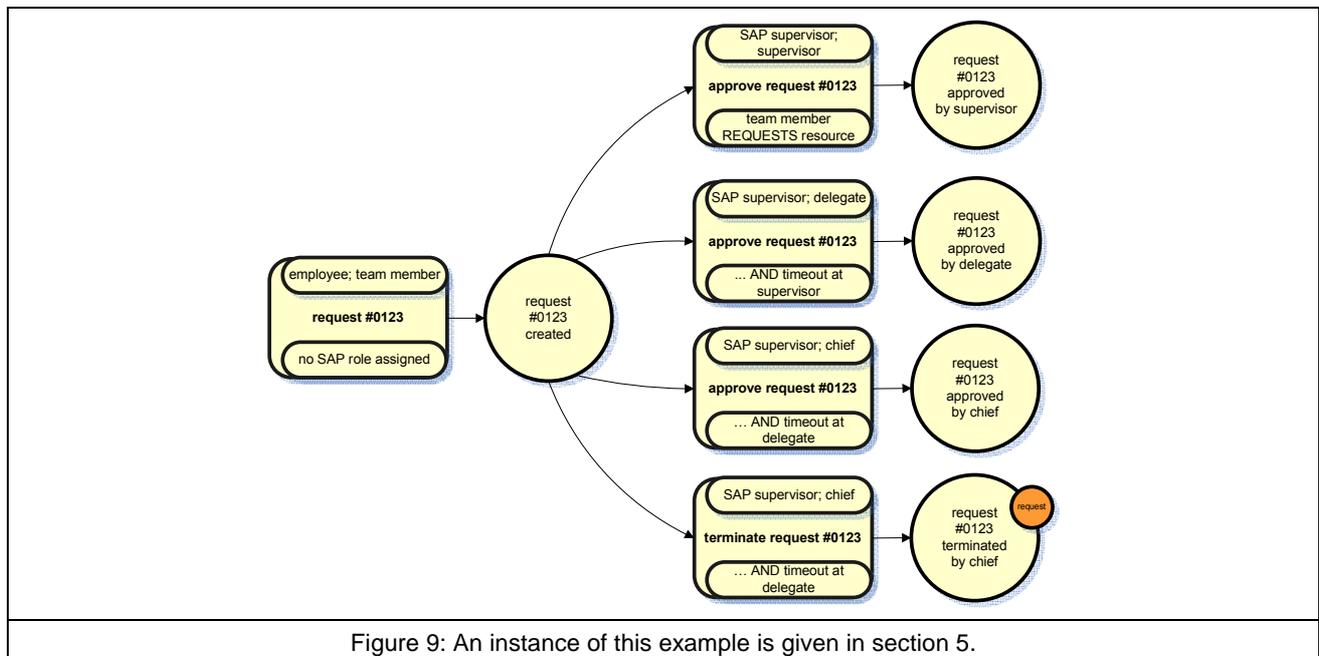
Going back again to the state “request created” the third case can be examined. If all “{n}” instances of a certain addressed role have approved a request the transition “approve” fires and the state “request approved” is reached.

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 10 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.



5 Example

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

The context of the following example is a classical situation for a consultant who has to perform a specific task within a corporation. For his work he needs access to some specific SAP-Tools which are running in a MS-Windows environment. So he needs to logon to MS-Windows too. For the internal project work like mailing and calendaring he additionally needs to access the internal groupware-system.

The provisioning process will only start when the request is completely approved. In case the request is only partially approved it will be finally rejected (at the end). After such a rejection the request is usually closed and the requestor is informed). This step is not shown to make the graphic 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 who is responsible for the external consultant – the supervisor of the consultant. In our example the internal security policy also defines that the system owners of SAP and the groupware system must additionally approve a request of this type.
- Based on the general security policy which grants the access to the operating system without any further approval, 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. This applies to the groupware system too.
- 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” and “superior” in the Petri net is an allowed extension which we used here.
- To keep the picture of the classical process description (right part) simple, we did not add the possibility for a rejection through the supervisor or the other responsible persons.
- For the escalation process the situation is the same as for the rejection. It is not contained in the right picture to be able to show the process on one page. The escalation (if nobody is reacting) would be triggered through a time event (which is also not put into the picture.)
- In this example constraints like separations of duties, etc. are not contained.
- In our example escalation is only used for handling timeouts. We do not have any content specific escalation – we have only a rejection in such a case.

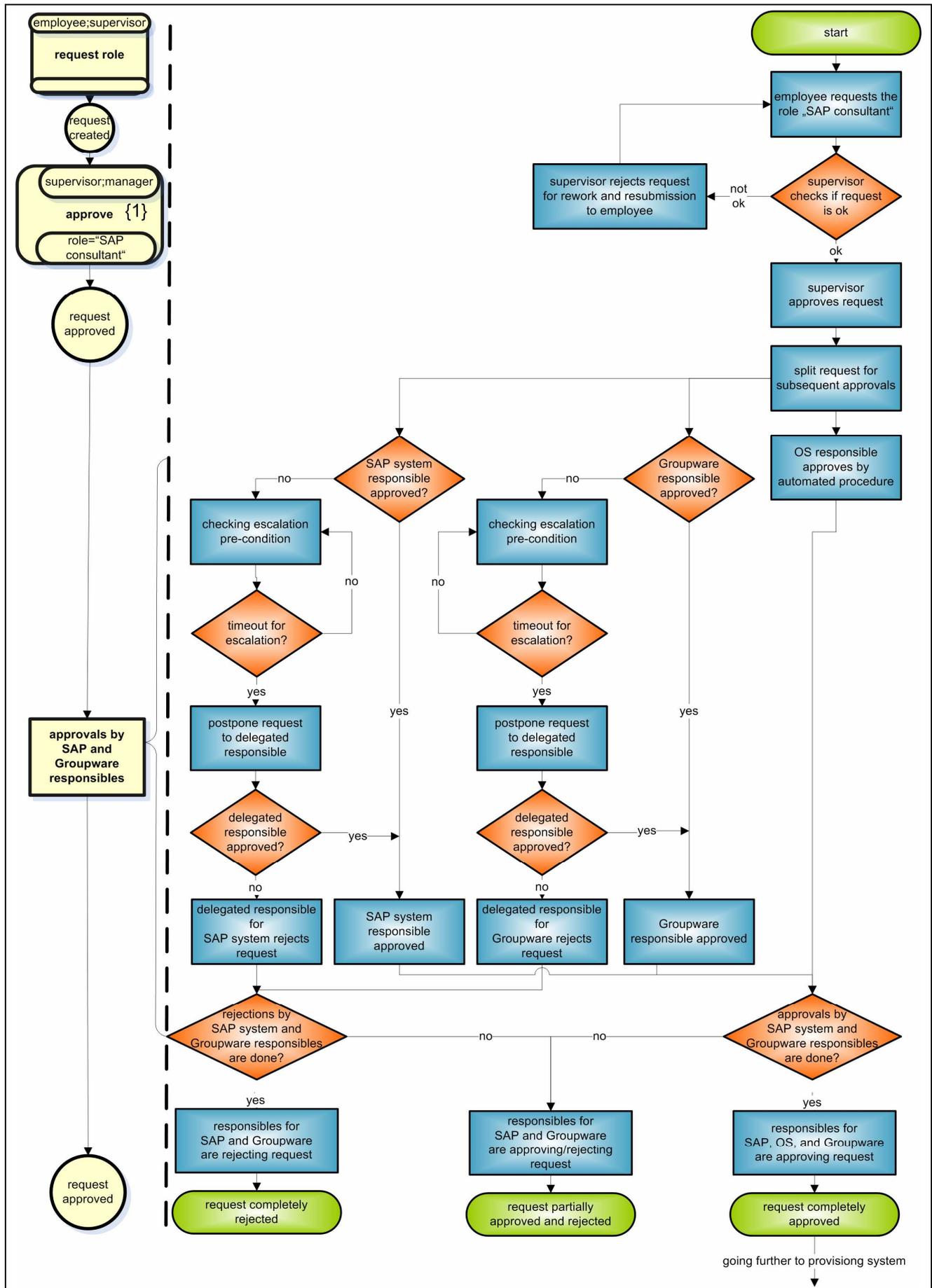


Figure 10:

6 Transfer the Petri net to a workflow model

It is essential 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 have to be transferred to these tools manually.

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. Generally it can be stated that Petri-Nets may 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.) 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 [Jünger, Ekkart and Weber, 2000].

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

Currently 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 therefore operates 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 actions / 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 People⁹.

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 (actions).

Physical actions are those performing transportation, translation, or checking for exceptions (e.g. errors). An example of a typical physical action 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. 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 format to be processed by workflow engines. 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

⁹ <http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/>

[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

[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

[Jünger, Ekkart and Weber, 2000]

Matthias Jünger, 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