

enaio[®]

Softwaredokumentation
enaio[®] Systemhandbuch WMS

Version 8.50

Sämtliche Softwareprodukte sowie alle Zusatzprogramme und Funktionen sind eingetragene und/oder in Gebrauch befindliche Marken der OPTIMAL SYSTEMS GmbH, Berlin oder einer ihrer Gesellschaften. Sie dürfen nur mit gültigem Lizenzvertrag benutzt werden. Die Software sowie die jeweils zugehörige Dokumentation sind nach deutschem und internationalem Recht urheberrechtlich geschützt. Das illegale Kopieren und Vertreiben der Software stellt Diebstahl geistigen Eigentums dar und wird strafrechtlich verfolgt. Alle Rechte vorbehalten, einschließlich der Wiedergabe, Übermittlung, Übersetzung sowie Speicherung mit/auf Medien aller Art. Für vorkonfigurierte Testszenarien oder Demo-Präsentationen gilt: Alle Firmennamen und Personen, die in Beispielen (Screenshots) erscheinen, sind frei erfunden. Eventuelle Ähnlichkeiten mit tatsächlich existierenden Firmen und Personen sind zufällig und unbeabsichtigt.

Copyright 1992 – 2014 by

OPTIMAL SYSTEMS GmbH
Cicerostraße 26
D-10709 Berlin

02.02.2017
Version 8.50

Inhalt

Vorwort	5
Einleitung	6
Grundlagen	7
Allgemein	7
Prozessmodelle	7
Aktivitäten	7
Transitionen	8
Rollen	8
Splits und Joins	9
Schleifen	10
Variablen, Anwendungen und Parameter	10
Speicherung von Prozessmodellen	13
Modellverwaltung	13
Organisationsstruktur	14
Architektur	15
Prozessdesign	17
Allgemein	17
Modellierung des Geschäftsprozesses	18
Abbildung der Organisationsstruktur	18
Organisationsexplorer	20
Design der Workflowmasken	22
Abbildung des Modells	23
Modellexplorer	23
Modelleditor	24
Erweiterte Prozessmodellierung	38
Schleifen	38
Komplexere Schleifen	43
Multi-Instanz-Aktivitäten	49
Adhoc-Aktivitäten	50
Events	51
Beispiele für Eventcodes	55
Einrichtung des Protokolls	60
Definition des Protokollvariablentyps	61
Anlegen der Protokollevnts	61
Integration der Digitalen Signatur und Schrittautorisierung	62
Aktivierung des Modells	62
Import und Export von Workflowprojekten	63
Exportieren	64
Importieren	64
Der Standard Ad-hoc-Workflow und der Taskflow	65
Standard Ad-hoc-Workflow	66
Taskflow	73
Betrieb und Administration	76
Lizenzierung und Berechtigungen	76
Workflowkomponenten im enaio® client	76
Workflowbereich	77

Einstellungen	80
Stellvertretung	80
Workflow-Administrator	81
Prozess- und Aktivitätenstatus	82
Prozess-Eigenschaften	83
Organisation	85
Server- und Datenbank-Eigenschaften	85
Statistik-Reports	85
Workflow Simulation	88
Automatisiertes Starten von Workflowprozessen	91
Integration über den Importassistenten	91
Workflow-Skript-Komponente	91
enaio® document-storage	93
Beispiele für Workflowszenarien	94
Urlaubsworkflow	94
Bestellworkflow	94
Anhang	98
Analysebogen für Geschäftsprozesse	98
Programmierreferenz	100
Umgebung	100
Standardfunktionen von VBScript	100
Active Data Object	100
File System Object	101
Schnittstellenobjekte	101
Reportkonfiguration XML	102
Allgemein	102
Prozessübersichtreportkonfiguration	102
Prozessdetailreportkonfiguration	103
Report XML	105
Prozessübersichtreport	105
Prozessdetailreport	108

Vorwort

Dieses Systemhandbuch wendet sich an alle, die über die reine Bedienung hinaus Informationen über Abläufe, Datenflüsse, Wechselwirkungen, Strukturen und Technologien des Dokumentenmanagement- und Workflowsystems enaio® benötigen. Es enthält detaillierte Informationen, die dem Einrichten, der Konfiguration und Betreuung des Systems dienen.

Es wurde versucht, alle Informationen zusammenzutragen und zu systematisieren, die benötigt werden, um eine Vorstellung über den Aufbau, die Systemarchitektur sowie die Wechselwirkungen zwischen den Komponenten des Systems zu erhalten. Damit sollte es möglich sein, Abläufe bei bestimmten Vorgängen besser verstehen zu können, den Sinn, Vor- und Nachteile bestimmter Einstellungen, Konfigurationen und Installationen zu erkennen.

Es wurde, soweit es für die Verständlichkeit möglich war, vermieden, Informationen der Benutzer-Handbücher zu wiederholen. An einigen Stellen wird auf weitergehende Beschreibungen verwiesen, die konkrete Informationen über Teilbereiche des DMS enthalten, wie z.B. die Datenbankbeschreibung, die Beschreibung des Applikationsservers, die Installationsbeschreibung etc.

Ziel dieses Handbuches ist, dem Leser die Systemzusammenhänge bei bestimmten Vorgängen zu verdeutlichen, den Sollzustand vor, während und bei Abschluss bestimmter Aktionen darzulegen. Somit wird er in die Lage versetzt, nachzuvollziehen, ob die konkrete Aktion erfolgreich durchgeführt wurde, wo möglicherweise Fehler aufgetreten sind. In diesem Sinne soll das Systemhandbuch auch Hilfe beim Beheben von Problemen und der Erkennung von Fehlern sein.

Verzichtet wurde auf die Darlegung der Algorithmen, die bei den jeweiligen Vorgängen programmtechnisch realisiert wurden.

Das Handbuch wurde in zwei Teile untergliedert. Im ersten Teil wird das Dokumenten Management System und in diesem zweiten Teil das Workflow Management System behandelt.

Einleitung

Mit der Workflowkomponente wird das Produktspektrum von enaio® um eine prozessorientierte Komponente ergänzt. Mit ihr wird es dem Betreiber ermöglicht, seine Geschäftsprozesse elektronisch abzubilden und zu steuern. Insbesondere die enge Kopplung an das Dokumentenmanagementsystem bietet erhebliche Optimierungspotentiale in der Bearbeitung von Geschäftsprozessen. Dokumente stehen an allen Bearbeitungsschritten für die berechtigten Anwender zur Verfügung und werden entsprechend dem jeweils festgelegten Prozessmodell automatisch nach der Bearbeitung an den nächsten Bearbeiter weitergeleitet.

Der enaio® Workflow ist eine im System integrierte Komponente und vereint in sich die bewährten Vorzüge von enaio®:

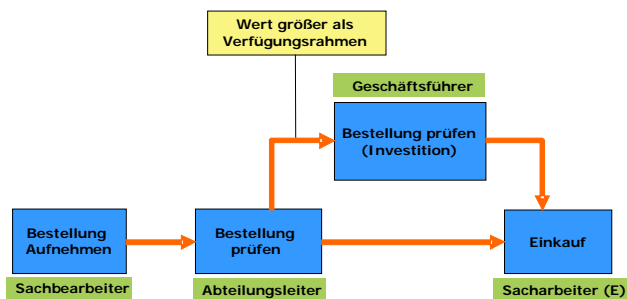
- § Flexibilität
- § Konfigurierbarkeit
- § Stabilität

Durch die im System enthaltenen Administrationstools ist es möglich, Geschäftsprozesse schnell und sicher zu implementieren, zu verwalten und im laufenden Betrieb zu überwachen.

Vorliegendes Dokument beschreibt die Funktionalität der Workflowkomponente und deren Integration in das Gesamtsystem. Im Folgenden werden zunächst Grundlagen für das Verständnis des Workflowmanagementsystems vermittelt. Anschließend wird das Design von Prozessen und Organisationsstrukturen betrachtet, die entsprechenden Tools vorgestellt und sodann der Betrieb des Workflowmanagementsystems beschrieben.

Beispiel

In einem Unternehmen soll der Bestellvorgang durch ein Workflowsystem elektronisch unterstützt werden. Dabei soll nach der Bedarfsmeldung ein Abteilungsleiter die Bestellung genehmigen und wenn es sich um ein Investitionsgut mit einem Wert über 400 € handelt, soll der Geschäftsführer die Bestellung freigeben, bevor diese durch den Einkauf realisiert wird.



Durch den enaio® Workflow können natürlich erheblich komplexere Vorgänge abgebildet werden, eine Auswahl zur Verfügung stehender Szenarien ist im Anhang aufgelistet.

Grundlagen

Allgemein

Ein Workflowmanagementsystem (WFMS) dient der elektronischen Abbildung eines Geschäftsprozesses innerhalb einer realen Unternehmensstruktur. Durch das System werden Arbeitsschritte automatisch an die jeweils zugeordneten Mitarbeiter weitergeleitet und die Ausführung der Arbeitsschritte unterstützt.

Um ein Workflowmanagementsystem zu betreiben, ist es notwendig, einerseits die an den Geschäftsprozessen beteiligten Anwender (z.B. Sachbearbeiter) abzubilden und andererseits die gewünschten Prozesse (z.B. Urlaubsantrag, oder Rechnungseingang) im System zu modellieren.

Die Teilnehmer an einem Workflowmanagementsystem werden in einer **Organisationsstruktur** zusammengefasst. Geschäftsprozesse werden im WFMS als Prozessmodelle abgebildet, dies nennt man auch **Prozessdesign**.

Um die Modellierung der Geschäftsprozesse zu verstehen, sollen hier zunächst die einzelnen Komponenten eines elektronisch abgebildeten Prozesses beleuchtet werden. Die genannten Begrifflichkeiten sind allgemeingültig für alle Workflowmanagementsysteme, die sich dann in der konkreten Ausprägung unterscheiden.

Prozessmodelle

Ein Prozessmodell ist die abstrakte Darstellung eines Geschäftsprozesses, um diesen für das Workflowmanagementsystem in einer interpretierbaren Form bereitzustellen. Modelle beschreiben also den konkreten Geschäftsfall, während Prozesse den konkreten Vorgang darstellen. Prozesse sind demnach Instanzen des Prozessmodells.

Der enaio® Workflow ist in seinem Modelldesign angelehnt an die Spezifikationen der Workflow Management Coalition (WfMC), insbesondere an die XML-Process Definition Language. Über die Spezifikationen hinaus gibt es Erweiterungen um einige optionale Attribute im Zusammenhang mit der Kopplung an Dokumentenmanagement- / Contentmanagement-Systeme. Des Weiteren sind nicht alle in der Spezifikation geforderten Interfaces realisiert, da sie für den Normalbetrieb des Systems nicht notwendig sind.

Aktivitäten

Ein Prozessmodell, wie etwa das Modell eines Bestellvorgangs besteht aus mehreren einzelnen Arbeitsschritten, die durchlaufen werden müssen, um den Vorgang vollständig durchzuführen. Diese Arbeitsschritte werden Aktivitäten genannt.

Eine Aktivität ist eine abgegrenzte und zusammenhängende Menge von Tätigkeiten, die ein Anwender an einem Schritt (z.B. Kenntnisnahme einer Bestellung) innerhalb eines Prozesses mit Unterstützung einer Anwendung (z.B. enaio® client) durchführt.

Jedes Prozessmodell besitzt eine Start- und eine Endaktivität. Diese beinhalten keine realen Arbeitsschritte, dienen aber der Steuerung des Workflowprozesses. Es kann darüber hinaus noch weitere Steueraktivitäten innerhalb eines Prozessmodells geben, was im Abschnitt über das Prozessdesign ausführlicher erörtert wird.

Im Beispiel des Bestellworkflows besteht das Prozessmodell aus folgenden Aktivitäten:

- § Startaktivität
- § Bestellung aufnehmen
- § Bestellung prüfen
- § Bestellung prüfen (Investitionsgut)
- § Einkauf
- § Endaktivität

Multi-Instanz-Aktivitäten

Eine Aktivität wird von genau einem Teilnehmer ausgeführt. Multi-Instanz-Aktivitäten können von mehreren Teilnehmern zugleich ausgeführt werden. Diese Aktivitäten eignen sich dadurch besonders für Abstimmungen und Kenntnisnahmen.

Von einer Multi-Instanz-Aktivität wird die Transition zur nächsten Aktivität beschriftet, sobald alle einzelnen Teilnehmer die Aktivität durch das Weiterleiten beendet haben. Alternativ kann die Transition durch ein Event ausgelöst werden.

Innerhalb der einzelnen Instanzen einer Multi-Instanz-Aktivität müssen Daten über lokale Variablen verwaltet werden. Diese Daten werden über ein Event zusammengeführt werden.

Adhoc Aktivitäten

Adhoc Aktivitäten sind Bereiche in denen Aktivitäten ohne vorgegebene Transitionen eingebunden werden. Bei diesen Aktivitäten legt der Benutzer über Lauflisten im enaio® client fest, welche Aktivitäten aufeinander folgen und welche Benutzer diese Aktivitäten ausführen sollen. Lauflisten können den weiteren Ablauf fest vorgeben oder so gestaltet werden, dass die jeweiligen Benutzer die Laufliste bearbeiten und erweitern können.

Transitionen

Nachdem herausgestellt wurde, dass sich die Tätigkeiten in einem Prozess in mehrere Aktivitäten unterteilen lassen, ist es nun wichtig festzuhalten, in welcher Reihenfolge und unter welchen Bedingungen die Aktivitäten ausgeführt werden sollen. Das ist wichtig, denn das Workflowsystem soll genau dazu eingesetzt werden, um die richtigen Schritte für die einzelnen Bearbeiter festzulegen und ihnen die Aktivitäten zuzuteilen. Also werden die Übergänge innerhalb des Workflowprozesses gesondert gekennzeichnet und als Transitionen bezeichnet.

Eine Transition ist der Übergang zwischen zwei Aktivitäten. Eine Transition hat immer genau eine Startaktivität und genau eine Endaktivität. Zusätzlich kann eine Bedingung angegeben sein, unter der die Transition zustande kommt.

An oben angeführtem Beispiel des Bestellworkflows lässt sich eine bedingte Transition gut demonstrieren. Die Bestellung soll vom Abteilungsleiter an den Geschäftsführer in dem Falle gehen, wenn der Bestellwert größer als 400 € ist. Dabei ist der Übergang vom Abteilungsleiter zum Geschäftsführer eine Transition, die Klausel (>400 €) ist die Bedingung. Die Transitionen im Prozessmodell sind die Übergänge zwischen den Aktivitäten:

Startaktivität	Endaktivität	Bedingung
Startaktivität	Bestellung aufnehmen	Keine
Bestellung aufnehmen	Bestellung prüfen	Keine
Bestellung prüfen	Bestellung prüfen (Investitionsgut)	Wert >= 400 €
Bestellung prüfen	Einkauf	Wert < 400 €
Bestellung prüfen (Investitionsgut)	Einkauf	Keine
Einkauf	Endaktivität	Keine

Tatsächlich ist es jedoch so, dass in einem realen Prozess noch weitere Transitionen notwendig sind, etwa, wenn eine Genehmigung nicht erteilt wurde und der Vorgang an einen Sachbearbeiter zurückgeleitet wird. Der Einfachheit halber sollen diese Transitionen hier jedoch nicht weiter betrachtet werden.

Rollen

Bisher wurde dargestellt, wie ein Prozessmodell prinzipiell aufgebaut ist. Um den Prozess im WFMS korrekt laufen zu lassen, muss noch festgelegt werden, wer die einzelnen Aktivitäten bearbeiten soll.

Der Anwender ist gewöhnlich eine natürliche Person, kann aber auch eine Gruppe von Anwendern oder eine Komponente im System sein, die automatisch arbeitet. Um hier eine Verallgemeinerung zu treffen, werden alle Anwender, ob nun natürliche Personen oder automatische Komponenten, als **Teilnehmer** bezeichnet. In unserem Beispiel sind die Sachbearbeiter, Abteilungsleiter, Geschäftsführer und Sachbearbeiter (Einkauf) die Teilnehmer des Prozessmodells. Im Kontext einer Aktivität werden Teilnehmer als **Bearbeiter** bezeichnet.

Da nun mehrere reale Teilnehmer für die Bearbeitung eines bestimmten Arbeitsschrittes zuständig sein können, ist es sinnvoll, diese in so genannten **Rollen** zusammenzufassen.

Eine Rolle ist eine logische Zusammenfassung von Teilnehmern, die gleichberechtigt, aber nicht gleichzeitig eine bestimmte Menge von Arbeitsschritten ausführen können. Die Zuordnung der Rollen zu den Aktivitäten ist Bestandteil des Prozessmodells.

So kann es in unserem Beispiel mehrere Sachbearbeiter im Einkauf geben, die gleichberechtigt die genehmigten Bestellungen bearbeiten können. Die Aktivität „Einkauf“ ist aber so gestaltet, dass sie gleichzeitig nur von einem Sachbearbeiter bearbeitet wird. Der Sachbearbeiter nimmt sich also aus dem Vorrat der zugewiesenen Bestellungen eine heraus und bearbeitet diese vollständig. Die Aktivität kann gleichberechtigt (da jeder Sachbearbeiter die gleiche Tätigkeit verrichten kann) aber nicht gleichzeitig ausgeführt werden, da nach der Zuordnung einer konkreten Bestellung zu einem Sachbearbeiter dieser die Bestellung auch bis zu Ende bearbeiten muss.

Einer Rolle können mehrere reale Anwender zugeordnet sein (Herr Maier und Frau Lehmann sind Sachbearbeiter), andererseits kann ein Anwender mehrere Rollen innehaben (Herr Maier ist Sachbearbeiter und Arbeitsschutzbeauftragter).

In unserem Beispiel gibt es folgende Zuordnungen der Aktivitäten zu Rollen.

Aktivität	Rolle
Bestellung aufnehmen	Sachbearbeiter (im Falle, dass Abteilungsleiter und Geschäftsführer auch Bestellungen auslösen können, würden diese hier auch aufgeführt sein)
Bestellung prüfen	Abteilungsleiter
Bestellung prüfen (Investitionsgut)	Geschäftsführer
Einkauf	Sachbearbeiter (Einkauf)

Wie oben schon gesagt, sind diese Zuordnungen Bestandteile des Prozessmodells.

Die Übernahme einer Aktivität während der Ausführung des Prozesses durch einen konkreten Teilnehmer innerhalb einer Rolle nennt man **Personalisierung**. Nach der Personalisierung muss die Aktivität aus dem sichtbaren Arbeitsvorrat für die anderen Teilnehmer der Rolle verschwinden.

Neben der Zuordnung als Teilnehmer von einzelnen Aktivitäten können den einzelnen Rollen weitere Verantwortungen zugeordnet werden. So wird festgelegt, welche Rollen Prozesse eines Modells starten dürfen und welche Rollen den laufenden Prozess beobachten können, ohne jeweils Bearbeiter des aktuellen Arbeitsschrittes zu sein.

Rollen können durch Serverskripte Personen und Arbeitsschritte zugeordnet werden. Dabei können die im enaio® editor-for-workflow angelegten Rollen verwendet werden oder durch Skripte komplett neue Prozessrollen erstellt werden. Prozessrollen erhalten eindeutige IDs. Mit der Anlage eines Arbeitsschrittes können Objekte einer Prozessrolle dem Arbeitsschritt als Teilnehmer zugeordnet werden. Spätere Änderungen an der Prozessrolle wirken sich nicht auf die Sichtbarkeit laufender Arbeitsschritte in den Eingangskörben aus. Prozessrollen werden in den Prozessen im Workflow-Empfänger-AddOn angezeigt und können so zugeordnet werden.

Splits und Joins

Sind im Workflowprozess Verzweigungen vorgesehen, so kommen oft spezielle Eigenschaften von Aktivitäten zum Einsatz, durch die Verzweigung und Zusammenführung von einzelnen Strängen innerhalb des Prozessmodells realisiert werden.

Splits

Splits treten an allen Stellen auf, wo sich nach der Bearbeitung einer Aktivität ein Prozess in unterschiedliche Bearbeitungsstränge verzweigt, also mindestens zwei ausgehende Transitionen für eine Aktivität definiert sind. Dies kann dann auftreten, wenn für die ausgehenden Transitionen Bedingungen gesetzt oder eine parallele Ausführung der nachfolgenden Aktivitäten vorgesehen ist.

Für jeden Split wird festgelegt, ob es sich um einen XOR oder AND Split handelt. Bei XOR Splits wird genau eine der ausgehenden Transitionen, deren Bedingung wahr ist, ausgeführt. Bei AND Splits werden alle ausgehenden Transitionen, deren Bedingung wahr ist, ausgeführt.

Joins

Joins führen mehrere Bearbeitungswege von Aktivitäten wieder zusammen. Je nach Typ kann der Join ausgeführt werden, wenn alle eingehenden Transitionen oder wenn mindestens eine der Transitionen beschriftet wurde. Diese Typen nennt man AND-Join (wenn für alle Transitionen ermittelt werden konnte, ob sie beschriftet wurden oder nicht) oder XOR-Join (wenn genau eine von mehreren Transitionen beschriftet wurde).

Als Beispiel für AND-Transitionen lässt sich die parallele Bearbeitung von unabhängigen Strängen eines Workflowprozesses definieren. Dies kann etwa bei Abstimmungsprozessen vorkommen, wo mehrere Beteiligte unabhängig voneinander einem Vorgang zustimmen oder ihn ablehnen sollen. Beim Zusammenführen der Bearbeitungsstränge soll mittels des AND-Joins angegeben werden, dass der Prozess erst dann wieder fortgesetzt werden kann, wenn alle Zweige bis zu Ende bearbeitet wurden.

XOR-Transitionen treten dann auf, wenn je nach dem Wert einer Variablen dieser oder jener Bearbeitungsweig beschriftet werden soll. Nach deren Bearbeitung werden diese Zweige aber mittels einer XOR-Transition wieder zusammengeführt, wobei im Prozessmodell hinterlegt ist, dass beim Eintreffen einer der Transitionen der Prozess fortgeführt werden soll.

Per Voreinstellung werden Splits und Joins als AND definiert, sodass hier kein besonderer Aufwand während des Designs betrieben werden muss.

Schleifen

In realen Prozessen besteht häufig die Forderung, Teile des Vorganges so lange zu wiederholen, bis ein bestimmter Zustand erreicht wurde. Beispielsweise soll für die Freigabe der Bestellung der Vorgang so lange zwischen dem anfordernden Mitarbeiter und dem zuständigen Abteilungsleiter weitergeleitet werden, bis die Freigabe explizit gesetzt wird. Dies kann dann notwendig sein, wenn zu einer Bestellung Rückfragen existieren oder nicht alle Informationen zur Entscheidung vorliegen.

Zu diesem Zwecke sind im Prozessmodell Schleifen vorgesehen. Schleifen sind besondere Arten von Aktivitäten, die einen bestimmten Typ (while, repeat until) und eine Schleifenbedingung besitzen. Innerhalb einer Schleife können dann eine oder mehrere Aktivitäten vorgesehen werden, die jedes Mal, solange die Schleifenbedingung gültig ist, durchlaufen werden.

Variablen, Anwendungen und Parameter

Oben wurde bereits der Begriff der Bedingung (Condition) eingeführt. Nun soll betrachtet werden, wie diese Bedingungen ausgewertet werden. Zu diesem Zwecke dienen **Variablen**, die im Modell festgelegt werden. Variablen stehen während der gesamten Ausführung eines Prozesses zur Verfügung und können durch das WFMS gelesen und manipuliert werden. Im Besonderen dienen sie dazu, um Bedingungen zu definieren. Variablen haben einen festgelegten Datentyp und können beim Start des Prozesses mit Werten vorbelegt werden. Als Datentypen stehen alle bereits aus dem DMS-Bereich bekannten Variablentypen (Integer, Real, String, Date etc.) zur Verfügung. Bei Bedarf können auch noch eigene Datentypen für die Verwaltung strukturierter Daten durch den Designer festgelegt werden.

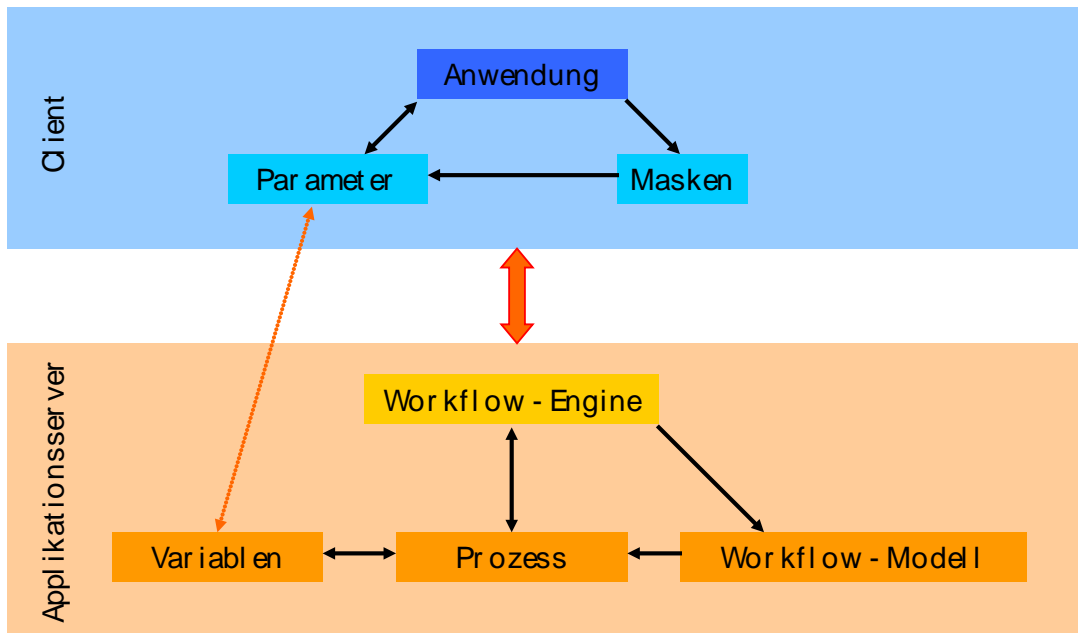
Im konkreten Beispiel wurde eine Bedingung für eine Transition definiert, wonach diese realisiert werden soll, wenn der Wert der Bestellung größer als 400 € ist. Im Prozessmodell wird also eine Variable „Wert“ eingeführt. Der Sachbearbeiter, der die Bestellung aufnimmt, trägt den Wert der Bestellung ein, belegt also die Variable. Nun kann die Variable in der Transitionsbedingung ausgewertet werden.

Bisher wurde allgemein von einem Workflowmanagementsystem gesprochen. Faktisch besteht ein solches System aber immer aus mehreren, zumindest logisch getrennten Teilen. In den meisten Systemen wird eine Workflow-Engine betrieben, d.h. eine Software, die den Start und die Bearbeitung der Prozesse überwacht. Meist ist die Workflow-Engine Bestandteil eines Applikationsservers, in kleineren Abteilungslösungen gibt es Systeme, bei denen die Workflow-Engine direkt in der Anwendersoftware integriert ist. Serverlösungen sind stabiler, leistungsfähiger und skalierbarer. Der enaio® Workflow basiert auf einer Server gestützten Workflow-Engine.

Die Workflow-Engine steuert also die Bearbeitung der Prozesse, von denen natürlich mehrere gleichzeitig aktiv sein können. Dabei überwacht sie unter anderem, wann eine Aktivität gestartet werden muss, wann sie beendet ist und für wen die nächsten Aktivitäten innerhalb eines Prozesses bestimmt sind. In diesem Zusammenhang ist es wichtig zu verstehen, dass die Engine die Prozesse nur steuert, die Ausführung der Aktivitäten wird aber von **Anwendungen** (applications) übernommen, die mit der Engine kommunizieren. Anwendungen dienen der Bearbeitung von Aktivitäten, wobei es für die Workflow-Engine unerheblich ist, ob es sich hier um eine interaktive Anwendung handelt, bei der der Anwender Daten in einer Maske bearbeitet oder ob es sich um eine andere Softwareinstanz handelt, die Operationen auf dem Datenbestand automatisch ausführt. Wichtig ist nur, dass die Anwendung mit der Workflow-Engine Daten austauschen kann, um an einem laufenden Prozess teilzunehmen. Die wichtigste Art von Daten, die von Anwendungen manipuliert werden, sind Prozessvariablen. Diese müssen der Anwendung übergeben und nach der Bearbeitung der Workflow-Engine wieder zugeführt werden. Um das zu ermöglichen, werden für Anwendungen so genannte **Parameter** definiert. Parameter werden im Modell Variablen zugeordnet, wodurch die Verknüpfung zwischen der Workflow-Engine und der Anwendung hergestellt wird.

Beim enaio® Workflow handelt es sich wie bei allen Anwendungen der enaio®-Produktfamilie um eine Client-Server-Architektur. Die Bearbeitung von Aktivitäten durch den Anwender erfolgt gewöhnlich im enaio® client. Also spielt hier der enaio® client die Rolle einer Anwendung, die mit der Workflow-Engine Daten zur Bearbeitung austauscht.

Anhand folgender Abbildung soll dies nochmals verdeutlicht werden.



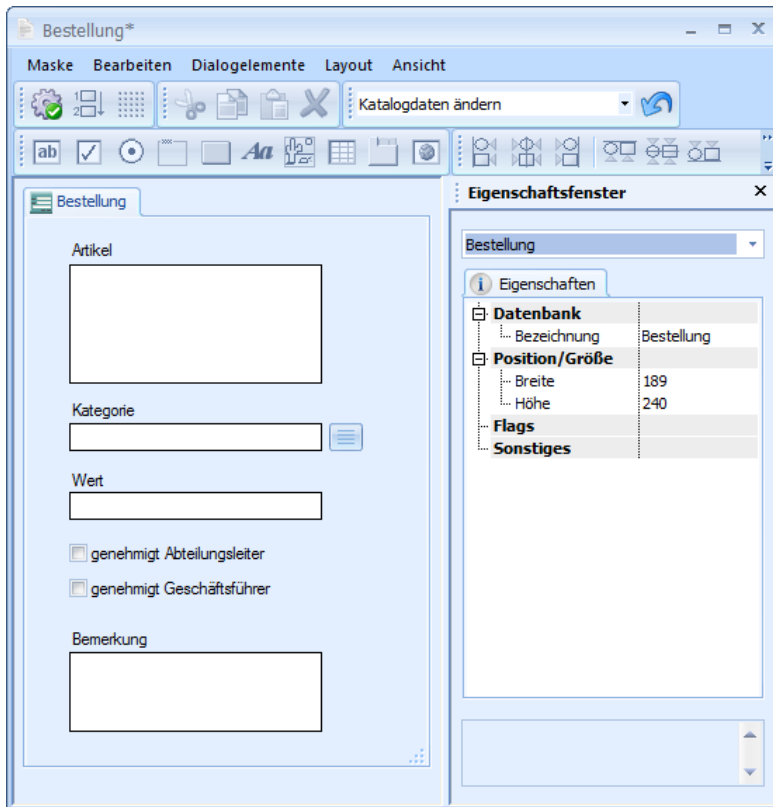
Wie oben bereits vermerkt, ist es nicht zwingend notwendig, dass Aktivitäten in einer interaktiven Clientanwendung bearbeitet werden. Ebenso kann eine Anwendung, die automatisch Daten im Zusammenhang mit anderen Subsystemen bearbeitet, Aktivitäten ausführen. Das Prinzip der Abbildung von Variablen auf Parameter der Anwendung bleibt aber auch hier bestehen.

Anwendungsparameter können in Abhängigkeit davon, in welcher Richtung die Interaktion mit Variablen erfolgt, einen unterschiedlichen Typ besitzen. Definiert sind folgende Typen:

Parametertyp	Bedeutung	Beispiel
In	Eingabeparameter – die Parameter werden von der Engine an die Anwendung übergeben, können aber nicht zurückgegeben werden.	<ul style="list-style-type: none"> - Statische Texte, die auf der Eingabemaske als Hinweise angezeigt werden sollen. - Runtime-Parameter für Anwendungen, die aber keine weitere Auswirkung auf die Variablen haben.
Out	Ausgabeparameter – die Parameter werden aus der Anwendung in den Prozess übergeben, ohne vorher übergeben worden zu sein.	<ul style="list-style-type: none"> - Eingangsvariablen, die von der Anwendung beim Starten eines Prozesses über Out-Parameter direkt übergeben werden. - Erstinitialisierung von Variablen
In / Out	Parameter, die zur Bearbeitung an eine Aktivität übergeben werden und dort geändert werden können.	<ul style="list-style-type: none"> - Parameter, die mit Eingabefeldern verknüpft und zur Veränderung vorgesehen sind. - Parameter zur Validierung und Bearbeitung in einer Aktivität

Im enaio® Workflow ist die am häufigsten genutzte Anwendung der enaio® client. Deshalb sind die Konfigurationsmöglichkeiten hierfür komfortabler ausgebaut. Die Interaktion zwischen der Workflow-Engine und dem Anwender erfolgt über Eingabemasken, die bereits aus dem DMS-Bereich bekannt sind. Diese Masken können auch hier graphisch erstellt werden und dann mit dem Prozessmodell verknüpft werden. Dabei agieren die Datenfelder auf den Eingabemasken als Anwendungsparameter und können den Variablen zugeordnet werden.

Im Beispiel des Bestellworkflows würde also hier pro Aktivität eine Eingabemaske erstellt werden, in die der jeweilige Anwender seine Daten eingeben kann. Faktisch kann eine Maske auch für mehrere Aktivitäten verwendet werden.



Im Prozessmodell würden also insgesamt vier Anwendungen (pro Aktivität eine) eingerichtet werden, die mit den jeweils vorgesehenen Masken verknüpft würden. Die Anwendungen können benannt werden, es macht Sinn, ihnen den gleichen Namen wie der Aktivität zu geben.

Oben wurde bereits eine für den Prozessablauf wichtige Variable „Wert“ benannt, da an ihr entschieden werden soll, ob der Vorgang noch durch den Geschäftsführer genehmigt werden muss, also eine entsprechende Transition zur Aktivität „Prüfung Geschäftsführer“ vorgenommen werden muss. Da die Eingabedaten aus dem Bestellvorgang an allen Schritten sichtbar sein sollen, müssen im Modell Variablen hinterlegt werden, die durch die Aktivitäten manipuliert werden. In diesem Beispiel würden nun folgende Variablen definiert werden:

Variable	Datentyp	Bedeutung
Artikel	String	Bezeichnung des Artikels
Kategorie	String	Kategorie des Bestellartikels (etwa Büroartikel, EDV...)
Wert	Integer	Wert in €
Genehmigt Abteilungsleiter	Integer	Gibt an, ob die Bestellung durch den Abteilungsleiter genehmigt wurde (Kontrollkästchen)
Genehmigt Geschäftsführer	Integer	Gibt an, ob die Bestellung durch den Geschäftsführer genehmigt wurde (Kontrollkästchen)
Bearbeitungsvermerke	String	Text für Bemerkungen

Die Variablen können beliebige Namen tragen, sinnvoll ist jedoch eine ähnliche Bezeichnung wie auf der Maske.

Pro Anwendung werden nun wiederum Parameter festgelegt, die mit den Prozessvariablen korrespondieren. Über den Typ der Parameter kann nun festgelegt werden, ob die Werte an der entsprechenden Aktivität auch bearbeitet werden können. So ist es im Beispiel sinnvoll, beim Einkauf die Änderungsmöglichkeit für die Parameter „genehmigt Abteilungsleiter“ und „genehmigt Geschäftsführer“ schreibgeschützt zu setzen, also als In-Parameter zu definieren.

Im enaio® Workflow werden standardmäßig die Datenfelder auf den Masken als Anwendungsparameter angeboten.

Schnittstellenvariablen von Prozessen

Eine Besonderheit in der Verwaltung von Variablen sind so genannte Schnittstellenvariablen von Prozessen. Diese haben die Eigenschaft, dass sie das Interface der Prozesse darstellen. Die Variablen können also als Eingabe- bzw. Ausgabeparameter von gesamten Prozessen angesehen werden. Dieser Fall tritt in folgenden zwei Situationen auf:

- § Starten eines Prozesses mit Übergabe von Variablen
- § Beendigung eines Prozesses und Übergabe von Variablen an die aufrufende Instanz

Schnittstellenvariablen werden vorwiegend in der Interprozesskommunikation von mehreren unabhängigen Prozessen verwendet. Dabei ist es möglich, dass die Ergebnisse eines bestimmten Prozesses in Form von Variablen an den nachgeordneten Prozess übergeben werden. Somit können die Modelle unabhängig voneinander erstellt und betrieben werden.

Im Beispiel wäre es also denkbar, dass nach der Aktivität „Einkauf“ und der somit vorgesehenen Beendigung des Prozesses definierte Variablen an einen weiteren Prozess im Warenwirtschaftssystem übergeben werden sollen. Schnittstellenvariablen dienen weiterhin im so genannten Posteingangsszenario der Übergabe von Daten zwischen enaio® capture zum Workflowsystem.

Speicherung von Prozessmodellen

Wie oben schon vermerkt, wird ein Prozessmodell in einer Metasprache beschrieben und im System gespeichert. Für die Beschreibung von Prozessmodellen existieren verschiedene Standards. Im enaio® Workflow wird die XML-Process Definition Language der Workflow Management Coalition verwendet. Wie der Name schon andeutet, werden alle zum Modell gehörigen Informationen in XML dargestellt. Das ermöglicht einen schnellen Zugriff, eine strukturierte Darstellung und ein hohes Maß an Kompatibilität.

Spezifika des enaio® Workflow werden gemäß Standard als erweiterte Attribute dargestellt. Darunter fallen insbesondere Verknüpfungen mit den Workflowmasken, Attribute zur Anbindung der digitalen Signatur und zur Darstellung der Workflowakte.

Zum Austausch der Prozessmodelle stehen zwei Möglichkeiten zur Verfügung, die im Prozessdesign bzw. im Kapitel zum Betrieb und Administration näher beschrieben sind. Es handelt sich hier einerseits um das Speichern des reinen XML-Modells ohne aufgelöste Referenzen auf die Masken und die Organisationsstruktur und andererseits um die Import- Exportfunktionalität.

Modellverwaltung

Im laufenden Betrieb durchlaufen Prozessmodelle oft unterschiedliche Bearbeitungsstadien. Diese bestehen aus:

- § Analyse der Organisation und der Arbeitsabläufe
- § Design der Modelle
- § Bereitstellung für den Betrieb
- § Freigabe für den laufenden Betrieb
- § Verwendung für bereits gestartete Prozesse

Damit diese unterschiedlichen Bearbeitungsstadien durch das Workflowmanagementsystem korrekt begleitet werden, ist eine integrierte Modellverwaltung notwendig. Diese dient dazu, verschiedene abgegrenzte Projekte voneinander zu unterscheiden und innerhalb der Projekte eine Versionierung der Prozessmodelle zu ermöglichen.

Im enaio® Workflow werden die Prozessmodelle in Workflowprojekte und Workflowfamilien organisiert.

Ein **Workflowprojekt** besteht aus einem oder mehreren Unterprojekten oder Workflowfamilien und fasst thematisch zusammenhängende Prozessmodellgruppen zusammen.

Eine **Workflowfamilie** besteht aus einem oder mehreren einzelnen Prozessmodellen. Die Prozesse innerhalb einer Familie können unterschiedliche Bearbeitungsstadien haben, es kann jedoch nur genau ein aktives Prozessmodell geben. Über das Konstrukt der Workflowfamilie kann eine Versionierung von Modellen ermöglicht werden, wobei die Verwaltung der Versionen beim Anwender liegt.

Nachfolgend eine Auflistung der unterschiedlichen Bearbeitungsstadien eines Prozessmodells im enaio® Workflow.

Bearbeitungsstadium	Beschreibung
Zur Bearbeitung gesperrt	Das Modell wird aktuell von einem Bearbeiter erstellt oder überarbeitet. Andere Benutzer haben keinen schreibenden Zugriff auf das Modell. Das Modell ist im System ausgecheckt.
Zur Bearbeitung freigegeben	Die Bearbeitung des Modells ist auf einem Zwischenstand abgeschlossen. Das Modell ist im System eingchecked. Es kann nun wieder bearbeitet oder für den laufenden Betrieb verwendet werden.
In Benutzung	Das Modell befindet sich im Echtbetrieb und ist von der Workflow-Engine geladen. Gewöhnlich wird dieser Status gesetzt, wenn bei einem Wechsel von Modellversionen bereits gestartete Prozesse mit der Version zu Ende laufen sollen, aber keine neuen Prozesse von diesem Modell gestartet werden dürfen.

Aktiv	Das Modell befindet sich im Echtbetrieb. Von diesem Modell können neue Prozesse durch die berechtigten Benutzer gestartet werden.
-------	---

Organisationsstruktur

In Unternehmen mit einer größeren Anzahl von Mitarbeitern existieren immer Strukturen, mit Hilfe derer die Unternehmensaufgaben organisiert, verteilt und bearbeitet werden. Diese Strukturen lassen sich unterscheiden in organisatorische Gruppierungen von Mitarbeitern, etwa nach Standorten, Abteilungen etc. und auch bezogen nach Aufgabenbereichen, wie z.B. Produktion, Vertrieb, Entwicklung. Meist existieren beide Strukturen nebeneinander.

Aus diesen Strukturen lassen sich wichtige Informationen für die Organisation von Arbeitsabläufen ableiten. Eine zentrale Frage besteht darin, wer für die Bearbeitung welcher Aufgabenbereiche zuständig ist. In einem Workflowmanagementsystem wird dies mit dem oben eingeführten Begriff der **Rolle** abgebildet.

Die Rollen leiten sich entweder aus den hierarchischen Strukturen des Unternehmens oder aus den Aufgaben her, die die Benutzer in der Rolle vornehmlich bearbeiten. So können etwa folgende Rollen definiert werden:

Organisatorischer Bezug

Mitarbeiter des Bereiches Produktion

Geschäftsführer

Mitarbeiter am Standort Berlin

Aufgabenorientierter Bezug

Antragsteller einer Bestellung

Prüfung eines Auftrages

Bearbeitung eines Antrages

Wie oben bereits bemerkt, kann ein realer Anwender mehreren Rollen zugeordnet sein, eine Rolle hingegen kann mehrere Mitglieder haben. Stellt man sich nun die Vielzahl der möglichen Geschäftsprozesse und die damit verbundene Anzahl an einzelnen Aktivitäten und Rollen vor, so ist es unerlässlich, durch das System eine Erleichterung zur Vereinfachung der Abbildung der Strukturen zu schaffen.

Gewöhnlich werden in größeren Unternehmen die Mitarbeiterstrukturen durch einen Verzeichnisdienst (z.B. LDAP) verwaltet, in dem sowohl die organisatorische Zugehörigkeit, als auch Rollen und Attribute der Mitarbeiter abgebildet werden.

Aufgabe eines Workflowmanagementsystems ist es nun, die Möglichkeit zu schaffen, die Strukturen in das eigene System so zu überführen, dass im Endeffekt eine Zuordnung der Mitarbeiter zu Rollen möglich wird. Dadurch soll erreicht werden, dass dem Anwender genau die Aufgaben oder Aktivitäten vorgegeben werden, die im Prozessmodell für die Rolle vorgesehen sind.

Hierarchische und aufgabenbezogene Strukturen eines Unternehmens werden in einer so genannten **Organisationsstruktur** zusammengefasst. Da Unternehmen meist völlig unterschiedlich aufgebaut sind, ist es notwendig, eine unternehmensbezogene Strukturierung zu ermöglichen, d.h. Gruppen von gleichartigen Elementen der Unternehmensstruktur zu klassifizieren und nach vorgegebenen Regeln zu beschreiben.

Eine Organisationsstruktur besteht einerseits aus Organisationsklassen, aus den einzelnen Elementen der Struktur und der Zuordnung der Elemente untereinander. Elemente einer Organisationsstruktur bezeichnet man als **Organisationseinheiten**. Beispiele für Organisationseinheiten sind etwa: Bereich Produktion, Sekretariat oder auch Mitarbeiter Meier-Schulze.

Organisationsklassen beschreiben den Aufbau eines bestimmten Typs von Organisationseinheiten. Die Beschreibung erfolgt durch bestimmte Attribute, die alle Organisationseinheiten dieses Typs besitzen. Beispiele für Organisationsklassen sind: Land, Abteilung, Team, Person. Die Organisationsklassen bilden also gewissermaßen die Metastruktur einer Organisationsstruktur.

Für alle Organisationsklassen können bestimmte **Attribute** beschrieben werden, die die einzelnen Elemente genauer spezifizieren. Jede Organisationsklasse hat ein festgelegtes Attribut, den Namen des Elementes. Darüber werden die einzelnen Elemente einer Klasse spezifiziert. Attribute können beim Design der Organisationsstruktur vergeben werden. Attribute sind wichtig für den Betrieb von Workflowmanagementsystemen, da auf die Werte während der Bearbeitung von Prozessen zugegriffen und somit der Verlauf des Prozesses beeinflusst werden kann. Insbesondere bei der Definition von Bedingungen in Transitionen kann der Zugriff auf Attribute der beteiligten Anwender sehr hilfreich sein.

In der folgenden Tabelle sollen Beispiele für Organisationsklassen und zugehörige Attribute aufgeführt werden. Die aufgeführten Organisationsklassen sind im enaio® Workflow bereits vordefiniert, können aber beliebig erweitert werden.

Organisationsklasse	Attribute
Land	Kontinent

	Vertriebsgebiet Amtl. Sprache
Organisation	Gründungsjahr Rechtsform Gemeinnützig
Abteilung	Produzierend Standort Kostenstellenbereich
Person	Vorname Nachname Geburtsdatum Einstellungsdatum Verfügungsrahmen

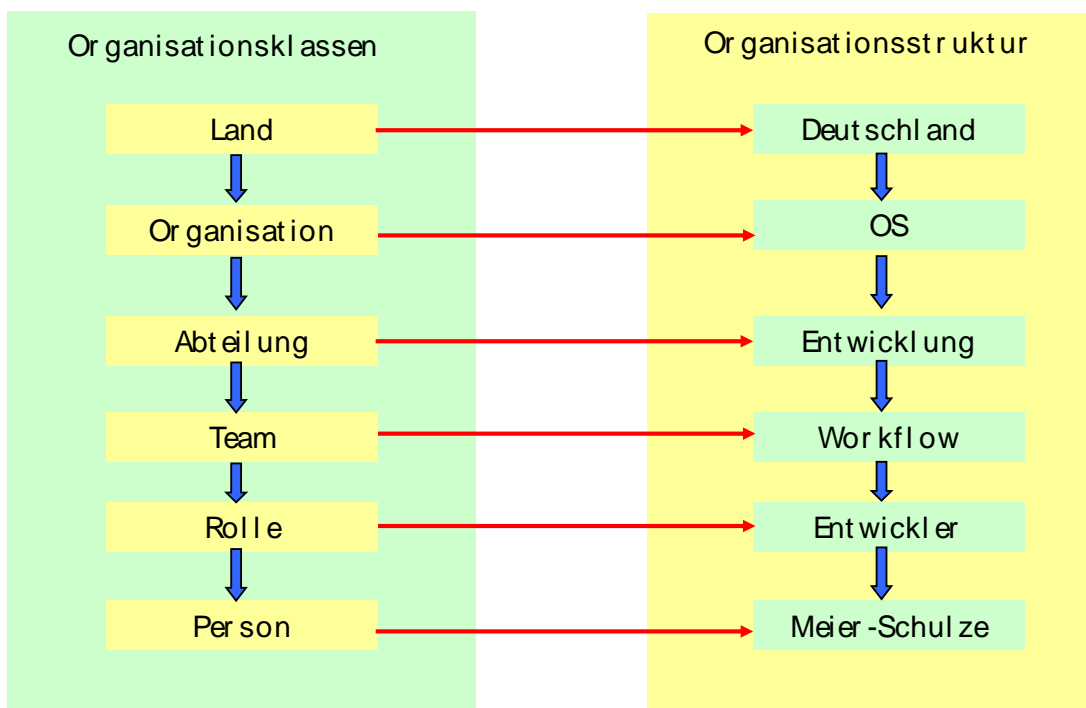
Die Rolle ist eine spezielle Organisationsklasse, die im Workflowmanagementsystem vordefiniert ist.

Weiterhin kann über die Organisationsklassen festgelegt werden, in welcher Form die einzelnen Organisationseinheiten einander zugeordnet werden können, d.h. welche Arten von Hierarchien zulässig sind. So kann etwa definiert werden, dass eine Person immer einer Abteilung zugewiesen werden muss, niemals aber direkt einem Land.

Organisationseinheiten sind die Instanzen der Organisationsklassen, also deren spezielle Ausprägungen. Die einzelnen Elemente erben von der Organisationsklasse die Definition der Attribute. Die Werte der Attribute werden hingegen für das einzelne Element festgelegt. So wird etwa der Kostenstellenbereich für die Abteilung Produktion mit dem Wert 3124-3152 oder das Geburtsdatum der Person Meier-Schulze mit dem Wert 12.03.1962 belegt.

Die Organisationseinheiten können hierarchisch angeordnet werden, wobei Mehrfachzuordnungen zu übergeordneten Organisationseinheiten möglich sind. Das tritt beispielsweise dann auf, wenn Personen aus organisatorischen Gründen Mitglieder in mehreren Abteilungen sind.

Die Beziehung zwischen Organisationsklassen und Organisationseinheiten sollen anhand der folgenden Graphik verdeutlicht werden.



Für Workflowmanagementsysteme und Verzeichnisdienste wird gewöhnlich die Darstellung einer Organisationsstruktur gemäß dem X.500-Standard verwendet. Dabei müssen die Organisationseinheiten in Form eines zusammenhängenden, orientierten, azyklischen Graphen dargestellt werden können. Oft wird hier die Bezeichnung X.500-Baum verwendet, wobei der Graph aufgrund der möglichen Mehrfachzuordnung zu übergeordneten Instanzen eigentlich kein Baum im Sinne der Graphentheorie ist.

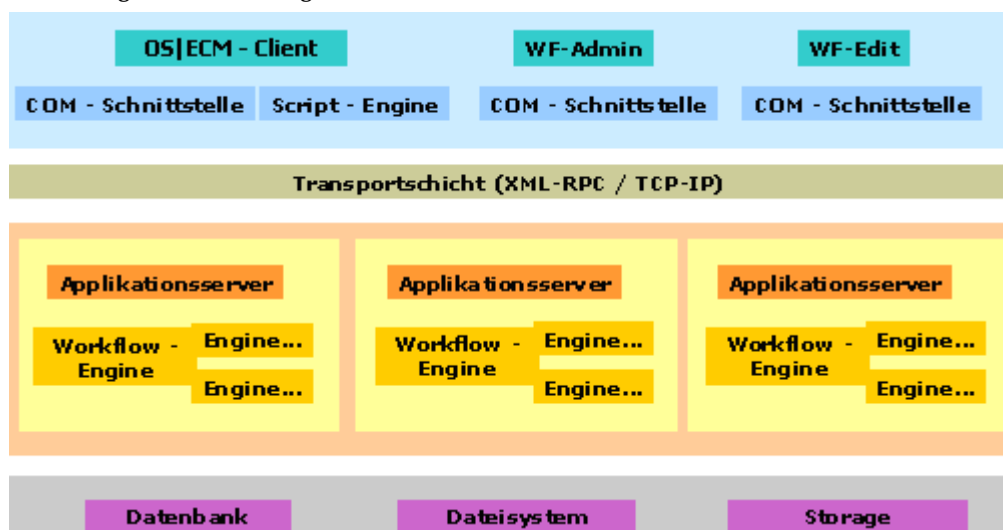
Architektur

Die Workflow-Engine ist integrierter Bestandteil des enaio® Applikationsservers. Sie wird als so genannter Exekutor (oxjobwfm.dll) geladen. Die gesamte Kommunikation mit den Workflow-Clients erfolgt über Serverjobs. Damit hat keine Anwendung direkten Zugriff auf Prozesse, Modelle und weitere Datenstrukturen innerhalb des Workflowmanagementsystems. Also läuft die Workflow-Engine in der Businessschicht im 3-Tier Architekturmodell. Dadurch kann der Workflow durch Loadbalancing der Applikationsserver fast beliebig skaliert werden.

Clientseitig erfolgt die Integration der Workflow-Jobs über folgende Module:

- § oxwfcnt.dll: COM-Bibliothek für die Darstellung der Schnittstelle zur Anfrage von startbaren Prozessen, Inhalt von Eingangskörben, Bearbeitung von Aktivitäten, Administrationsjobs etc.
- § oxwfedit.dll: COM-Bibliothek für die Darstellung der Schnittstelle zum Prozessdesign und zur Verwaltung der Organisationsstruktur
- § oxwfjobc.dll: Bibliothek zur Abbildung der Funktionen in XML-RPC Syntax, zur Durchführung der Workflow-Jobs und zur Aufbereitung der Ergebnisse

In nachfolgender Zeichnung soll die Architektur nochmals verdeutlicht werden.



Zur Job-Kommunikation mit der Workflow-Engine wird nur genau ein Job verwendet, der alle Funktionsparameter und Ergebnisse als XML-Struktur kapselt. Somit ist es zur Anbindung von Fremdsystemen auf der Transportebene hinreichend, einen Funktionsaufruf auf der Basis von XML-RPC zu implementieren.

Zusätzlich zum Jobaufruf, über den direkt Funktionen der Workflow-Engine aufgerufen werden, also etwa das Starten eines neuen Prozesses oder die Speicherung eines Modells, kann der Applikationsserver den Client über den Notification-Kanal aktiv über Veränderungen benachrichtigen. Dies erfolgt vorwiegend dann, wenn eine neue Aktivität in den Eingangskorb des Benutzers gestellt wurde.

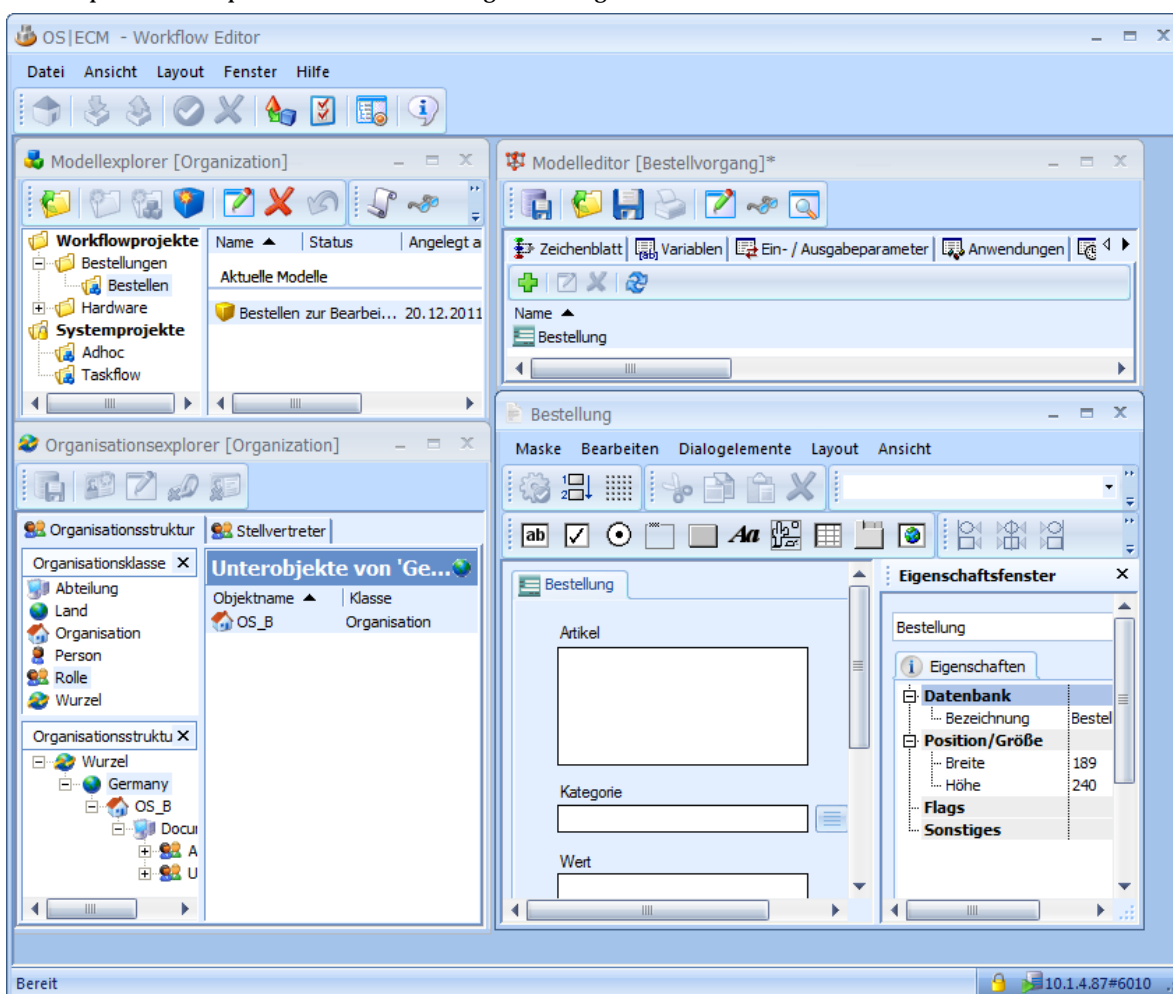
In diesem Fall sendet der Applikationsserver eine Nachricht an den Client, woraufhin dieser seinen Eingangskorb aktualisieren kann. Somit kann vermieden werden, dass alle gestarteten Clients regelmäßig bei der Workflow-Engine den Inhalt des Eingangskorbes abfragen müssen, wodurch die Netzlast verringert wird. Technisch besteht aber nur eine aktive TCP-Verbindung und der Server nutzt diese Verbindung, um dem Client die Benachrichtigung zukommen zu lassen. Es wird also keine zusätzliche Verbindung zwischen Server und Client geöffnet.

Prozessdesign

Allgemein

In diesem Abschnitt wird schrittweise das Design eines Geschäftsprozesses mit den Tools des enaio® Workflow erläutert. Die oben theoretisch gefassten Grundlagen werden anhand praktischer Beispiele umgesetzt. Das Prozessdesign erfolgt vorwiegend mit dem enaio® editor-for-workflow, der folgende Funktionsbereiche zur Verfügung stellt:

- § Abbildung der Organisationsstruktur
- § Maskendesigner für Workflowmasken
- § Design des Prozessmodells
- § Import- und Export von Modellen und ganzen Organisationsstrukturen



Alle in diesem Kapitel beschriebenen Anwendungsfunktionalitäten beziehen sich auf den enaio® editor-for-workflow. Weitere Programme, die im Zusammenhang mit dem Betrieb des enaio® Workflow eine Rolle spielen, werden im Abschnitt 'Betrieb und Administration' genannt.

Die Programme werden bei der Installation als Bestandteil der Administrationskomponenten in das Verzeichnis \clients\admin installiert.

Modellierung des Geschäftsprozesses

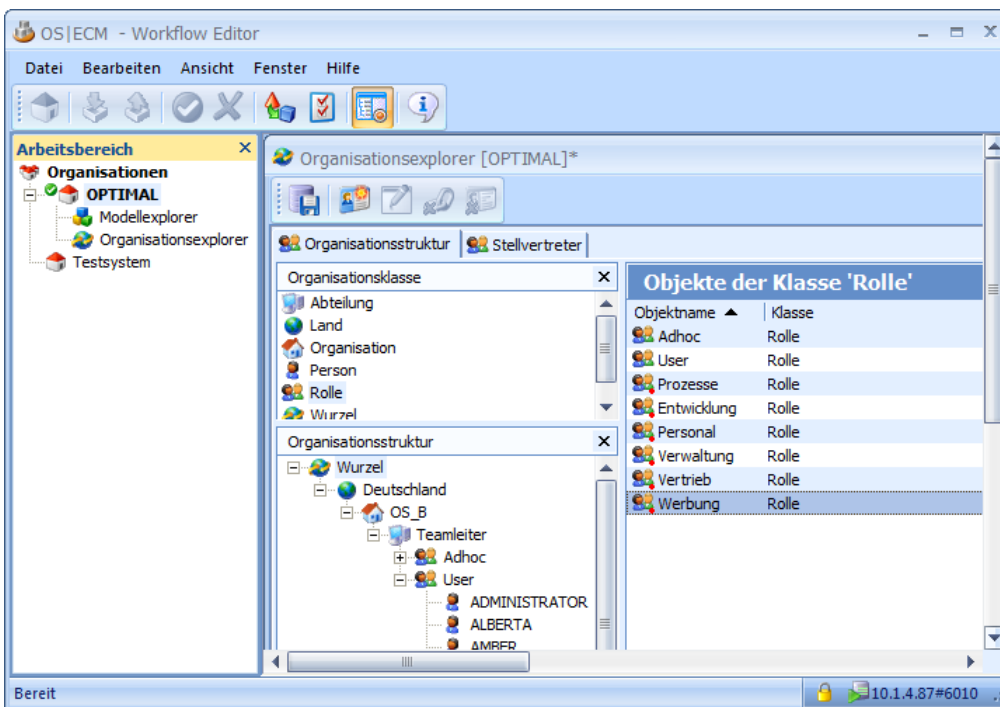
Wie oben schon vermerkt wurde, ist es für den effektiven Betrieb eines Workflowmanagementsystems notwendig, die Geschäftsprozesse vor der Modellierung genau zu analysieren. Dabei müssen folgende Punkte geklärt sein:

- § Wer nimmt am Geschäftsprozess teil?
- § Aus welchen Teilschritten besteht der abzubildende Geschäftsprozess?
- § Wie lassen sich die Teilschritte in Form von Aktivitäten (mit Eingabemasken) abbilden?
- § Wer darf den Geschäftsprozess starten?
- § Wer darf die einzelnen Aktivitäten bearbeiten?
- § Wer darf den Ablauf des Prozesses mit verfolgen?
- § Gibt es eine Bindung zum DMS und wie sieht diese an den einzelnen Aktivitäten aus?
- § Welche Übergänge gibt es zwischen den einzelnen Aktivitäten?
- § Unter welchen Bedingungen gelten die Übergänge?
- § Welche Rücksprungspunkte gibt es, wenn eine Aktivität nicht bearbeitet wurde?
- § Was soll am Ende des Prozesses mit den Dokumenten der Workflowakte passieren?
- § Welche Aktionen (Events) sollen bei der Bearbeitung von Aktivitäten automatisch ausgeführt werden?
- § Welche Fristen müssen beachtet werden?

Im Anhang befindet sich ein Analysebogen für Geschäftsprozesse, der die Beantwortung der Fragen und damit das Prozessdesign erleichtert. Es wird empfohlen, vor dem eigentlichen Design des Prozesses im System eine Analyse anhand des Bogens durchzuführen.

Abbildung der Organisationsstruktur

Im System können über den enaio® editor-for-workflow mehrere Organisationen angelegt werden, die sich in der Bezeichnung unterscheiden müssen. Zu jeder Organisation gehören Organisationsklassen, die Organisationsstruktur und beliebig viele Workflowprojekte. Organisationen werden über das Kontextmenü auf dem Knoten „Organisationen“ hinzugefügt oder gelöscht.



Jeweils eine Organisation ist 'aktiv'. Nur Workflowmodelle, die der aktiven Organisation zugeordnet sind, können verwendet werden. Die aktive Organisation ist durch einen grünen Haken gekennzeichnet. Organisationen werden über das Kontextmenü aktiviert.

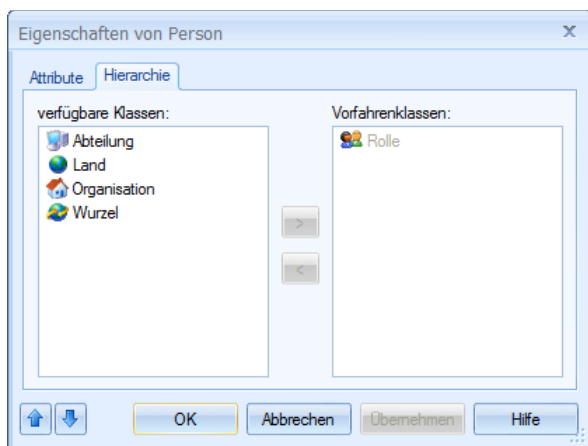
Wird eine Organisation durch das aktivieren einer anderen Organisation deaktiviert, werden alle laufenden Prozesse der Organisation sofort gestoppt. Die Benutzer erhalten im enaio® client einen entsprechenden Hinweis die Daten im Bereich 'Workflow' werden aktualisiert.

Nachdem eine neue Organisation angelegt wurde, können die Organisationsklassen bearbeitet werden. Vom System werden bereits sechs Organisationsklassen vorgegeben.

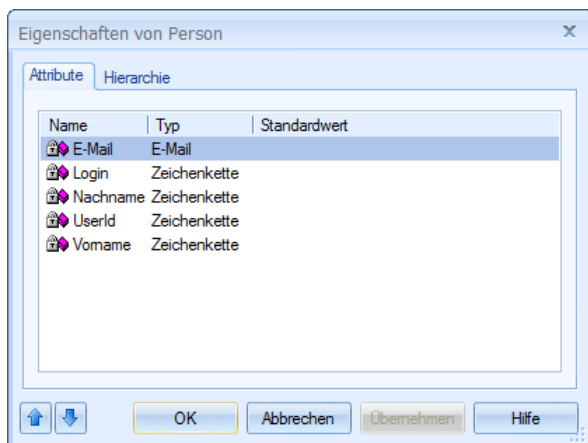
Organisationsklasse
Wurzel
Land
Organisation
Abteilung
Rolle
Person

Bei Bedarf können per Kontextmenü im Baum weitere Organisationsklassen angelegt werden. Beim Anlegen einer Organisationsklasse kann eine Bezeichnung angegeben werden.

Um festzulegen, an welcher Stelle die Instanzen der Organisationsklasse auftauchen dürfen, kann eine Vorfahrklasse angegeben werden.



Organisationsklassen können mit vordefinierten Attributen versehen werden. Alle Instanzen der Organisationsklassen können dann diese Attribute mit Werten belegen. Diese Attribute dienen dazu, um bestimmte Eigenschaften der Organisationseinheiten festzulegen. Als Beispiele von Attributen dienen etwa: Standort für Organisationen, Verfügungsrahmen für Abteilungen, oder das Geburtsdatum bei Personen.



Die Organisationsklassen Rolle und Person haben eine besondere Bedeutung. Bei den Aktivitäten im vorgesehenen Prozessmodell können als Teilnehmer Rollen und Personen eingetragen sein. Außerdem können Rollen und Personen Prozesse starten und deren Verlauf verfolgen, wenn dies im Modell entsprechend konfiguriert wurde. Damit eine Verbindung zwischen den Rollen und real existierenden Anwendern hergestellt werden kann, ist es notwendig, den betroffenen Rollen die angelegten Personen zuzuordnen und darüber hinaus den Personen einen entsprechenden Login-Namen zuzuweisen. Für die Organisationsklasse Person sind demnach die Attribute Login und UserId bereits vorbelegt.

Diese können nicht gelöscht werden. Das Hinzufügen, Bearbeiten und Löschen von Attributen erfolgt über das Kontextmenü.

Im nachfolgenden Dialog können dann die Eigenschaften des Attributes festgelegt und jeweils ein Standardwert angegeben werden.

Dialog 'Neues Klassenattribut' mit den folgenden Feldern:

- Name:
- ☒ Optional
- Typ:
- Standardwert:
- Buttons: OK, Abbrechen

Organisationsexplorer

Der Organisationsexplorer bietet eine sehr strukturierte Sicht auf die Beziehungen der Organisationseinheiten und erleichtert durch intuitive Bedienbarkeit die Erstellung der Organisationsstruktur.

Organisationsexplorer [OPTIMAL]*

Organisationsstruktur

- Organisation
- Land
- Abteilung
- Person
- Rolle
- Wurzel

Organisationsstruktur

- Wurzel
 - Deutschland
 - OS_B
 - Teamleiter
 - Adhoc
 - User

Objekte der Klasse 'Person'

Objektname	Klasse	UserId	E-Mail	Login
ADMINISTRATOR	Person	7C8421EDC4204B3D9889C62...		ADMINISTRAT...
ALBERTA	Person	A4B2F08C734946B396E1B3B6...		ALBERTA
AMBER	Person	BAAD688786F3474EB9BF977B...		AMBER
ANNE-SOPHIE	Person	658E4A2ECE0045018BDDFC5...		ANNE-SOPHIE
ARNAUD	Person	D240B686CA2C44C5814DFD9...		ARNAUD
CLAIRE	Person	87D0F2B8B5824C9295DC06D...		CLAIRE
CLAUDIA	Person	D6998CFD92AB4245AD7FC6C...		CLAUDIA
CÉCILE	Person	F00E8ABBC4AD47C7AC788F6...		CÉCILE
DANNY	Person	704A88533AF84648BA89B04...		DANNY
EUGENE	Person	982326A2FD1F4E7183999076...		EUGENE
FRANCK	Person	64DACFD1D17B47F6B616841...		FRANCK

Die eigentliche Definition der Organisationsstruktur erfolgt mit dem Organisationsexplorer. Hier werden die Instanzen der entsprechenden Organisationsklassen definiert und danach die Struktur aufgebaut. Die Definition der Organisationsstruktur erfolgt in zwei Schritten.

Anlage der Objekte

Dazu wird jeweils eine Organisationsklasse markiert und über das Kontextmenü oder die entsprechenden Buttons ein neues Objekt angelegt. Diesem Objekt muss dann eine Bezeichnung zugeordnet und dessen in der Organisationsklasse definierte Attribute belegt werden.

Wie oben schon angedeutet müssen die Attribute UserId und Login bei Anlage einer Person gefüllt werden. Dies erfolgt durch die unten beschriebene Zuordnung der Person zu einem DMS-Benutzer. Bei anderen Attributen hängt dies davon ab, ob in der Organisationsklasse das Attribut als optional definiert wurde.

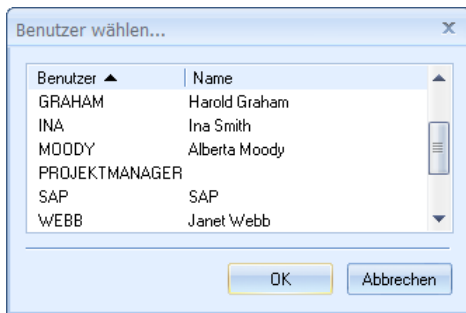
Eigenschaften von Jill

Attribute

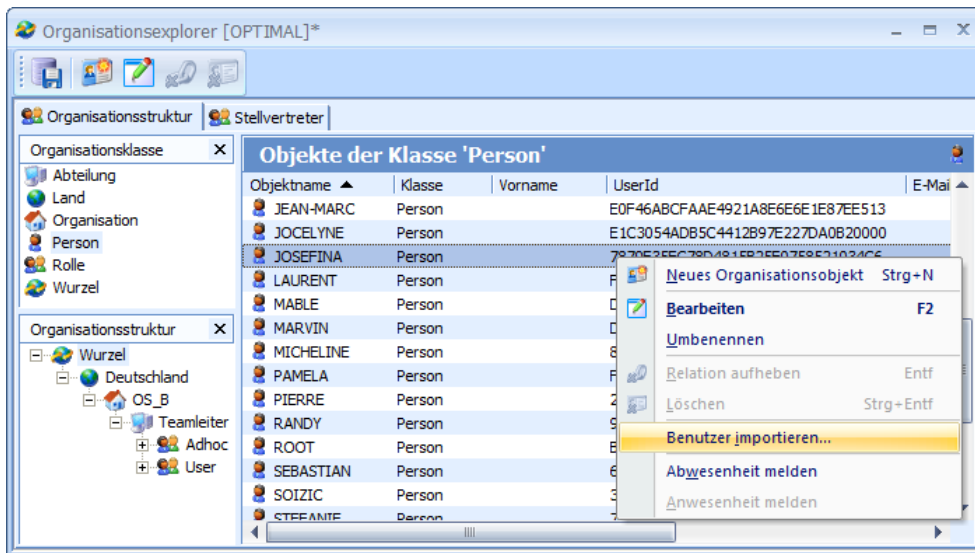
Name	Wert
E-Mail	jill@optimal-systems.de
Login	INA
Nachname	Schall
UserId	8C237171833E48FB870E5B95E87EAF5C
Vorname	Jill

Buttons: OK, Abbrechen, Übernehmen, Hilfe

Die Zuordnung zu einem DMS-Benutzer erfolgt durch einen Auswahldialog, in dem der Login-Name angezeigt wird.

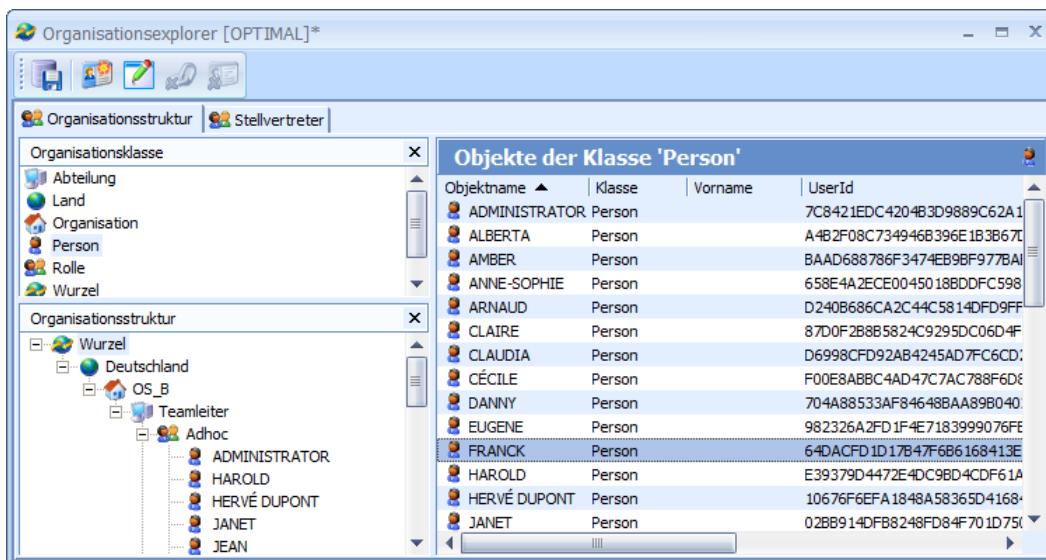


Zusätzlich besteht die Möglichkeit, automatisch für bereits konfigurierte DMS Benutzer eine entsprechende Person in der Organisation anzulegen. Dies geschieht über den Eintrag **Benutzer importieren...** im Kontextmenü der Objektsicht des Organisationsexplorers.



Zuordnung der Objekte zu den Organisationseinheiten

Nachdem die Objekte angelegt wurden, können diese in der Baumsicht den übergeordneten Organisationseinheiten zugeordnet werden. Zu diesem Zwecke kann eine Organisationsklasse markiert werden, woraufhin die vorhandenen Objekte in der Liste angezeigt werden. Diese können dann per Drag & Drop den gewünschten Organisationseinheiten zugeordnet werden. Objekte, die noch keiner Organisationseinheit zugeordnet sind, sind mit einem rot markierten Symbol in der Liste gekennzeichnet.



Da die Organisationsstruktur kein Baum, sondern ein azyklischer, gerichteter Graph ist, können Objekte an mehreren Stellen in der Ansicht auftauchen. Im Organisationsexplorer wurde wegen der Übersichtlichkeit trotzdem die

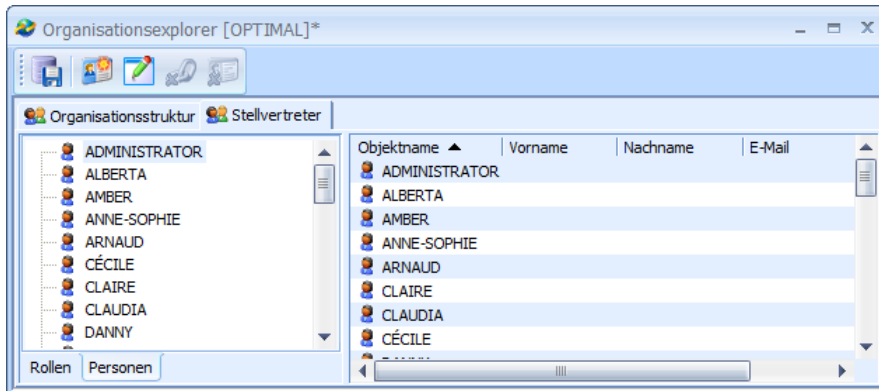
Baumansicht gewählt. Beim Drag & Drop wird überprüft, ob die Objekte gemäß der definierten Hierarchie der Organisationsklassen eingefügt werden können. Eine nicht vorgesehene Zuordnung wird unterbunden.

Stellvertretung

Stellvertretungen werden ebenfalls über den Organisationsexplorer eingerichtet. Personen und Rollen können Personen als Stellvertreter zugeordnet werden. Benutzer melden im enaio® client ihre Abwesenheit oder Anwesenheit. Sind alle Teilnehmer einer Aktivität abwesend gemeldet, werden die Aktivitäten in den Eingangskorb aller Stellvertreter angezeigt. Bereits personalisierte Aktivitäten werden ebenfalls übertragen.

■ Aktivitäten werden nicht an Stellvertreter von Stellvertretern übertragen. ■

Die Stellvertreterverwaltung kann über den Reiter **Stellvertreter** geöffnet werden.

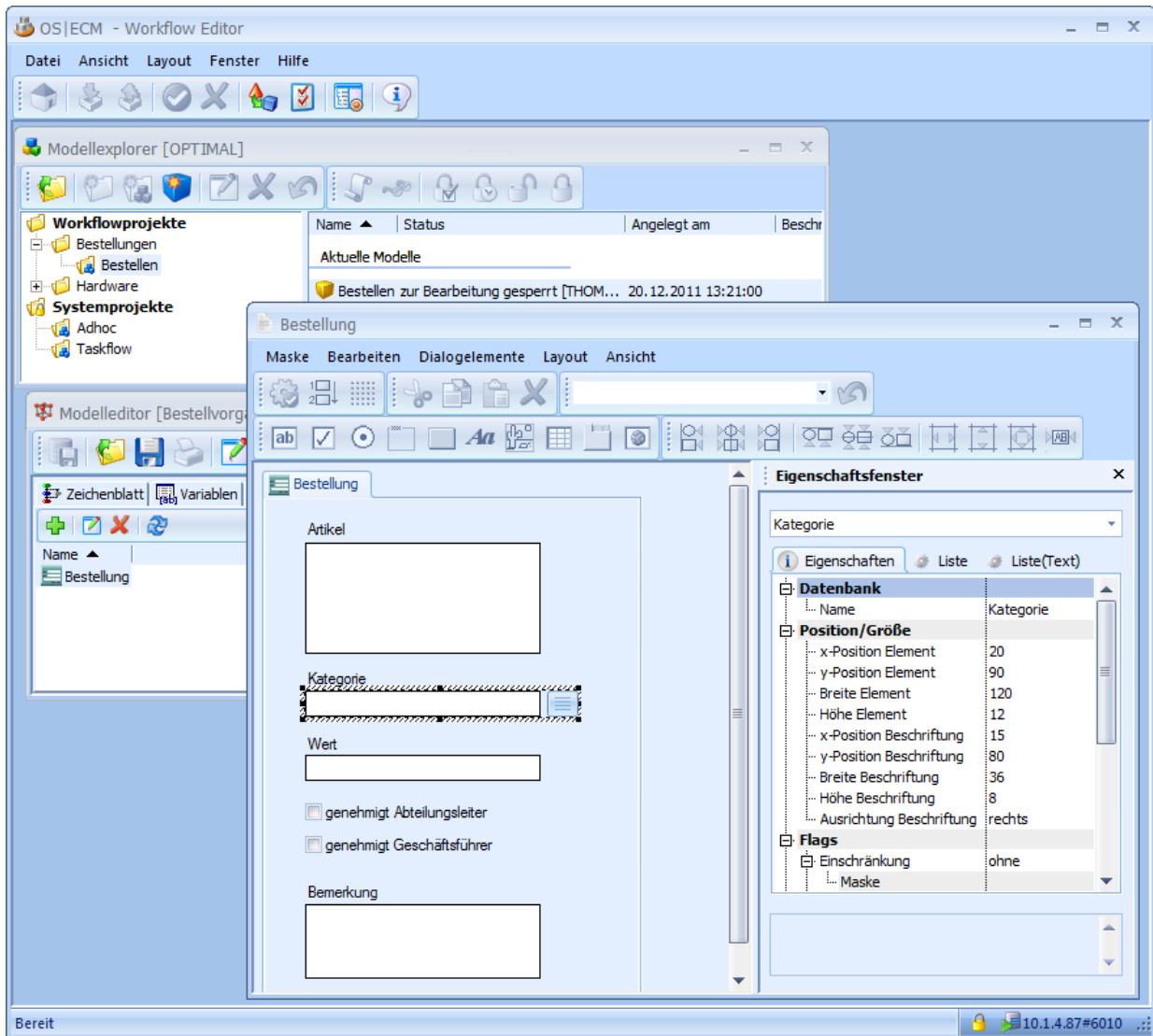


Per Drag&Drop können jeder Person und jeder Rolle Personen aus der Liste möglicher Stellvertreter zugeordnet werden.

Stellvertretungen gelten auch für Wiedervorlagen abwesender Benutzer.

Design der Workflowmasken

Das Design der Workflowmasken erfolgt mit dem enaio®-Workflow Editor auf gleiche Art und Weise wie im enaio®-Editor. Es stehen jedoch nur die Eigenschaften zur Verfügung, die nicht speziell auf das DMS abgestimmt sind.



Workflowmasken werden für ein Prozessmodell (Workflowmodell) erstellt, können aber einfach kopiert und so in anderen Modellen verwendet werden. Die Masken dienen später beim Prozessdesign als Interface von Anwendungen. Jedem Feld auf der Maske muss ein Parameter zugeordnet werden.

Masken enthalten Felder, Felder haben funktionale Eigenschaften, die über einen Eigenschaften-Dialog angegeben werden und graphische Eigenschaften, die über die grafische Oberfläche gestaltet werden.

Für Workflowmasken gibt es Größenvorgaben:

- § Masken dürfen maximal 750 Dialogeinheiten hoch und 1000 Dialogeinheiten breit sein. Sie werden mit einer minimalen Höhe von 180 Dialogeinheiten und Breite von 220 Dialogeinheiten im enaio® client angezeigt, auch wenn kleine Werte angegeben werden. Elemente auf der Maske, die sichtbar sein sollen, müssen vollständig auf der Maske positioniert werden.

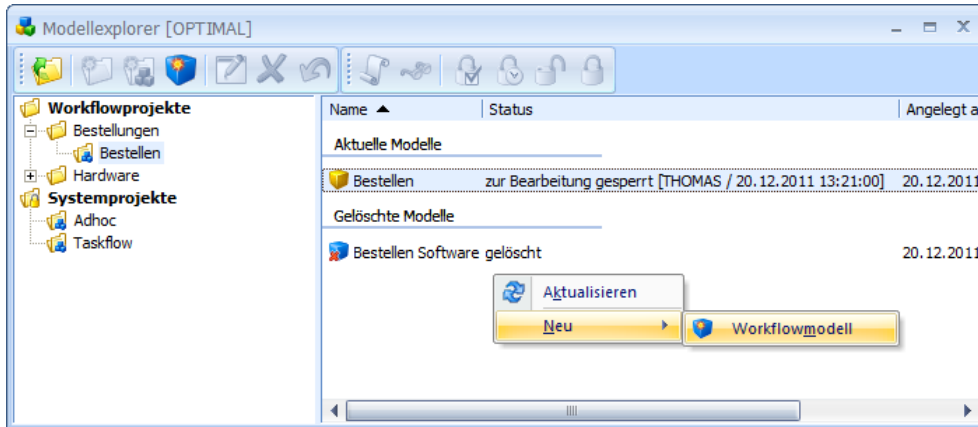
Abbildung des Modells

Die Abbildung des Modells erfolgt in mehreren Schritten, die nachfolgend dargestellt werden. Voraussetzung ist hier in jedem Fall eine genaue Vorstellung über den abzubildenden Geschäftsprozess.

Über den Arbeitsbereich des Workflow-Editors legen Sie Workflowprojekte an, einem Workflowprojekt ordnen Sie Workflowfamilien zu. Alle Daten zu einer Workflowfamilie werden im Modellexplorer bearbeitet.

Modellexplorer

Modelle werden innerhalb einer Workflowfamilie verwaltet. In einer bestehenden Workflowfamilie kann über das Kontextmenü des Modellexplorers oder die entsprechenden Buttons ein neues Modell angelegt werden.



Das Modell wird mit einer innerhalb der Familie eindeutigen Bezeichnung versehen.

Nach dem Anlegen eines neuen Modells wird es gleich im Modelleditor geöffnet. Es hat im Modellexplorer den Status 'zum Bearbeiten gesperrt'. Angegeben ist ebenfalls der Benutzer. Nur dieser darf das Modell bearbeiten oder freigeben. Andere Benutzer können allerdings das Modell schreibgeschützt öffnen oder eine Kopie erzeugen und bearbeiten.

Über Buttons auf der Symbolleiste des Modellexplorers kann der Status des Modells geändert werden (vgl. 'Modellverwaltung').

Aus einer Workflowfamilie kann nur jeweils ein Modell den Status 'aktiv' erhalten.



Bevor ein Modell den Status 'aktiv' erhält, sollte es überprüft werden. Überprüft wird dabei, ob alle notwendigen Angaben gemacht wurden. Ist das nicht der Fall, wird ein entsprechender Hinweis angezeigt.

Eine Überprüfung garantiert nicht, dass der Prozess fehlerfrei ist.

Löschen Sie ein Modell, wird es auf die Registerkarte **Gelöschte Modelle** übertragen. Die Daten sind noch nicht aus der Datenbank gelöscht. Sie können das Modell wiederherstellen oder endgültig löschen.

Über den Modellexplorer öffnen Sie ebenfalls den Eventeditor. Über den Eventeditor erstellen Sie Events. Ein Event ist ein Script, das einer Aktivität im enaio® client zugeordnet ist und aus dem enaio® client automatisch durch die Aktivität gestartet wird. So kann beispielsweise beim Weiterleiten durch ein Script eine Plausibilitätsprüfung erfolgen oder es können automatisch Daten ergänzt werden (vgl. S.51).

Modelleditor

Nach einem Doppelklick auf ein Modell wird der Modelleditor geöffnet.

Das Modell konfigurieren Sie über sieben Registerkarten:

§ Zeichenblatt

Auf dem Zeichenblatt werden grafisch die Aktivitäten und die Transitionen zwischen den Aktivitäten festgelegt. Je nach Arbeitsweise können Sie mit einem grafischen Entwurf des Modells beginnen oder zuerst Variablen und Anwendungen festlegen.

§ Variablen

Die Variablen werden definiert. Der Wert der Variablen wird durch die Aktivitäten geändert.

§ Ein- / Ausgabeparameter

Variablen können als Ein- / Ausgabeparameter definiert werden. Dann können externe Prozesse auf diese Variablen zugreifen.

§ Anwendungen

Anwendungen sind Workflowmasken mit Feldern. Anwendungen werden Aktivitäten zugeordnet. Festgelegt wird, welche Variable über ein Feld der Workflowmaske innerhalb einer Aktivität geändert werden kann.

§ Mahn- / Sperrfristen

Mahnfristen sollen sichern, dass Aktivitäten oder Folgen von Aktivitäten innerhalb eines bestimmten Zeitraums durchgeführt werden. Sperrfristen legen dagegen fest, dass Aktivitäten oder Folgen von Aktivitäten nicht vor Erreichen eines bestimmten Zeitraums abgeschlossen werden können.

§ MASKEN

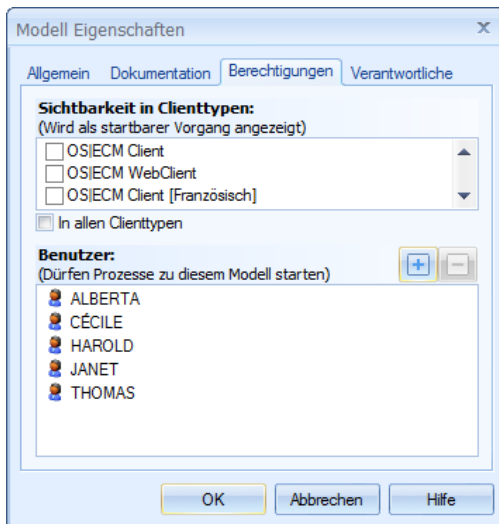
Masken erlauben die Eingabe von Daten in den Clientanwendungen.

§ Events

Ein Event ist ein Skript (VBScript für enaio® client, JavaScript für enaio® webclient), das einem Ereignis im Ablauf eines Workflowprozesses zugeordnet ist.

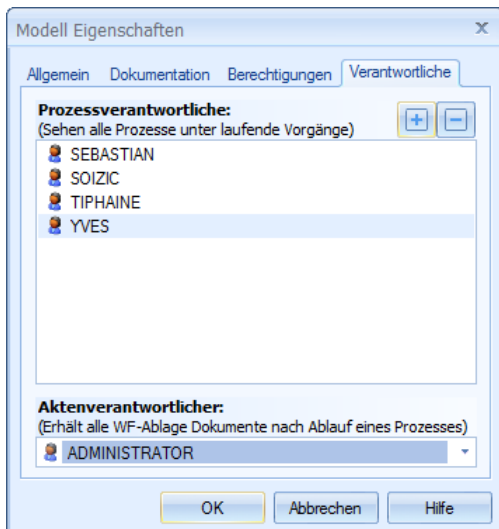
Zusätzlich geben Sie für jedes Modell an, welche Benutzer berechtigt sind, einen Vorgang dieses Modells zu starten und aus welchen Clients heraus Vorgänge dieses Modells überhaupt gestartet werden können.

Die gewünschten Benutzer und Clienttypen tragen Sie über den Reiter **Berechtigungen** des Eigenschaften-Dialogs des Modells ein.



Werden Prozessverantwortliche angegeben, können diese im enaio® client auf der Registerkarte **Laufende Vorgänge** Modellvorgänge einsehen und somit den Verlauf von Vorgängen verfolgen. Außerdem lässt sich ein Aktenverantwortlicher einrichten. Diese Person erhält die Dokumente, die nach Ablauf eines Vorgangs noch in der WF-Ablage liegen.

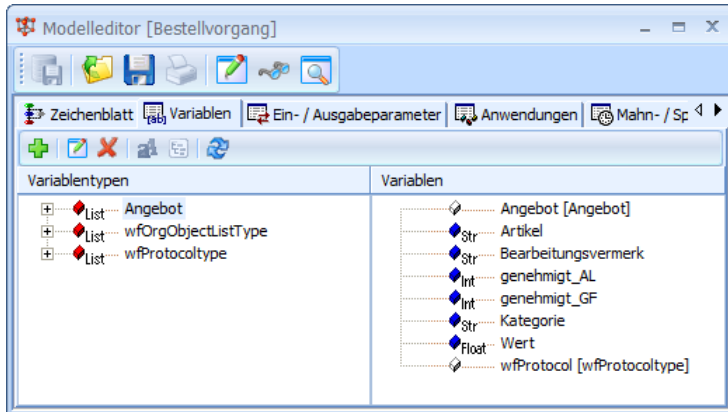
Die gewünschten Verantwortlichen tragen Sie über den Reiter **Verantwortliche** des Eigenschaften-Dialogs des Modells ein.



Variablen

Variablen beinhalten Daten, die während der Ausführung des Prozesses den Ablauf beeinflussen und durch Aktivitäten geändert werden können. Deshalb müssen diese zunächst im Modell definiert werden.

Die Variablen richten Sie im Modelleditor über die Registerkarte **Variablen** ein.



Voreingetragen ist die Variable `wfprotocol`. Diese Variable vom Typ 'List (Record)' dient der Protokollierung und wird einem Anwendungsparameter zugeordnet und über Skripte verwaltet.

Variablen sind typisiert, d.h. es erfolgt eine primäre Festlegung des Datenformats. Zur Verfügung stehen einfache, zusammengesetzte und nutzerdefinierte Typen von Variablen.

Variablenbezeichnungen dürfen keine Umlaute, Leer- oder Sonderzeichen beinhalten.

Einfache Typen

- String** beinhaltet eine Zeichenkette mit dem Standardzeichensatz. Die Längenbegrenzung ist durch das zugrunde liegende Datenbanksystem vorgegeben und bezieht sich auf die maximale Größe eines BLOB-Feldes (Größe ist abhängig vom verwendeten DBMS).
- Integer** stellt eine Ganzzahl im Bereich von -2^{31} bis 2^{31} dar
- DateTime** nimmt Datum/Zeit im jeweils unter Windows eingestellten Datumsformat auf
- Float** beinhaltet eine Gleitkommazahl

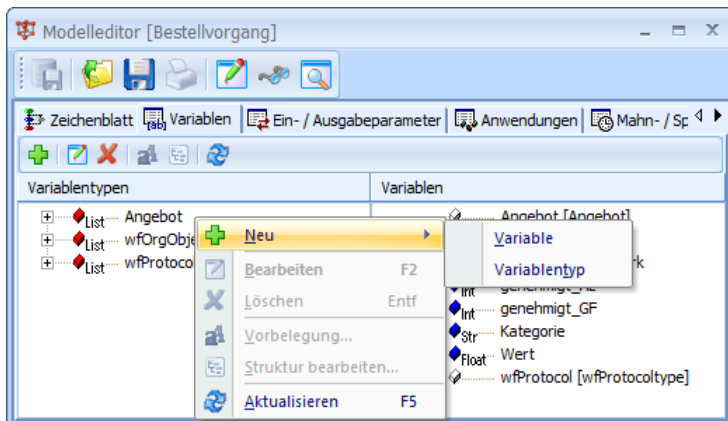
Zusammengesetzte Typen

- List** ist eine Liste von Daten einfachen, zusammengesetzten (außer Liste) oder nutzerdefinierten Typs
- Record** stellt eine zusammengesetzte Struktur mehrerer einfacher Typen dar

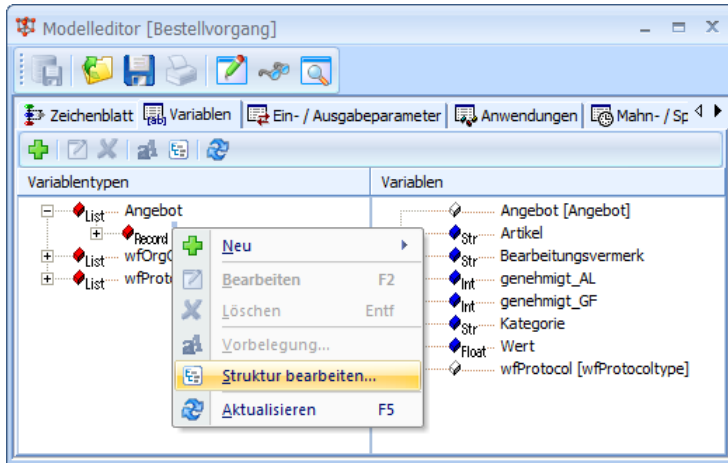
Benutzerdefinierte Typen

Durch das System wird die Möglichkeit gegeben, komplexe Datenstrukturen als Variable zu definieren. Dies kann erforderlich sein, wenn als Variablen nicht nur einfache Werte, sondern Strukturen, wie etwa Tabellen oder Listen von unterschiedlichen Einträgen übergeben werden sollen. Beispielsweise wird für die Protokollierung von einem nutzerdefinierten Typ ausgegangen, der aus einer Liste von Records besteht. Diese Variablentypen können durch den Anwender definiert werden.

Die Einrichtung der Variablen erfolgt über die Registerkarte **Variablen**.



Sie wählen über das Kontextmenü **Neue Variable**, geben eine Bezeichnung, eine optionale Beschreibung und den Typ an. Wurde als Typ der Variablen „Record“ angegeben, so kann die Zusammensetzung der Struktur über den Eintrag **Struktur bearbeiten** aus dem Kontextmenü definiert werden. Es erscheint dann ein Dialog, in dem die einzelnen Variablen, die zum Record zusammengesetzt werden sollen, festgelegt werden können.



Festlegung von benutzerdefinierten Typen

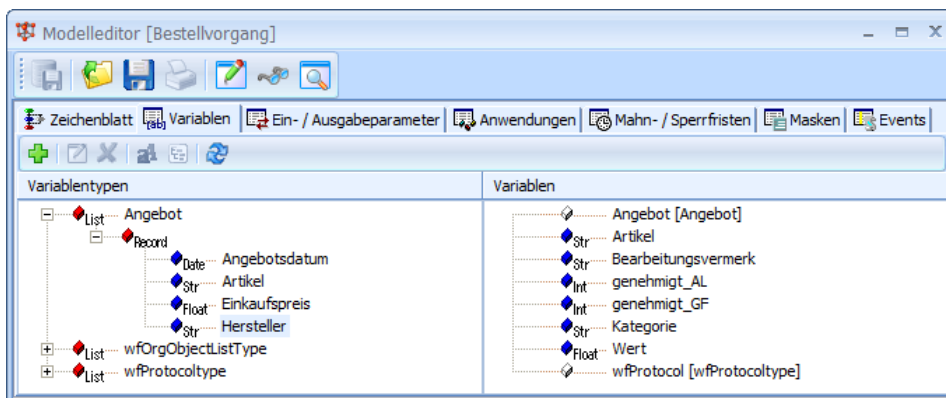
Im Beispiel könnte etwa eine Tabelle vorgesehen sein, in die ein Sachbearbeiter mögliche Angebote für die Bestellung eingeben kann. In diesem Falle würde eine Liste von Records definiert werden müssen. Jede Zeile der Tabelle nimmt einen Record auf, die Auflistung der einzelnen Zeilen stellt dann die Tabelle dar. So kann die gesamte Tabelle in einer einzigen Workflowvariable gespeichert werden.

Dazu wird ein neuer Variablentyp „Angebote“ vom Typ List (Record) erstellt. Diese Liste soll als Elemente Records aufnehmen, die in ihrer Struktur im folgenden Dialog festgelegt werden. Die Struktur soll ein Angebotsdatum, eine Artikelbezeichnung, den Hersteller und einen Angebotspreis enthalten. Also müssen der Recordstruktur Variablen der entsprechenden Typen hinzugefügt werden.

Das Hinzufügen von Elementvariablen eines Records erfolgt genauso wie das Hinzufügen einfacher Variablen. Im Record müssen Variablen eindeutig bezeichnet sein und von einem einfachen Typ sein. Die Verschachtelung von Records in anderen Records gleichen Typs ist nicht möglich.



Nachdem der neue Typ definiert ist, legen Sie eine entsprechende Variable an, indem Sie aus der Liste verfügbarer Typen den angelegten Record auswählen.



Ein- und Ausgabeparameter

Sollen Prozesse bereits mit vorgelegten Werten gestartet werden, so müssen so genannte Schnittstellenvariablen definiert werden. Dies kann beispielsweise in folgenden Fällen auftreten:

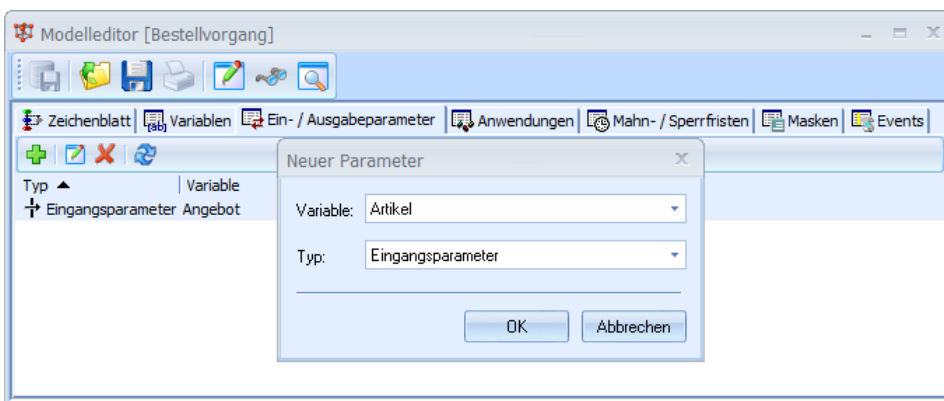
- § Starten von Prozessen aus enaio® capture über die Workflowstartkomponente
- § Starten von Prozessen aus Microsoft Outlook mit der Übergabe von E-Mail Daten
- § Verkettung von mehreren gleichberechtigten Prozessen

Mögliche Werte für den Schnittstellenmodus von Variablen sind:

- § Eingabeparameter:
die Variable wird beim Starten des Prozesses übergeben. Dies tritt dann auf, wenn eine externe Applikation einen Workflowprozess startet.
- § Ausgabeparameter:
die Variable wird nach Beendigung des Prozesses zurückgegeben.
- § Ein-/Ausgabeparameter:
die Variable wird beim Start des Prozesses übergeben und nach Beendigung wieder mit Werten belegt.

Nur die Variablen, die bereits auf der Registerkarte **Variablen** eingerichtet wurden, können zu Schnittstellenvariablen von Workflowmodellen werden.

Über die Registerkarte **Ein- / Ausgabeparameter** definieren Sie Variablen als Eingabeparameter, Ausgabeparameter oder Ein- und Ausgabeparameter.

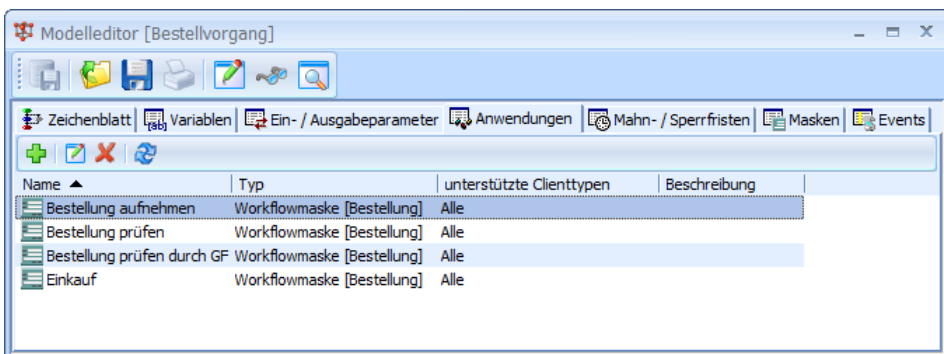


Anwendungen

Über Anwendungen erfolgt eine Bearbeitung der Variablen in den einzelnen Aktivitäten. Dies kann durch automatisch ablaufende Softwareinstanzen, im Allgemeinen aber durch Programme mit einer Anwenderinteraktion erfolgen. In enaio® ist die Benutzerschnittstelle und damit auch ein Großteil der Workflowanwendungen in den enaio® client integriert. Anwendungen werden standardmäßig mit Workflowmasken assoziiert, über die Änderungen an den Prozessdaten vorgenommen werden.

Wie in den Grundlagen erläutert, ist es notwendig, für Anwendungen Parameter festzulegen, über die der Datenaustausch zwischen der Workflow-Engine und der Anwendung erfolgt.

Die Einrichtung der Anwendungen erfolgt auf der Registerkarte **Anwendungen**.



Für alle Aktivitäten, die ein Benutzer ausführt, richten Sie eine Anwendung ein. Sofern verschiedene Aktivitäten über die gleiche Maske und die gleichen Anwendungsparameter ausgeführt werden können, kann eine Anwendung auch mehreren Aktivitäten zugeordnet werden.

Derzeit können im Workfloweditor nur Anwendungen eingerichtet werden, die sich auf eine Workflowmaske beziehen.

Nachdem eine Bezeichnung für die Anwendung eingegeben wurde, muss eine Maske ausgewählt werden, die vorher mit dem Maskendesigner angelegt wurde. Es empfiehlt sich, Masken, Anwendungen und Aktivitäten gleich zu benennen, wenn sie für dieselbe Aktivität vorgesehen sind.

Es können für die gleiche Aktivität je nach Clienttyp unterschiedliche Anwendungen definiert werden. Es ist aber auch möglich die gleiche Anwendung für alle möglichen Clienttypen festzulegen. Bei der Bearbeitung der Anwendung wird festgelegt, für welche Clienttypen die Anwendung definiert ist.

Sie können sowohl für unterschiedliche Sprachen Anwendungen erstellen, die die gleiche Funktion haben, aber Masken mit anderen Bezeichnungen, wie auch Anwendungen für den enaio® client oder den enaio® webclient, um den unterschiedlichen Umgebungen gerecht zu werden.

Festlegen der Anwendungsparameter

Nach der Anlage der Anwendungen müssen deren Parameter erstellt werden. Den Konfigurationsdialog öffnen Sie über das Kontextmenü.

Parameter	Typ
<input checked="" type="checkbox"/> Artikel	Ein-/Ausgangsparameter
<input checked="" type="checkbox"/> Bemerkung	Ein-/Ausgangsparameter
<input checked="" type="checkbox"/> Kategorie	Ein-/Ausgangsparameter
<input checked="" type="checkbox"/> Wert	Ein-/Ausgangsparameter
<input checked="" type="checkbox"/> genehmigt Abteilungsleiter	Ein-/Ausgangsparameter
<input checked="" type="checkbox"/> genehmigt Geschäftsführer	Ein-/Ausgangsparameter

Im Falle der Abbildung von Anwendungen auf Masken, also im Standardfall, korrespondieren die Anwendungsparameter mit den Feldern auf der Maske. Wichtig ist, dass für jedes Feld auf der Maske ein Anwendungsparameter existieren muss. Im Dialog werden die Feldnamen angezeigt.

Neben der Bezeichnung für den neuen Parameter muss auch ein Typ ausgewählt werden. Dieser kann folgenden Wert annehmen:

Ausgangsparameter der Wert wird von der Anwendung, also der Maske an das Workflowsystem übergeben. Der Wert des Parameters wird nicht initialisiert. Geeignet für Eingabefelder, die vorher nicht initialisiert wurden.

Eingangsparameter

Der Wert wird an die Anwendung / Maske übergeben, aber nicht an die Workflow-Engine zurückgeliefert. Geeignet für die Anzeige statischer Informationen oder für Felder, die an der Aktivität nicht mehr geändert werden sollen.

Hinweis: Eingangsparameter sind auf einer Maske nicht gekennzeichnet oder schreibgeschützt.

Ein-/Ausgangsparameter

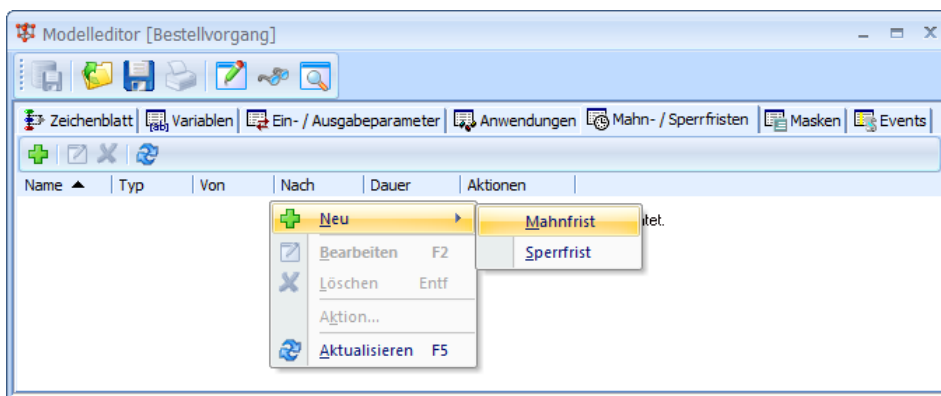
Der Wert wird an die Maske übergeben, kann dort geändert werden und wird dann an die Engine zurückgeliefert. Geeignet für Daten, die während der Aktivität geändert werden sollen.

Für jede Anwendung müssen alle Parameter definiert werden. Gewöhnlich kann man sich an die Definition der Workflowvariablen und die Eingabemasken halten. Zu beachten ist, dass für jedes Feld ein Parameter erstellt werden muss, es muss jedoch nicht jede Variable an die Masken übergeben werden und nicht jeder Parameter muss über Masken bearbeitet werden.

Mahn- und Sperrfristen

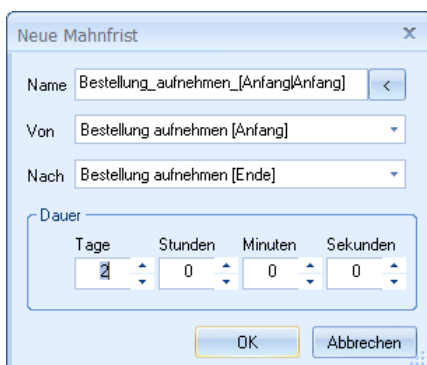
Mahnfristen sollen sichern, dass Aktivitäten oder Folgen von Aktivitäten innerhalb eines bestimmten Zeitraums durchgeführt werden. Zu einer Mahnfrist kann festgelegt werden, welche Aktion ausgeführt wird, wenn der festgelegte Zeitraum überschritten wurde.

Sperrfristen legen dagegen fest, dass Aktivitäten oder Folgen von Aktivitäten nicht vor Erreichen eines bestimmten Zeitraums abgeschlossen werden können.



Mahnfristen

Eine neue Mahnfrist wird über den Eintrag **Neue Mahnfrist** aus dem Kontextmenü eingerichtet.



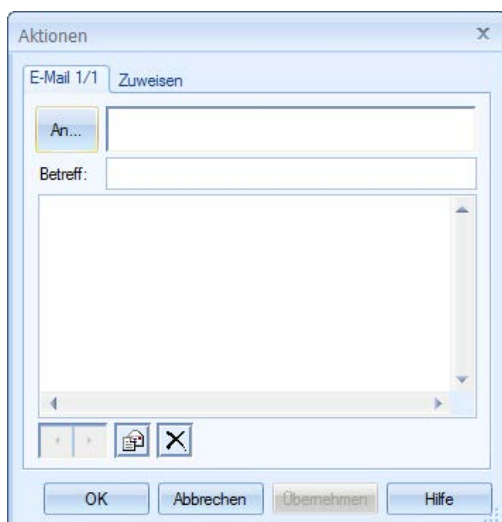
Für die Festlegung des Zeitraums kann von jeder Aktivität jeweils der Anfang oder das Ende gewählt werden.

Der Anfang einer Aktivität ist das Personalisieren im enaio® client, das Ende das Weiterleiten nach der vollständigen Bearbeitung der Aktivität.

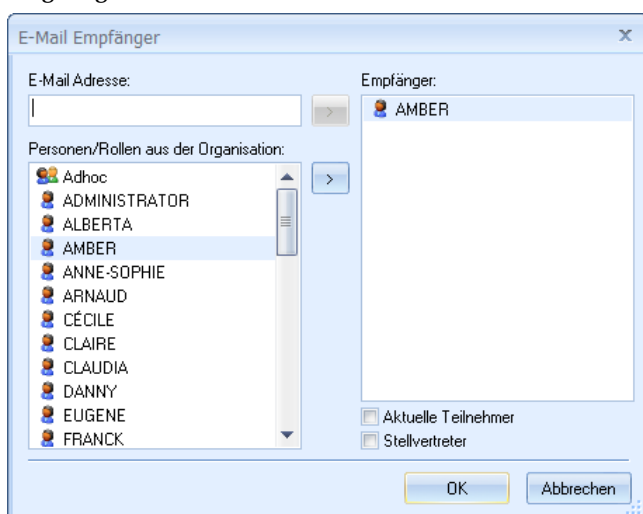
Für Multi-Instanz-Aktivitäten können zusätzlich Mahnfristen für den Zeitraum von Instanzanfang zu Instanzende einer jeweiligen Instanz festgelegt werden. Instanzanfang und Instanzende stehen aber nur den Zeitraum innerhalb der Instanz zur Verfügung.

Die Dauer kann in Tagen, Stunden, Minuten und Sekunden angegeben werden.

Für eine Mahnfrist wird ebenfalls eine Aktion festgelegt. Mögliche Aktionen sind das Senden von E-Mails und das Zuweisen an Benutzer oder Rollen. Eine Aktion fügen Sie über das Kontextmenü hinzu:



Über den Button **An** auf der Registerkarte **Email** wird eine Liste mit allen Rollen und Personen aus der Organisation geöffnet. Personen/Rollen können über den Pfeilbutton oder durch Doppelklick als Empfänger gewählt werden. Es können aber auch beliebige E-Mail Adressen angegeben werden. Über die Entfernen-Taste werden Empfänger wieder ausgetragen.



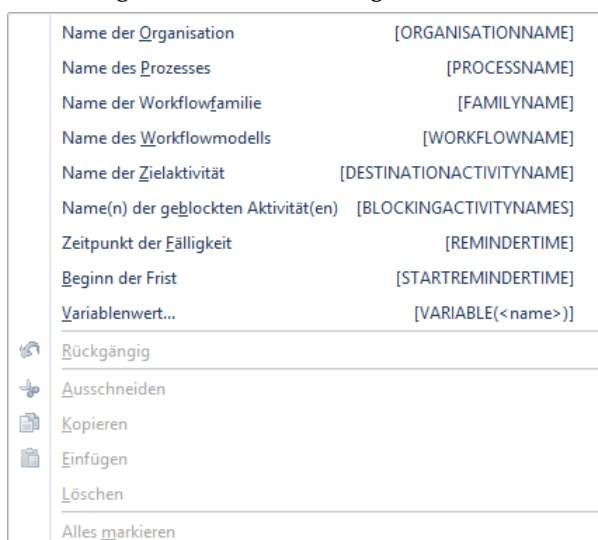
Weitere E-Mails mit anderem Betreff oder Text können über den E-Mail-Button eingerichtet werden.

Bereits eingerichtete werden über den Löschbutton gelöscht.



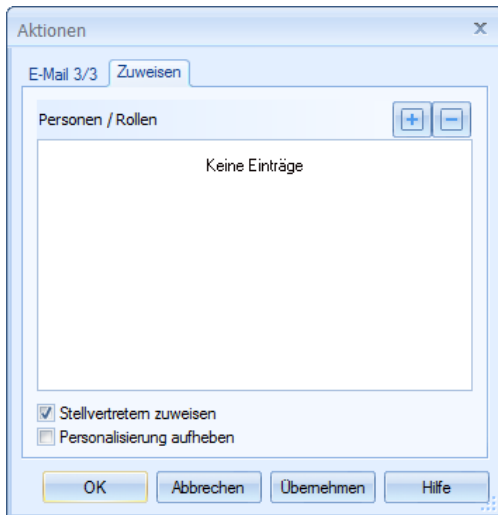
In das Betreff- und das Nachrichtenfeld können folgende Platzhalter eingetragen werden.

Den Dialog zur Auswahl der verfügbaren Platzhalter wird über das Kontextmenü im Textbereich geöffnet.



Platzhaltern können beliebige Zeichen voran- oder nachgestellt werden.

Auf der Registerkarte **Zuweisen** können Personen oder Rollen angegeben werden, in deren Eingangskorb die aktuelle Aktivität verschoben wird, falls die Mahnfrist überschritten wurde.



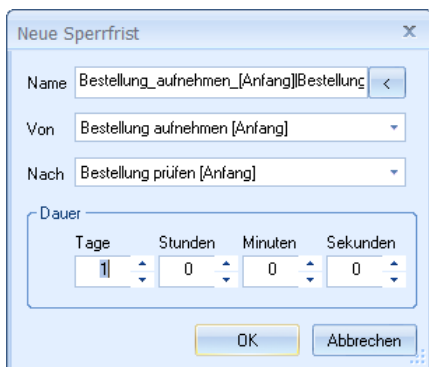
Wird dabei die Personalisierung nicht aufgehoben, wird die Aktivität nur in den Eingangskorb des Adressaten übertragen. Sie bleibt als personalisiert gekennzeichnet. Der Adressat kann die Personalisierung aufheben und die Aktivität selbst starten. Der Benutzer, der die Mahnfrist überschritten hat, kann die Aktivität abschließen, solange sie noch nicht durch den Adressaten personalisiert ist.

Wird die Personalisierung aufgehoben oder ist eine Aktivität nicht personalisiert, können sowohl der Adressat, wie auch alle anderen Benutzer, die als Teilnehmer der Aktivität eingetragen sind, die Aktivität ausführen.

Die Aktivität kann ebenfalls den Stellvertretern zugewiesen werden.

Sperrfristen

Eine neue Sperrfrist wird über den Eintrag **Neue Sperrfrist** aus dem Kontextmenü eingerichtet.



Für die Festlegung des Zeitraums kann von jeder Aktivität jeweils der Anfang gewählt werden. Die Dauer kann in Tagen, Stunden, Minuten und Sekunden angegeben werden.

Für eine Sperrfrist wird keine Aktion festgelegt. Will ein Benutzer eine Aktivität ausführen, bevor die Sperrfrist abgelaufen ist, erhält er im enaio® client einen entsprechenden Hinweis und kann die Aktivität nicht ausführen oder abschließen.

Zeichenblatt

Auf dem Zeichenblatt werden grafisch die Aktivitäten und die Transitionen zwischen den Aktivitäten festgelegt.

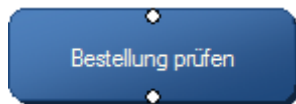
Bereits angelegt sind zwei Aktivitäten:



Die Startaktivität ist der Ausgangspunkt der Aktivitätsfolge. Sie hat keine Eigenschaften, die konfiguriert werden müssen.

Die Endaktivität ist der Endpunkt der Aktivitätsfolge. Konfiguriert werden für die Endaktivität Joins und die Variablenübernahme.

Drei Typen von Aktivitäten können Sie anlegen:



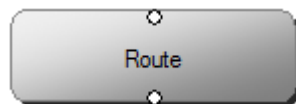
Benutzergesteuerte Aktivitäten

Für benutzergesteuerte Aktivitäten konfigurieren Sie eine Anwendung, Teilnehmer, Joins/Splits, Variablenübernahme und den Eingangskorb



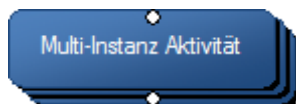
Schleifenaktivitäten

Für Schleifenaktivitäten konfigurieren Sie den Typ-'Repeat/Until' oder 'While', die entsprechende Bedingung, Joins/Splits und die Variablenübernahme. Schleifen haben keine Teilnehmer und keine Anwendung.



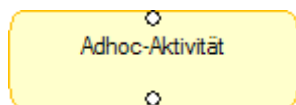
Routenaktivitäten

Routen sind Hilfsaktivität zum Zusammenfassen von Ablaufzweigen bei Joins/Splits und Schleifen. Für Routen konfigurieren Sie Joins/Splits und die Variablenübernahme.



Multi-Instanz-Aktivität

Benutzergesteuerte Aktivitäten werden von genau einem der Teilnehmer ausgeführt. Multi-Instanz-Aktivitäten werden parallel von allen zugeordneten Teilnehmern ausgeführt. Erst wenn alle Teilnehmer den Aktivitätsschritt weitergeleitet haben, geht der Ablauf weiter.



Adhoc-Aktivität

Adhoc-Aktivitäten sind Bereiche, in denen andere Aktivitäten angeordnet werden. Die Transitionen zwischen diesen Aktivitäten werden im enaio® client von den Benutzern über Lauflisten festgelegt. Lauflisten können einen Ablauf fest vorgeben oder so gestaltet werden, das die folgenden Benutzer den weiteren Ablauf selbst festlegen.

Die Aktivitäten verbinden Sie durch Transitionen. Transitionen werden als Pfeile zwischen dem Ausgang einer Aktivität und dem Eingang einer Aktivität dargestellt.

Richten Sie von einer Aktivität Transitionen zu mehreren anderen Aktivitäten ein, geben Sie für jede Transition die Bedingung an, unter der die Transition beschriftet wird.

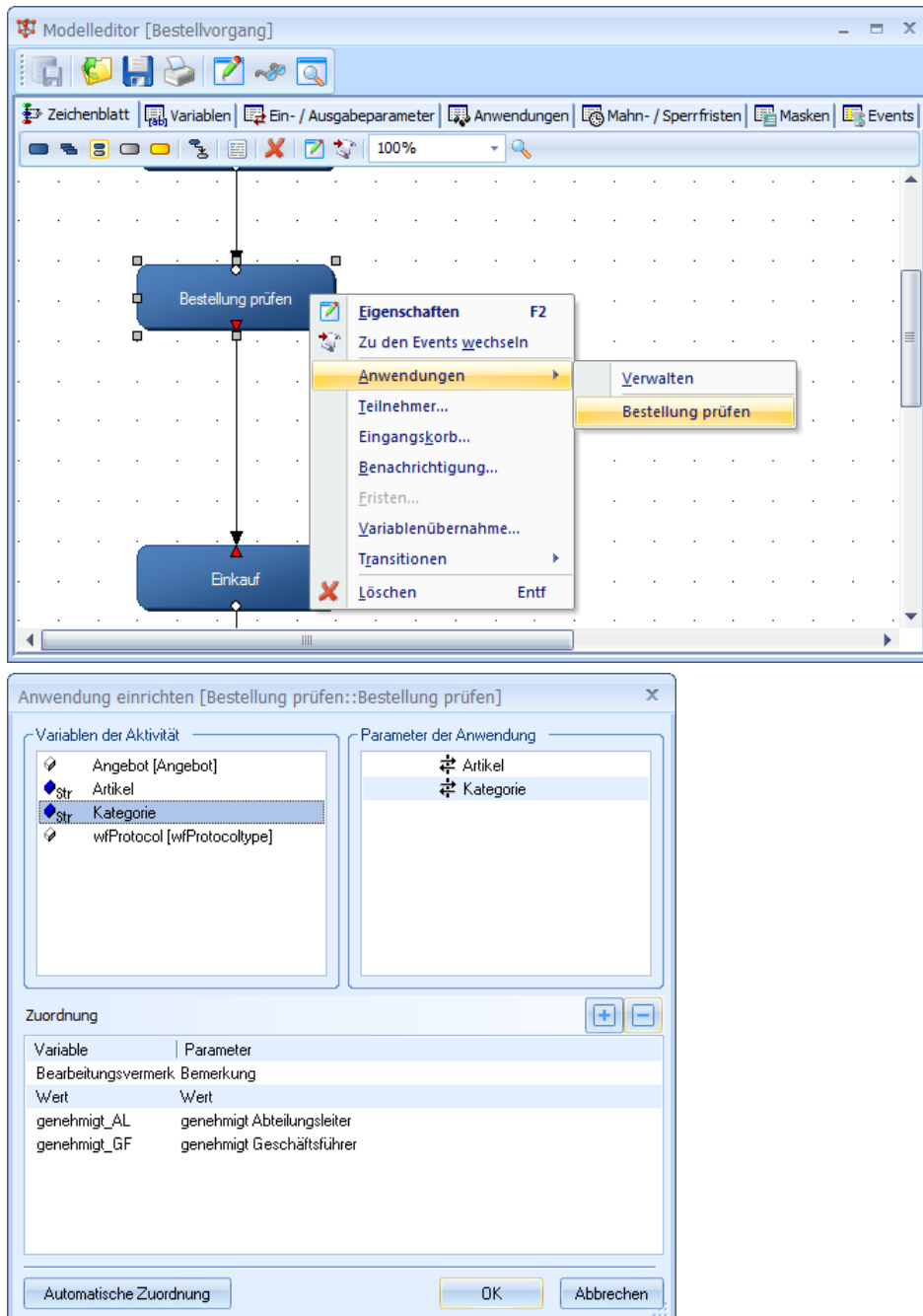
Schleifen haben zusätzlich einen Schleifenausgang und Schleifeneingang. Eine Transition vom Schleifenausgang zur ersten Aktivität innerhalb der Schleife hat die Eigenschaft 'From Loop', eine Transition von der letzten Aktivität innerhalb der Schleife zur Schleife hat die Eigenschaft 'To Loop'. Diese Transitions-Eigenschaften werden automatisch vergeben. Aktivitäten innerhalb einer Adhoc-Aktivität haben keine vorgegebenen Transitionen.

Aktivitäten und Anwendungen






Um Änderungen an Prozessvariablen vorzunehmen, müssen, wie in den Grundlagen erläutert, Anwendungen gestartet werden, wenn die Variablen nicht innerhalb der Engine manipuliert werden. Deshalb muss für jede benutzergesteuerte Aktivität festgelegt werden, welche Anwendung für die Bearbeitung vorgesehen ist. Die Anwendung ist im Standardfall mit einer Workflowmaske verbunden. Ebenfalls festgelegt wird, welchem Anwendungsparameter welche Variable zugeordnet ist.

Es ist möglich einer Aktivität mehrere Anwendungen zuzuordnen, wenn diese Anwendungen unterschiedliche Clienttypen, unterstützen. Anwendungen werden einer Aktivität über deren Kontextmenüpunkt **Anwendungen Verwalten** zugeordnet.

Für jede zugeordnete Anwendung werden nun die Anwendungsparameter jeweils einer Variablen zugeordnet. Es gibt unterschiedliche Arten von Anwendungsparametern. Eingangsparameter übernehmen Daten von der Workflow-Engine, übertragen diese Daten aber nicht an sie zurück. Dagegen werden Ausgangsparameter an die Workflow-Engine übertragen, jedoch nicht von dort übertragen. Ein-/Ausgangsparameter übertragen die Daten in beide Richtungen.



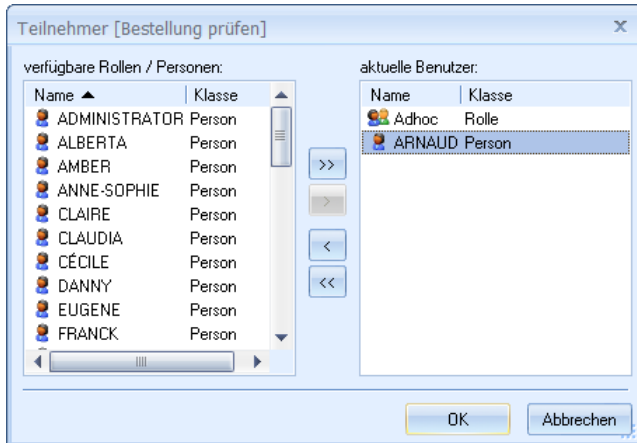
Sie wählen die gewünschte Anwendung und ordnen den Parametern der Anwendung die Variablen zu.
Die Parameter der Anwendung sind gekennzeichnet:

-  Maskenparameter
-  Parameter ohne Maskenfeld
-  Ein- / Ausgangsparameter
-  Ausgangsparameter
-  Eingangsparameter

Jedem definierten Parameter muss eine Variable zugeordnet werden. Wurde dies im Modell nicht festgelegt, so bricht der Prozess an dieser Aktivität mit einem Fehler ab.

Aktivitäten und Teilnehmer

Jeder benutzergesteuerten Aktivität ordnen Sie über das Kontextmenü Teilnehmer zu.



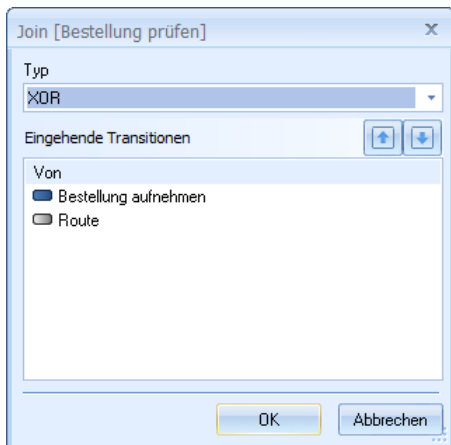
Als Teilnehmer können Rollen und Benutzer gewählt werden. Rollen und Benutzer werden über den Organisationsexplorer eingerichtet.

Zusätzlich stehen virtuelle Rollen zur Verfügung. Virtuelle Rollen sind Rollen, die Personen durch ihre Funktion im konkreten Ablauf eines Prozesses bekommen. Beispielsweise beinhaltet die virtuelle Rolle 'Prozessbeteiligte' alle konkreten Personen, die bereits am Prozess beteiligt waren.

Aktivitäten und Joins/Splits

In komplexeren Prozessmodellen ist es notwendig, das Verzweigen und das Zusammenführen von Bearbeitungssträngen gesondert zu behandeln. Dies wird durch so genannte Joins und Splits realisiert.

Joins und Splits können über das Kontextmenü einer Aktivitäten festgelegt werden. In unserem Beispiel wäre es an der Aktivität „Einkauf“ möglich, einen XOR-Join anzulegen, d.h. die Aktivität wird nur dann ausgeführt, wenn genau eine der eingehenden Transitionen eingetroffen ist. Auf Grund der bedingten Verzweigung ist dies auch so schon gegeben, aber an dem Beispiel können die Joins nochmals demonstriert werden.



Für die Aktivität 'Einkauf' werden alle eingehenden Transitionen im Dialog aufgelistet. Voreingestellt sind Transitionen immer vom Typ 'AND'. Der Typ kann einfach über eine Liste geändert werden.

Auf dem Zeichenblatt sind Ein- und Ausgangspunkte der Aktivitäten bei Transitionen des Typs 'XOR' rot gekennzeichnet.

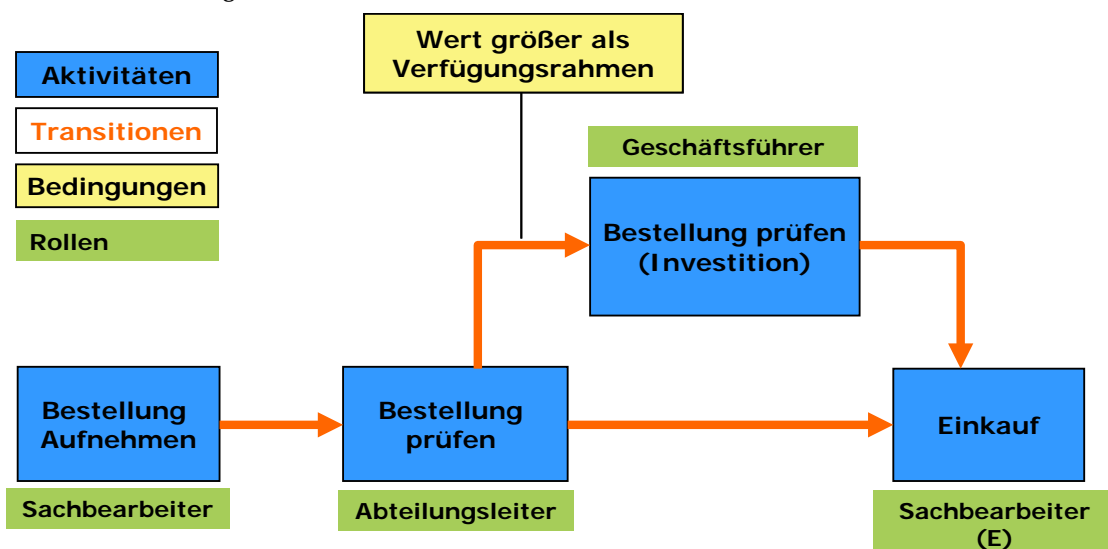
Aktivitäten und Variablenübernahme

Variablen können im Prozessverlauf mit mehreren Werten belegt werden. Sie haben innerhalb einer Aktivität den Charakter lokaler Variablen. Dieser Unterschied kommt dann zum Tragen, wenn mehrere Bearbeitungsschritte parallel ausgeführt werden. In diesem Falle besitzt jeder Bearbeitungsstrang eine lokale Kopie der Variablen, die in jedem Zweig unabhängig voneinander geändert werden können.

Gewöhnlich ist es hinreichend, genau eine Aktivität anzugeben, wodurch dann alle Variablenwerte von dieser Aktivität geerbt werden. Die Auswahl der Vorgängeraktivität wird im Dialog „Variablenübernahme“ vorgenommen. Jeder Aktivität, außer der Startaktivität, wird eine Standardaktivität für die Variablenübernahme zugeordnet.

Ist im Prozess vorgesehen, dass mehrere Bearbeitungsstränge wieder zusammengeführt werden, also etwa nach der Zusammenführung paralleler Bearbeitungsschritte, so muss definiert werden, von welcher Aktivität die Werte der Variablen übernommen werden sollen. Dabei ist es unerheblich, ob die Bearbeitungsstränge parallel oder exklusiv ausgeführt wurden. Wichtig ist, dass die Variablen mit den richtigen Werten der entsprechenden Vorgängeraktivität belegt werden.

Dies lässt sich am Beispiel des Bestellworkflows (siehe Grundlagenkapitel) gut verdeutlichen. Die Aktivität „Einkauf“ lässt sich über zwei unterschiedliche Wege erreichen. Einerseits kann die Genehmigung direkt vom Abteilungsleiter erteilt werden, wenn es sich jedoch um ein höherwertiges Investitionsgut handelt, dann muss noch die Zustimmung des Geschäftsführers eingeholt werden.



In diesem Falle können alternativ zwei Wege im Prozessgraphen beschriftet werden. An der Aktivität Einkauf hingegen müssen jedoch alle Variablen des Prozesses mit einem gültigen Wert belegt werden. Diese Werte können standardmäßig von der Aktivität „Bestellung prüfen“ vererbt werden, bis auf den Wert der Variablen „genehmigt (Geschäftsführer)“. Diese Variable kann durch die Aktivität „Bestellung prüfen“ nicht geändert worden sein, da der Abteilungsleiter dieses Feld nicht bearbeiten konnte. Aus diesem Grund muss im Modell hinterlegt werden, dass die Variable im Sonderfall von der Aktivität „genehmigt (Geschäftsführer)“ geerbt wird.

Die spezielle Zuordnung der Variablenübernahme erfolgt durch Markieren der gewünschten Variable und der entsprechenden Aktivität und nachfolgende Betätigung des Buttons **Hinzufügen**. Soll eine Zuordnung aufgehoben werden, so kann man dies über den Button **Entfernen** erreichen.

Je nach Prozessmodell kann es passieren, dass die Variablen von mehreren möglichen Aktivitäten übernommen werden können. In diesem Falle können auch mehrere Zuordnungen der Variablenübernahme definiert werden. Die Workflow-Engine geht dann so vor, dass die Liste der Variablenübernahme abgearbeitet wird und falls eine Variable belegt werden kann, da die entsprechende Aktivität durchlaufen wurde, dann erfolgt die Zuordnung und die restlichen Einträge werden für diese Variable ignoriert.

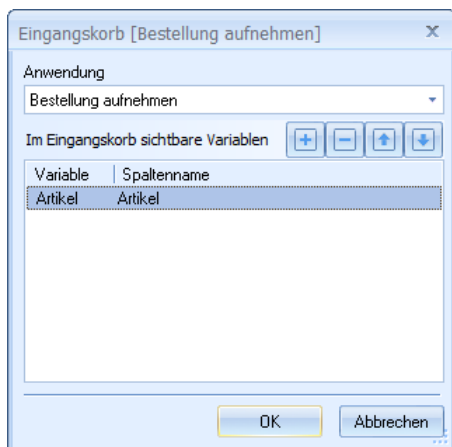
Existiert keine spezielle Zuordnung, so werden die Werte von der Standardaktivität übernommen. Wurde die Standardaktivität nicht ausgeführt und kann demnach keine Belegung der Variablen erfolgen, so werden die Variablen mit einem Fehlereintrag belegt. In diesem Falle ist das Ergebnis des weiteren Prozessverlaufes nicht definiert.

Für Schleifen richten Sie eine erweiterte Variablenübernahme ein (vgl. S.41)

Aktivitäten und der Eingangskorb

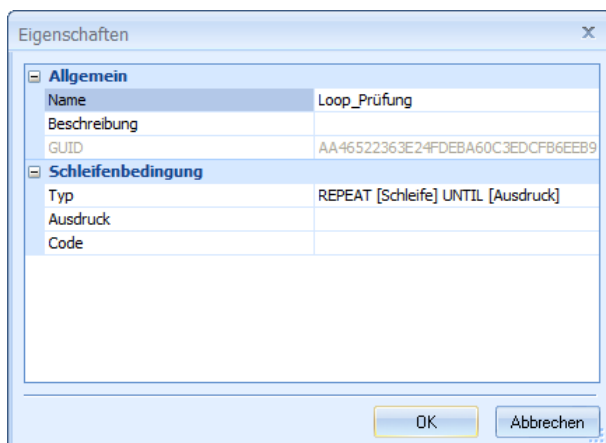
Die Teilnehmer finden Aktivitäten, die Sie ausführen können, in Ihrem Eingangskorb. Im Eingangskorb kann zu jeder Aktivität der aktuelle Wert von Variablen angezeigt werden. Das ermöglicht dem Benutzer, laufende Aktivitäten eines Modells besser zu unterscheiden.

Über das Kontextmenü einer Aktivität konfigurieren Sie die Anzeige von Variablen im Eingangskorb.



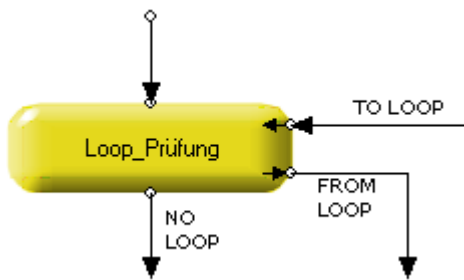
Schleifentyp, -bedingung und -transition

Für Schleifen geben Sie über den Eigenschaften-Dialog aus dem Kontextmenü den Schleifentyp und eine Bedingung an.



Ist die Schleife vom Typ 'Repeat/Until', wird die Transition zur folgenden Schleifenaktivität beschriftet, bis die Bedingung erfüllt ist. Die Transition zur folgenden Schleifenaktivität hat die Eigenschaft 'From Loop'. Ist die Bedingung erfüllt, wird die Transition mit der Eigenschaft 'No Loop' beschriftet.

Ist die Schleife vom Typ 'While', wird die Transition zur folgenden Schleifenaktivität beschriftet, solange die Bedingung erfüllt ist.



Die Bedingung tragen Sie im Feld **Ausdruck** ein.

Eigenschaften	
Allgemein	
Name	Loop Prüfung
Beschreibung	
GUID	AA46522363E24FDEBA60C3EDCFB6EEB9
Schleifenbedingung	
Typ	REPEAT [Schleife] UNTIL [Ausdruck]
Ausdruck	genehmigt_Geschaeftsfuehrer.value = 1
Code	

Kommentare sind im Bereich **Ausdruck** nicht möglich und führen zu Fehlern.

Über die Zeile **Code** kann Scriptcode angegeben werden, der vor der Prüfung des Ausdrucks ausgeführt wird. Damit besteht die Möglichkeit, die Werte der Variablen neu zu berechnen oder Entscheidungskriterien anhand von weiteren Prüfungen festzulegen.

Detaillierte Beispiele zu Schleifen finden Sie im Abschnitt 'Schleifen'.

Transitionsbedingungen

Richten Sie von einer Aktivität Transitionen zu mehreren anderen Aktivitäten ein, geben Sie für jede Transition die Bedingung an, unter der die Transition beschriftet wird. Die Transition, die beschriftet wird, muss eindeutig aus den Bedingungen folgen.

Eine Transitionsbedingung geben Sie über den Eigenschaften-Dialog einer Transition über die Zeile **Ausdruck** ein.

Eigenschaften	
Allgemein	
GUID	CC4853471D5D485BABE1628
Von	Bestellung aufnehmen
Nach	Route
Typ	NO LOOP
Ausdruck	Wert.value >= 400
Code	

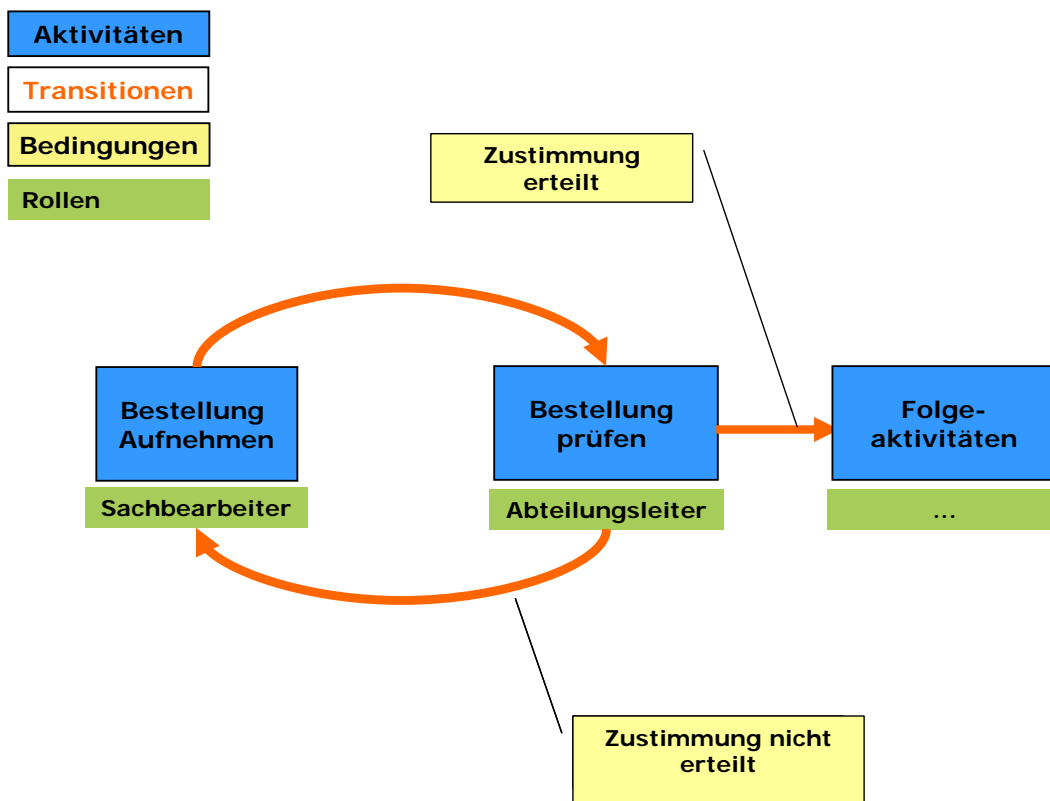
Über die Zeile **Code** kann Scriptcode angegeben werden, der vor der Prüfung des Ausdrucks ausgeführt wird.

Erweiterte Prozessmodellierung

Schleifen

Wie oben schon angedeutet, kann es notwendig sein, die Ausführung von einzelnen Aktivitäten oder ganzen Teilen des Prozessmodells solange zu wiederholen, bis eine bestimmte Bedingung eingetreten ist. Dies wird im Prozessmodell mit Schleifen abgebildet, die besondere Aktivitäten darstellen.

In unserem Beispiel könnte angenommen werden, dass vor der Freigabe durch den Abteilungsleiter möglicherweise Rückfragen an den Auftraggeber der Bestellung auftreten, die mit Hilfe des Workflowmodells abgebildet werden sollen. Da die Anzahl der Rückfragen nicht von vornherein bekannt ist, muss hier eine Schleife vorgesehen werden. Die Entscheidung, ob der Vorgang an den Einkauf bzw. Geschäftsführer weitergeleitet wird, soll davon abhängen, ob der Abteilungsleiter zugestimmt hat. Ist dies nicht der Fall, so soll der Vorgang an den Auftraggeber zurückgewiesen werden, wie in der Zeichnung verdeutlicht wird.



Beim Prozessdesign stehen nicht die Möglichkeiten einer höheren Programmiersprache zur Verfügung, in denen der Ablauf des Prozesses durch Anweisungen wie nachfolgend aufgeführt formuliert werden können.

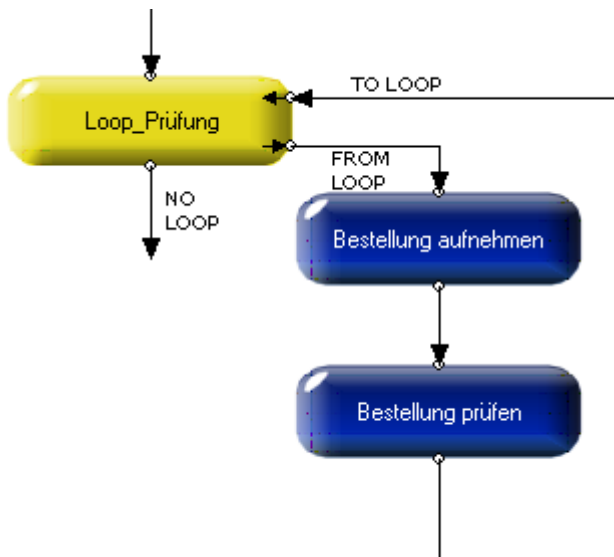
```
while <Bedingung>
{
  // Anweisung
}
```

```
repeat
{
  // Anweisungen
}
```

Deswegen greift man hier zu einer Hilfskonstruktion, nämlich zu speziellen Schleifenaktivitäten. Diese zeichnen sich dadurch aus, dass sie für außerhalb der Schleife liegende Aktivitäten wie eine Einheit auftreten und innerhalb der Schleife beliebig viele Aktivitäten beinhalten können. Die Schleifenaktivität ist also eine Art Container für die innerhalb der Schleife liegende Aktivitäten. Jede Schleifenaktivität besitzt eine Bedingung, anhand der entschieden werden kann, ob die Schleife durchlaufen bzw. wiederholt wird.

Nachfolgend wird der Mechanismus der Schleifenaktivitäten nochmals graphisch dargestellt. Von der Startaktivität (nicht dargestellt) wird eine Transition zum Eingang der Schleifenaktivität (Loop-Prüfung) hergestellt. Von der Schleifenaktivität wird die erste Aktivität "Bestellung aufnehmen" über den Schleifenausgang angesteuert und von dort

aus die Aktivität "Bestellung prüfen". Je nach Schleifentyp wird die Bedingung vor dem Eintritt in die Schleife oder nach dem Durchlauf ausgewertet, die darüber eine Aussage trifft, ob die Schleife wiederholt werden soll.



Schleifenaktivitäten sind entweder vom Typ **while** oder **repeat/until**. Über den Typ wird ausgedrückt, ob vor den Eintritt in die Schleife die Bedingung geprüft oder ob die Schleife mindestens einmal durchlaufen wird, bevor die Bedingung zur Auswertung kommt.

Bei while-Schleifen wird zuerst die Bedingung geprüft, bevor in die Schleife verzweigt wird. While-Schleifen können mit folgender Aussage assoziiert werden: Wiederhole die Schleife, solange die Bedingung wahr ist.

Bei repeat/until-Schleifen wird die Schleife mindestens einmal durchlaufen. Danach wird anhand der Bedingung geprüft, ob die Schleife wiederholt werden muss. Das ist genau dann der Fall, wenn die angegebene Bedingung noch nicht eingetreten ist. Man kann sich den Mechanismus einer repeat/until Schleife mit folgender Erklärung merken: Wiederhole die Schleife mindestens einmal, aber solange, bis die Bedingung wahr ist.

In unserem Beispiel wäre die Schleifenbedingung, dass der Abteilungsleiter noch nicht zugestimmt hat. Da es sich hier um eine Integer-Variable handelt, welche die Werte 0 (FALSE) oder 1 (TRUE) annimmt, müssen die Bedingungen entsprechend formuliert werden. Wollte man eine while-Schleife verwenden, so müsste die Bedingung folgendermaßen formuliert werden:

```
genehmigt_Abteilungsleiter.value = 0
```

Das bedeutet, dass die Schleife solange durchlaufen wird, wie der Wert der Variable genehmigt_Abteilungsleiter gleich 0 ist, dieser also noch nicht zugestimmt hat. Um in unserem Beispiel eine while-Schleife zu verwenden, muss der Wert natürlich mit 0 vorbelegt werden, da nach der Aufnahme der Bestellung der Abteilungsleiter noch keine Entscheidung getroffen hat.

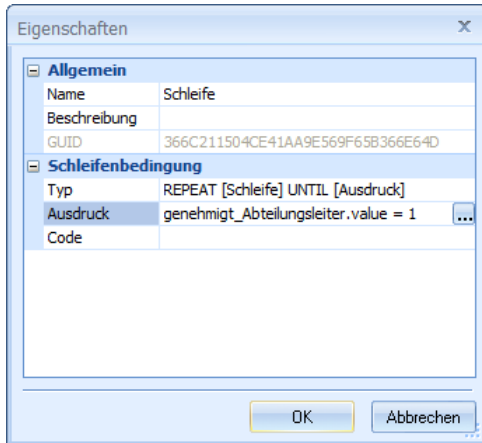
Sinnvoller wäre jedoch, hier eine repeat/until-Schleife vorzusehen, denn dadurch würde das Innere der Schleife mindestens einmal durchlaufen. Die zugehörige Bedingung lautet hier:

```
genehmigt_Abteilungsleiter.value = 1
```

Die Schleife soll demnach so lange durchlaufen werden, bis der Abteilungsleiter zugestimmt hat.

Die Bedingung kann über die Zeile **Ausdruck** des Eigenschaften-Dialogs der Schleifenaktivität angegeben werden. Da hier eine Schleife vom Typ "repeat/until" gewählt wurde, muss folgende Bedingung eingegeben werden:

```
genehmigt_Abteilungsleiter.value = 1
```

Damit die Schleife auch wirklich durchlaufen wird, müssen nun weitere Transitionen hinzugefügt werden, die die Schleifenaktivität mit den äußeren und inneren Aktivitäten verbindet. Je nachdem, welche Transitionen betroffen sind, ist der Typ der Transition anzugeben. Transitionen, welche die Schleifenaktivität mit den inneren Aktivitäten verbinden, sind gesondert gekennzeichnet. Zu diesem Zwecke stehen die Typen "From Loop" und "To Loop" zur Verfügung. In diesem speziellen Fall sind dies die Transitionen, die die Schleifenaktivität "Loop_Prüfung" mit der Aktivität "Bestellung aufnehmen" (From Loop) und die Aktivität "Bestellung prüfen" mit der Schleifenaktivität (To Loop) verbinden. Dies ist auch oben dargestellt.

Zusammenfassend müssen folgende Transitionen eingerichtet werden.

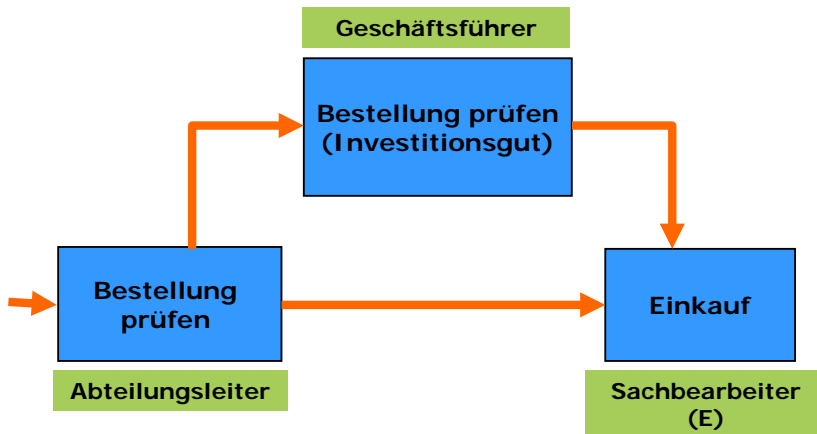
Von Aktivität	Zu Aktivität	Typ
StartActivity	Loop_Prüfung	no loop
Loop_Prüfung	Bestellung aufnehmen	from loop
Bestellung aufnehmen	Bestellung prüfen	no loop
Bestellung prüfen	Loop_Prüfung	to loop
Loop_Prüfung	Bestellung prüfen(Investitionsgut)	no loop
Loop_Prüfung	Einkauf	no loop
Bestellung prüfen(Investitionsgut)	Einkauf	no loop
Einkauf	EndActivity	no loop

Der Transitionstyp wird automatisch vergeben. Die Ein- und Ausgangspunkte für die inneren Aktivitäten sind durch Pfeile gekennzeichnet. Transitionen, die am Ausgangspfeil einer Schleife ansetzen, sind vom Typ 'From Loop', Transitionen, die zum Eingangspfeil einer Schleife führen, sind vom Typ 'To Loop'.

Variablenübernahme bei Schleifen

Die Variablenübernahme bei Aktivitäten ist notwendig, um für die nachfolgenden Aktivitäten entscheiden zu können, welche Werte übernommen werden sollen. Hat eine Aktivität genau einen Vorgänger, so ist es hinreichend, diese Aktivität als Standardaktivität anzugeben, von der die Variablen kopiert werden sollen. Kann hingegen eine Aktivität über mehrere Wege erreicht werden, so muss die Engine entscheiden, von welcher Vorgängeraktivität Variablen kopiert werden müssen. Dies kann nicht automatisch geschehen, da die Variablen auf den unterschiedlichen Zweigen unterschiedliche Werte annehmen können. Deshalb wird beim Prozessdesign eine Reihenfolge von Aktivitäten angegeben, von denen Variablen potentiell übernommen werden können.

An unserem Beispiel lässt sich dies gut für die Aktivität "Einkauf" verdeutlichen. Hier existieren zwei Wege, auf denen die Aktivität "Einkauf" erreicht werden kann. Prinzipiell können Variablen von jeder vorangegangenen Aktivität übernommen werden, also nicht zwangsläufig vom direkten Vorgänger. Meist sollen jedoch die Werte, die sich ja möglicherweise in den direkt vorangegangenen Aktivitäten verändert haben, in die nachfolgende Aktivität kopiert werden. Um für jeden Weg zu kennzeichnen, welche Variablen übernommen werden sollen, muss die Reihenfolge angegeben werden, in der Vorgängeraktivitäten gesucht werden.



Hier hat die Aktivität "Einkauf" die Vorgängeraktivitäten "Bestellung prüfen" und "Bestellung prüfen (Investition)". Deshalb wird als Standardaktivität "Bestellung prüfen" gewählt und eine explizite Variablenübernahme aus der Aktivität "Bestellung prüfen(Investitionsgut)" festgelegt. In diesem konkreten Falle spielt die Reihenfolge keine Rolle, sodass auch die Aktivität "Bestellung prüfen(Investitionsgut)" als Standardaktivität hätte eingerichtet werden können.

Die Variablenübernahme bei Schleifen gestaltet sich etwas komplexer als bei herkömmlichen Aktivitäten. Der Designer des Prozesses muss entscheiden, ob die Variablen bei jedem Durchlauf neu auf den Wert gesetzt werden, wie zu Beginn der Schleife oder ob sich die Variablen innerhalb der Schleife mit verändert und bei einem neuen Durchlauf erhalten bleiben sollen.

Wurden spezielle Zuordnungen für die Variablen vorgenommen, so werden diese in der angegebenen Reihenfolge ausgewertet. Wenn eine Vorgängeraktivität gefunden wurde, die der Zuordnung entspricht, so werden die Werte der angegebenen Variablen von dieser Aktivität übernommen. Ansonsten erfolgt die Übernahme von der Default-Aktivität.

In unserem Beispiel wird nun die Variablenübernahme der Schleifenaktivität "Loop_Prüfung" betrachtet.

Für die Schleifenaktivität "Loop_Prüfung" wird im ersten Schritt die Variablenübernahme für den Zeitpunkt festgelegt, zu dem die Schleife das erste Mal erreicht wird. Im Beispiel geschieht das durch die Transition 'StartActivity-Loop_Prüfung'. Eingetragen wird also 'StartActivity' als Standardaktivität. Eine weitere Zuordnung erfolgt hier nicht, da beim ersten Erreichen der Schleife Daten nur vom direkten Vorgänger übernommen werden.

Alle weiteren Schleifenvariablen werden als Eingangs- oder Ausgangsparameter eingerichtet. Den Dialog öffnen Sie über den Button **Schleifenvariablen**.

Ein- u. Ausgangsparameter der Schleife

Variablen

- Angebot [Angebot]
- Artikel
- Bearbeitungsvermerk
- genehmigt_Abteilungsleiter
- genehmigt_Geschäftsführer
- Kategorie
- Wert
- wfProtocol [wfProtocoltype]

Aktivitäten

- Bestellung aufnehmen
- Bestellung prüfen
- Bestellung prüfen (Investitionsgut)
- Einkauf
- EndActivity
- Loop_Prüfung
- Loop_Investition
- Route_Investition
- Route_Prüfung

Eingangsparameter

Variable	Aktivität
Angebot	Loop_Prüfung
Artikel	Loop_Prüfung
Bearbeitu...	Loop_Prüfung
genehmig...	Loop_Prüfung
genehmig...	Loop_Prüfung
Kategorie	Loop_Prüfung
Wert	Loop_Prüfung
wfProtocol	Loop_Prüfung

Ausgangsparameter

Variable	Aktivität
Angebot	Bestellung prüfen
Artikel	Bestellung prüfen
Bearbeitu...	Bestellung prüfen
genehmig...	Bestellung prüfen
genehmig...	Bestellung prüfen
Kategorie	Bestellung prüfen
Wert	Bestellung prüfen
wfProtocol	Bestellung prüfen

OK Abbrechen

Eingangsparameter sind die Variablen, die durch die in der Schleife enthaltenen Aktivitäten verändert werden und beim erneuten Schleifeneintritt wieder in der geänderten Form innerhalb der Schleife zur Verfügung stehen sollen. Die letzte Aktivität, die die Variablen bearbeitet, bevor sie erneut in die Schleife eintreten, ist die Schleife selbst, die prüft, ob die Bedingung für den Austritt erfüllt sind. Im Beispiel sind die Eingangsparameter also die Variablen, die innerhalb der Schleife in geänderter Form zur Aktivität 'Bestellung aufnehmen' weitergeleitet werden und aus der Aktivität 'Loop_Prüfung' stammen.

Ausgangsparameter sind die Variablen, die beim Austritt aus der Schleife an die Folgeaktivität weitergeleitet werden. Die Belegung der Variablen stammt aus der letzten Aktivität innerhalb der Schleife. Im Beispiel stammen die Ausgangsparameter aus der Aktivität 'Bestellung prüfen'.

Komplexere Schleifen

Im oben vorgestellten Beispiel wird der Vorgang von der Aktivität 'Bestellung prüfen' über eine Schleife an die Aktivität 'Bestellung aufnehmen' zurück verwiesen, solange der Abteilungsleiter Rückfragen hat und nicht zugestimmt hat. Praxisnäher wird der Prozess, wenn ebenfalls der Geschäftsführer Rückfragen an den Abteilungsleiter stellen kann, also der Vorgang von der Aktivität 'Bestellung prüfen (Investition)' an die Aktivität 'Bestellung prüfen' zurück verwiesen werden kann.

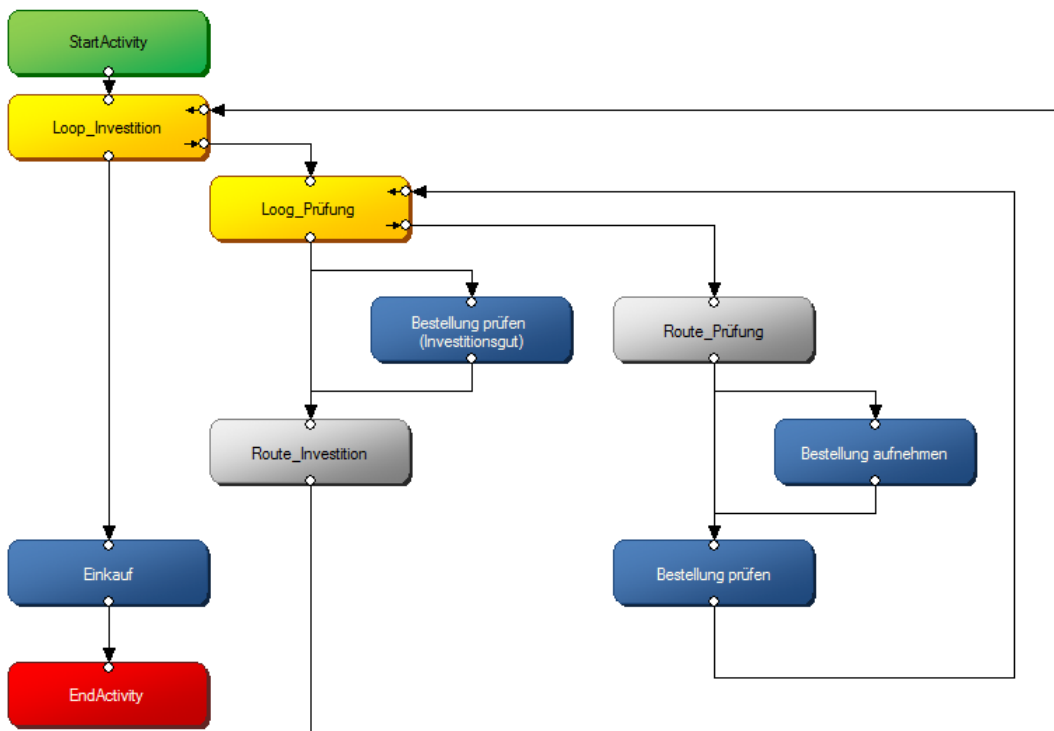
Für diese Anforderung gibt es zwei Lösungsmöglichkeiten. Erstens kann eine weitere Schleife verwendet werden, zweitens können innerhalb der einen Schleife weitere Verzweigungen verwendet werden.

Für beide Lösungsmöglichkeiten werden Aktivitäten vom Typ 'Route' verwendet. Diese Hilfsaktivitäten führen Verzweigungen zusammen und sind notwendig, da für Transitionen vom Typ 'To Loop' und 'From Loop' keine Bedingungen möglich sind. Die erste und die letzte Aktivität innerhalb einer Schleife muss immer durchlaufen werden.

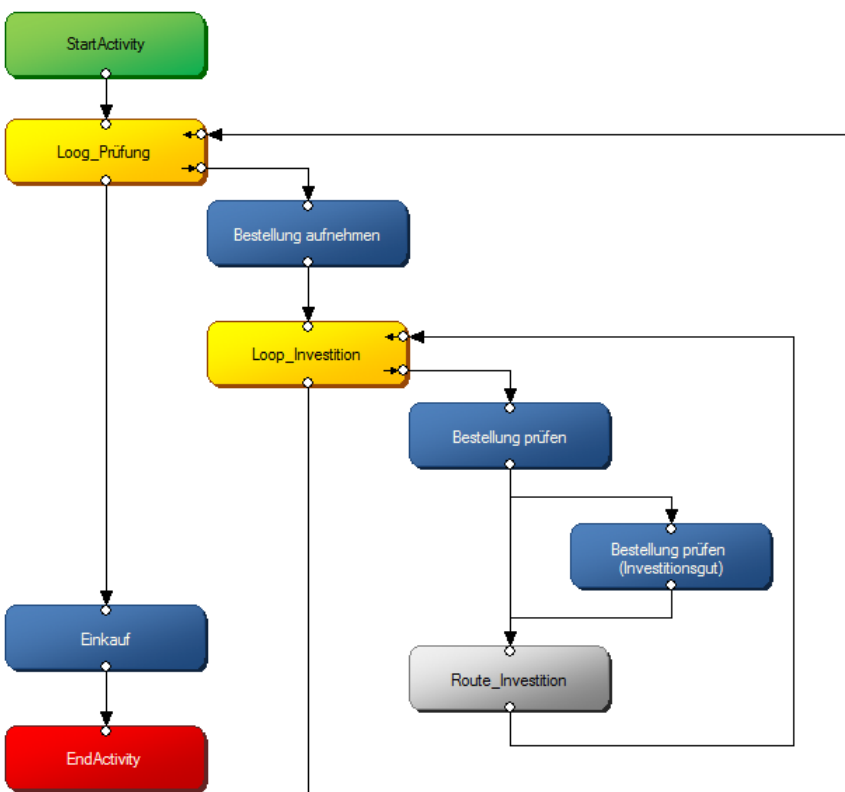
Verschachtelte Schleife

Eine Lösungsmöglichkeit ist, für eine weitere Rückfrage eine weitere Schleife zu verwenden. Für diese Lösung gibt es zwei Möglichkeiten.

Modell A:



Modell B:



Modell B ist etwas einfacher, da nur eine Hilfsaktivität vom Typ 'Route' notwendig ist und nur eine Verzweigung über Transitionsbedingungen. Dieses Modell wird im Folgenden detaillierter beschrieben.

Für beide Modelle ist ebenfalls ein Rückverweis von der Aktivität 'Bestellung prüfen (Investition)' zur Aktivität 'Bestellung aufnehmen' möglich, wenn der Geschäftsführer in der Aktivität 'Bestellung prüfen (Investition)' die Genehmigung des Abteilungsleiters aus der Aktivität 'Bestellung prüfen' zurücknehmen kann.

Aktivitäten

Die folgenden Aktivitäten werden benötigt:

Aktivität	Typ	Bedingungen
StartActivity	Prozedur	
EndActivity	Prozedur	
Bestellung aufnehmen	Anwendung	
Bestellung prüfen	Anwendung	
Bestellung prüfen (Investition)	Anwendung	
Einkauf	Anwendung	
Loop_Investition	Schleife (Repeat/Until)	Schleifenbedingung: genehmigt_Geschaeftsfuehrer.value=1 Or Wert.value<400 Or (Wert.value>=400 And genehmigt_Abteilungsleiter.value=0)
Loop_Prüfung	Schleife (Repeat/Until)	Schleifenbedingung: genehmigt_Abteilungsleiter.value=1
Route_Investition	Route	

Die Schleife 'Loop_Investition' ermöglicht eine Rückfrage zwischen dem Geschäftsführer und dem Abteilungsleiter. Die Schleife wird erst verlassen, wenn eine der folgenden Bedingungen erfüllt ist:

§ genehmigt_Geschaeftsfuehrer.value=1

Der Geschäftsführer hat seine Genehmigung erteilt.

§ Wert.value<400

Der Wert ist kleiner als 400€. Eine Genehmigung des Geschäftsführers ist nicht notwendig.

§ (Wert.value>=400 And genehmigt_Abteilungsleiter.value=0)

Der Wert ist größer oder gleich 400€, aber die Genehmigung des Abteilungsleiters liegt noch nicht vor.

Da alle Vorgänge die Schleife durchlaufen, müssen auch die Vorgänge die Schleife zur Aktivität 'Loop_Prüfung' wieder verlassen, die nicht oder noch nicht den Geschäftsführer betreffen.

Die Schleife 'Loop_Prüfung' ermöglicht eine Rückfrage zwischen dem Abteilungsleiter und dem Sachbearbeiter. Die Schleife wird erst zur Aktivität 'Einkauf' verlassen, wenn die folgende Bedingung erfüllt ist:

§ genehmigt_Abteilungsleiter.value=1

Der Abteilungsleiter hat seine Genehmigung erteilt.

Diese Schleifenbedingung reicht aus, da Vorgänge, die eine Genehmigung des Abteilungsleiters haben und zusätzlich eine Genehmigung des Geschäftsführers benötigen, die Schleife erst erreichen, wenn beide Genehmigungen vorliegen. Hat der Geschäftsführer noch nicht genehmigt, bleibt der Vorgang in der Schleife 'Loop_Investition'.

Die Routen-Aktivität 'Route_Investition' ist notwendig, da die Daten für die Schleife 'Loop_Investition' aus den Aktivitäten 'Bestellung prüfen' oder 'Bestellung prüfen (Investition)' stammen können, aber nur eine Transition vom Typ 'to loop' zur Schleife 'Loop_Investition' erlaubt ist.

Transitionen

Folgende Transitionen werden benötigt:

von	nach	Typ
-----	------	-----

StartActivity	Loop_Prüfung	no loop
Loop_Prüfung	Bestellung aufnehmen	from loop
Bestellung aufnehmen	Loop_Investition	no loop
Loop_Investition	Bestellung prüfen	from loop
Bestellung prüfen	Bestellung prüfen (Investition)	no loop Transitionsbedingung: Wert.value >= 400 And genehmigt_Abteilungsleiter.value = 1
Bestellung prüfen	Route_Investition	no loop Transitionsbedingung: Wert.value < 400 Or (Wert.value >= 400 And genehmigt_Abteilungsleiter.value = 0)
Bestellung prüfen (Investition)	Route_Investition	no loop
Route_Investition	Loop_Investition	to loop
Loop_Investition	Loop_Prüfung	to loop
Loop_Prüfung	Einkauf	no loop
Einkauf	EndActivity	no loop

Ausgehend von der Aktivität 'Bestellung prüfen' gibt es zwei Transitionen:

- § Die Transition zur Aktivität 'Bestellung prüfen (Investition)' wird beschritten, wenn der Wert größer oder gleich 400€ ist und der Abteilungsleiter die Genehmigung erteilt hat.
- § Die Transition zur Aktivität 'Route_Investition' wird beschritten, wenn der Wert kleiner als 400€ ist oder der Wert größer/gleich 400€ ist, aber der Abteilungsleiter die Genehmigung noch nicht erteilt hat.

Über die formulierten Transitionsbedingungen muss eine eindeutige Entscheidung, welche Transition gewählt wird, möglich sein.

Variablenübernahme

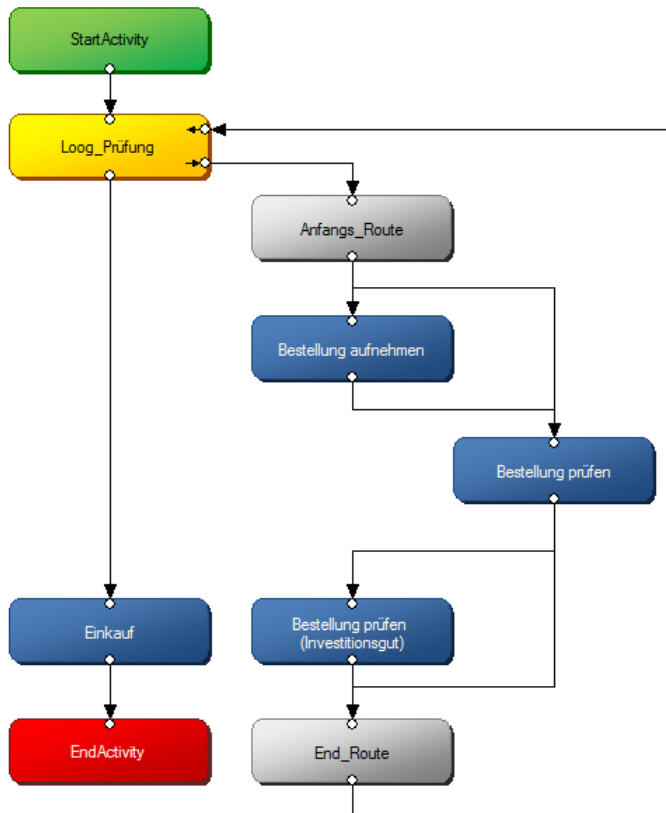
Folgende Variablenübernahmen werden benötigt:

Aktivität	Default-Aktivität	Zuordnung
StartActivity	-	
EndActivity	Einkauf	
Bestellung aufnehmen	Loop_Prüfung	
Bestellung prüfen	Loop_Investition	
Bestellung prüfen (Investition)	Bestellung prüfen	
Einkauf	Loop_Prüfung	
Loop_Investition	Bestellung aufnehmen	Schleifenvariablen Eingangsparameter: Loop_Investition Ausgangsparameter: Route_Investition
Loop_Prüfung	StartActivity	Schleifenvariablen Eingangsparameter: Loop_Prüfung Ausgangsparameter: Loop_Investition

Route_Investition	Bestellung prüfen	Bestellung prüfen (Investition)
-------------------	-------------------	---------------------------------

Verzweigung innerhalb einer Schleife

Statt über zwei verschachtelte Schleifen kann die Anforderung einer Rückfrage zwischen Abteilungsleiter und Sachbearbeiter und einer Rückfrage zwischen Geschäftsführer und Abteilungsleiter ebenfalls durch Verzweigungen über Transitionsbedingungen innerhalb einer Schleife erfüllt werden.



Möglich ist ebenfalls ein Rückverweis von der Aktivität 'Bestellung prüfen (Investition)' zur Aktivität 'Bestellung aufnehmen', wenn der Geschäftsführer in der Aktivität 'Bestellung prüfen (Investition)' die Genehmigung des Abteilungsleiters aus der Aktivität 'Bestellung prüfen' zurücknehmen kann.

Aktivitäten

Die folgenden Aktivitäten werden benötigt:

Aktivität	Typ	
StartActivity	Prozedur	
EndActivity	Prozedur	
Bestellung aufnehmen	Anwendung	
Bestellung prüfen	Anwendung	
Bestellung prüfen (Investition)	Anwendung	
Einkauf	Anwendung	
Loop	Schleife (Repeat/Until)	Schleifenbedingung: genehmigt_Geschaeftsfuehrer.value=1 Or (genehmigt_Abteilungsleiter.value=1 And Wert.value<400)
Anfangs_Route	Route	
End_Route	Route	

Die Schleife 'Loop' ermöglicht eine Rückfrage zwischen dem Geschäftsführer und dem Abteilungsleiter sowie dem Abteilungsleiter und dem Sachbearbeiter. Die Schleife wird erst zur Aktivität 'Einkauf' verlassen, wenn eine der folgenden Bedingungen erfüllt ist:

§ `genehmigt_Geschaeftsfuehrer.value=1`

Der Geschäftsführer hat seine Genehmigung erteilt.

§ `(genehmigt_Abteilungsleiter.value=1 And Wert.value<400)`

Der Abteilungsleiter hat seine Genehmigung erteilt und der Wert ist kleiner als 400€. Eine Genehmigung des Geschäftsführers ist in diesem Fall nicht notwendig.

Die Routen-Aktivität 'Anfangs_Route' ist notwendig, da aus der Schleife nur eine Aktivität vom Typ 'From Loop' erlaubt ist, aus der Schleife also nicht über Transitionsbedingungen zur Aktivität 'Bestellung aufnehmen' oder zur Aktivität 'Bestellung prüfen' weitergeleitet werden kann.

Die Routen-Aktivität 'End_Route' ist notwendig, da die Daten für die Schleife aus den Aktivitäten 'Bestellung prüfen' oder 'Bestellung prüfen (Investition)' stammen können, aber nur eine Transition vom Typ 'To Loop' zur Schleife erlaubt ist.

Transitionen

Folgende Transitionen werden benötigt:

von	nach	Typ
StartActivity	Loop	no loop
Loop	Anfangs_Route	from loop
Anfangs_Route	Bestellung aufnehmen	no loop Transitionsbedingung: <code>genehmigt_Abteilungsleiter.value=0</code>
Anfangs_Route	Bestellung prüfen	no loop Transitionsbedingung: <code>genehmigt_Abteilungsleiter.value=1</code>
Bestellung aufnehmen	Bestellung prüfen	no loop

Bestellung prüfen	Bestellung prüfen (Investition)	no loop Transitionsbedingung: Wert.value>=400 And genehmigt_Abteilungsleiter.value=1
Bestellung prüfen	End_Route	no loop Transitionsbedingung: Wert.value<400 Or genehmigt_Abteilungsleiter.value=0
Bestellung prüfen (Investition)	End_Route	no loop
End_Route	Loop	to loop
Loop	Einkauf	no loop
Einkauf	EndActivity	no loop

Ausgehend von der Aktivität 'Anfangs_Route' gibt es zwei Transitionen:

- § Die Transition zur Aktivität 'Bestellung aufnehmen' wird beschriftet, wenn der Abteilungsleiter die Genehmigung noch nicht erteilt hat.
- § Die Transition zur Aktivität 'Bestellung prüfen' wird beschriftet, wenn der Abteilungsleiter die Genehmigung bereits erteilt hat.

Ausgehend von der Aktivität 'Bestellung prüfen' gibt es ebenfalls zwei Transitionen:

- § Die Transition zur Aktivität 'Bestellung prüfen (Investition)' wird beschriftet, wenn der Wert größer oder gleich 400€ ist und der Abteilungsleiter die Genehmigung erteilt hat.
- § Die Transition zur Aktivität 'End_Route' wird beschriftet, wenn der Wert kleiner als 400€ ist oder der Abteilungsleiter die Genehmigung noch nicht erteilt hat.

Variablenübernahme

Folgende Variablenübernahmen werden benötigt:

Aktivität	Default-Aktivität	Zuordnung
StartActivity	-	
EndActivity	Einkauf	
Bestellung aufnehmen	Anfangs_Route	
Bestellung prüfen	Anfangs_Route	Bestellung aufnehmen
Bestellung prüfen (Investition)	Bestellung prüfen	
Einkauf	Loop	
Loop	Bestellung aufnehmen	Schleifenvariablen Eingangsparameter: Loop Ausgangsparameter: End_Route
Anfangs_Route	Loop	
End_Route	Bestellung prüfen	Bestellung prüfen (Investition)

Multi-Instanz-Aktivitäten

Innerhalb von Abstimmungs- oder Kenntnisnahmeprozessen kann es sinnvoll sein, eine Aktivität zugleich mehreren Teilnehmern vorzulegen. Multi-Instanz-Aktivitäten werden in einem Ablauf allen Teilnehmern im Eingangskorb zur Verfügung gestellt und können von allen Teilnehmern parallel ausgeführt werden. Erst wenn alle Teilnehmer den Aktivitätsschritt weitergeleitet haben, wird die Transition zur nächsten Aktivität beschriftet.

Bei Multi-Instanz-Aktivitäten ist es in der Regel notwendig, die Daten der einzelnen Instanzen durch ein Event zusammenzuführen. Innerhalb einer Instanz werden lokale Variablen verwendet, die Werte der lokalen Variablen aller Instanzen werden durch ein Skript ermittelt, ausgewertet und an die Folgeaktivität weitergeleitet.

Beispiel:

Alle Teilnehmer einer Aktivität geben über Optionsschaltflächen eine positive oder negative Bewertung ab.

Die Auswertung erfolgt über ein EndActivity-Event. Gibt es mehr positive Bewertungen als negative, erhält die Variable 'ergebnis' durch das folgende Skript den Wert '1'.

```
lCount = thisprocess.CurrentActivity.MultiInstances.Count
lpos = 0
lneg = 0
for i = 0 to lCount -1
if CInt ( thisprocess.CurrentActivity.MultiInstances.Item(i).
    GetDataFieldByName("bewertung").Value) = 0 then
    lpos = lpos +1
else
    lneg = lneg +1
end if
next
if lpos > lneg then
    ergebnis.Value = 0
else
    ergebnis.Value = 1
end if
```

Dieses Ergebnis kann beispielsweise bei der Folgeaktivität über die Variable 'ergebnis' angezeigt werden.

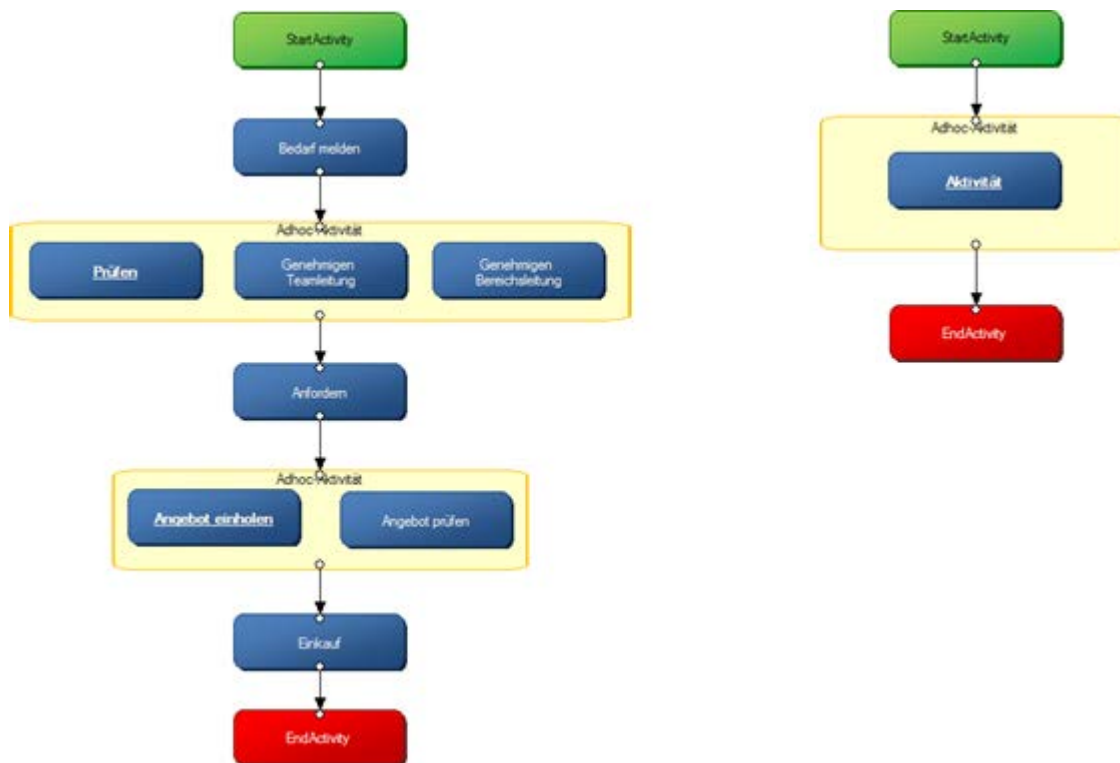
Einer Multi-Instanz-Aktivität, die bereits ausgeführt wird, kann per Skript eine weitere Instanz hinzugefügt werden.

Adhoc-Aktivitäten

Arbeitsprozesse können so komplex sein, dass eine genaue Abbildung in einem Workflowmodell sowie Pflege und Anpassungen solcher Modelle sehr aufwendig und umständlich ist. Adhoc-Aktivitäten erlauben es, innerhalb von Modellen Bereiche festzulegen, in denen die Abläufe von den Benutzern der jeweiligen Situation angepasst werden dürfen. Dadurch können Modelle einfacher entwickelt und gepflegt werden.

Adhoc-Aktivitäten können an beliebiger Position in einem Modell und beliebig oft eingefügt werden. Ein Modell kann aber auch aus nur einer Adhoc-Aktivität bestehen. Innerhalb der Adhoc-Aktivität wird mindestens eine Aktivität angeordnet, allerdings keine Schleife und weitere Adhoc-Aktivität.

Beispiele:



Eine Aktivität innerhalb der Adhoc-Aktivität wird als Standardaktivität gekennzeichnet. Diese Aktivität ist die erste, die gestartet wird und von der ausgehend der Benutzer die weitere Folge der Aktivitäten in einer Laufliste konfiguriert. Dabei kann der Benutzer alle Aktivitäten aus dem Adhoc-Bereich auswählen und auch mehrmals in der Laufliste anordnen. Zusätzlich kann der Benutzer festlegen, ob die gesamte Laufliste oder einzelne Teile der Laufliste im Ablauf von den folgenden Benutzern geändert werden kann.

Da der Ablauf der Aktivitäten im Adhoc-Bereich nicht vorgegeben ist, werden die Daten über globale Variablen weitergeleitet, andernfalls müssen die Daten über Events organisiert werden.

Events

enaio® client und enaio® webclient unterscheiden sich in Bezug auf den Einsatz von Clientskripten. Die folgenden Informationen beziehen sich in den konkreten Beispielen und Bezügen auf den enaio® client.

Der Aufruf von Workflowjobs aus serverseitigen Workflowevents über den 'running context' ist nicht möglich. Hierbei kommt es zum Absturz des enaio® Servers.

Bei der Bearbeitung von Aktivitäten ist es oft wünschenswert, wenn einige Arbeitsschritte automatisch ablaufen. Teilweise kann dies durch das Modell abgebildet werden, mitunter ist es jedoch notwendig, dass Aktionen aus dem speziellen Kontext heraus durchgeführt werden. Zu diesem Zwecke kann am Client und in der Workflow-Engine bei bestimmten Ereignissen Scriptcode gestartet werden, mit Hilfe dessen diese Routinen automatisch ausgeführt werden. Die Schnittstelle gleicht den aus dem DMS bekannten Events. Häufige Anwendungsfälle im Kontext des Workflow sind:

- § Initialisierung von Datenfeldern auf den Masken
- § Plausibilitätsprüfung von abhängigen Feldern
- § Vollständigkeitsprüfung bei der Dateneingabe
- § Automatische Bearbeitung der Workflowakte durch Erzeugen, Hinzufügen oder Löschen von Dokumenten
- § Übertragung von DMS-Objekten (Ordner, Register, Dokumente) zum oder vom DMS
- § Prüfung von Berechtigungen für das Setzen von Werten in der Maske
- § Prüfungen vor dem Starten einer Aktivität in der Workflow-Engine
- § Auswertung der Organisationsstruktur
- § Skriptgesteuerte Festlegung der Bearbeiter von Aktivitäten
- § Skriptgesteuertes Anlegen, Bearbeiten und Löschen von Fristen

Es existieren mehrere Einsprungspunkte für die Ausführung von Eventcode. Der Code kann mit VBScript geschrieben werden. Dadurch bieten sich weit gefächerte Anwendungsmöglichkeiten. Standardmäßig können folgende Objekte aus dem Scriptcode angesprochen werden:

- § alle Objekte des Windows-Script-Host (File-System-Objekt, System-Objekt)
- § Datenbankobjekte über Active Data Objects (ADO)

In Clientskripten stehen zusätzlich noch folgende COM-Schnittstellen zur Verfügung:

- § die COM-Schnittstelle des enaio® clients (s. Dokumentation COM-Schnittstelle)
- § die oxactive.dll-Schnittstelle zum Zugriff auf Übergabedateien (s. Dokumentation Event-Editor)
- § Erweiterungen der COM-Schnittstelle zum Zugriff auf den laufenden Prozess und die aktuelle Aktivität
- § Methoden der Funktionsbibliothek oxvbbas.dll, die im Clientverzeichnis registriert ist

In der Workflow-Engine, also für Serverskripte existiert eine Schnittstelle zum Zugriff auf die Organisationsstruktur, den laufenden Prozess, dessen Variablen und Fristen und die Workflowakte. Dabei werden einige Objektvariablen bereits initialisiert und dem Skript übergeben.

Eine genaue Aufstellung der zur Verfügung stehenden Objekte ist der Workflow-Programmierreferenz im Anhang zu entnehmen.

Weitere Objekte können eingebunden werden, wenn sie über skriptfähige COM-Schnittstellen verfügen. In Clientskripten werden häufig Office-Anwendungen angesprochen, um in diesen Dokumente zu erzeugen oder Daten abzufragen. Sollen gesonderte Aktionen ausgeführt werden, die mit Scriptcode nicht abgebildet werden können, so besteht die Möglichkeit, in einer höheren Programmiersprache (z.B. VB, C++, Delphi) ActiveX-DLLs zu schaffen, die dann über die Skripts eingebunden werden. Zu beachten ist hier, dass die Schnittstelle skriptfähig sein muss. Dass ist beim Design der Schnittstellen insbesondere bei der Übergabe von Strings zu beachten.

Objekte zur Einbindung in den Servercode unterliegen gesonderten Richtlinien und sollen nur durch OPTIMAL SYSTEMS erstellt werden.

Für Aktivitäten stehen folgende Events zur Verfügung:

Event	Ausführung	Einsprungspunkt	Anwendungsbeispiele
ButtonClick	Client	Bei Betätigung einer Schaltfläche auf einer Workflow-Maske	Erzeugung eines Dokumentes per Scriptcode
ValueChanged	Client	Wenn die Eingabe in einem Textfeld abgeschlossen wurde	Plausibilitätschecks
FocusGained	Client	Wenn der Fokus auf ein Textfeld gelegt wird.	Vorbelegung
StartActivity	Server	Nach Erstellung einer Aktivität aber vor der Bearbeitung durch eine Anwendung	Initialisierung von Variablen am Server
BeforeOpen	Client	Bevor die Aktivitätenmaske dargestellt wird	Initialisierung von Variablen, Vorbelegung von Feldern, Automatische Übernahme von Dokumenten aus dem DMS / in das DMS
EndActivity	Server	Nach Bearbeitung einer Aktivität durch eine Anwendung, aber vor der eigentlichen Beendigung	Dynamische Änderung von Variablen aufgrund von Workflowvariablen und am Server bekannten Daten
BeforeForward	Client	Nachdem der Button Weiterleiten gedrückt wurde, aber bevor die Aktivität beendet wurde. Aktion kann durch Returncode abgebrochen werden. S.u.	Plausibilitätschecks, Ablage von Dokumenten im DMS
BeforeCancel	Client	Nachdem der Button Schließen gedrückt wurde.	
ButtonClick	Client	Nachdem der Benutzer auf ein Bedienelement geklickt hat.	Überprüfen von Maskenaktionen
EnterPage	Client	Nachdem der Benutzer eine Registerkarte aktiviert hat.	Überprüfen von Maskenaktionen

Event	Ausführung	Einsprungspunkt	Anwendungsbeispiele
LeavePage	Client	Nachdem der Benutzer eine Registerkarte verlassen hat.	Überprüfen von Maskenaktionen
ValueChanged	Client	Nachdem der Benutzer einen Wert in einem Eingabefeld geändert hat.	Überprüfen von Maskenaktionen
FocusGained	Client	Nachdem der Benutzer den Fokus auf ein Bedienelement gesetzt hat.	Überprüfen von Maskenaktionen
BeforeAddRow	Client	Bevor der Benutzer eine Tabellenzeile hinzugefügt hat.	Überprüfen von Maskenaktionen
BeforeDeleteRow	Client	Bevor der Benutzer eine Tabellenzeile gelöscht hat.	Überprüfen von Maskenaktionen
CellValueChanged	Client	Nachdem der Benutzer einen Wert in einer Tabellenzeile geändert hat.	Überprüfen von Maskenaktionen
CellFocusGained	Client	Nachdem der Benutzer den Fokus auf eine Tabellenzeile gesetzt hat.	Überprüfen von Maskenaktionen
SimulateMaskEdit	Simulationstool	Wird von einem Simulationstool zur Simulation der Anwendungsbearbeitung genutzt.	Wird für Produkktivsysteme nicht benötigt.
PersonalizeWorkItem	Server	Nachdem ein Benutzer einen nicht personalisierten Vorgangsschritt gestartet hat und bevor der Schritt personalisiert wird.	
GetWorkItemParams	Server	Nachdem ein Benutzer einen personalisierten Vorgangsschritt erneut gestartet hat und die Variablen eingelesen wurden.	
StartInstance	Server	Nachdem ein Benutzer eine Multi-Instanz-Aktivität gestartet hat.	Variablenverwaltung für Multi-Instanz-Aktivitäten
EndInstance	Server	Nachdem ein Benutzer eine Multi-Instanz-Aktivität weitergeleitet hat.	Variablenverwaltung für Multi-Instanz-Aktivitäten
TimerFired	Server	Nachdem die Mahnfrist der verbundenen Aktivität abgelaufen ist	

Die Events werden, wenn sie implementiert sind, in folgender Reihenfolge ausgeführt.

Event	Ort
StartActivity	Server
PersonalizeWorkItem	Server
GetWorkItemParams	Server
BeforeOpen	Client
ButtonClick ValueChanged FocusGained EnterPage LeavePage	Client
BeforeForward	Client
EndActivity	Server

BeforeForward-Event

Über das BeforeForward-Event kann die Bearbeitung einer Aktivität gesteuert werden. So kann je nach Ausgang einer Plausibilitätsprüfung festgelegt werden, ob die Aktivität weitergeleitet oder der aktuelle Vorgang abgebrochen werden

soll. Mit Hilfe des Returncodes kann die Art und Weise der weiteren Bearbeitung festgelegt werden. Dabei stehen folgende mögliche Werte zur Verfügung.

Returncode	Bedeutung
0	Vorgangsschritt kann weitergeleitet werden
-1	Abbruch des Weiterleitens, Maske bleibt geöffnet, Daten werden nicht gespeichert
-2	Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden gespeichert
-3	Abbruch des Weiterleitens, Maske wird geschlossen, Daten werden nicht gespeichert

BeforeCancel-Event

Das BeforeCancel-Event erhält als Variablenwerte immer die aktuellen Werte aus der Maske. Wenn der Benutzer beim Abbruch das Speichern der aktuellen Maskenwerte gewählt hat, dann hat 'SaveWorkItem' den Wert 'true'.

BeforeOpen und ButtonClick

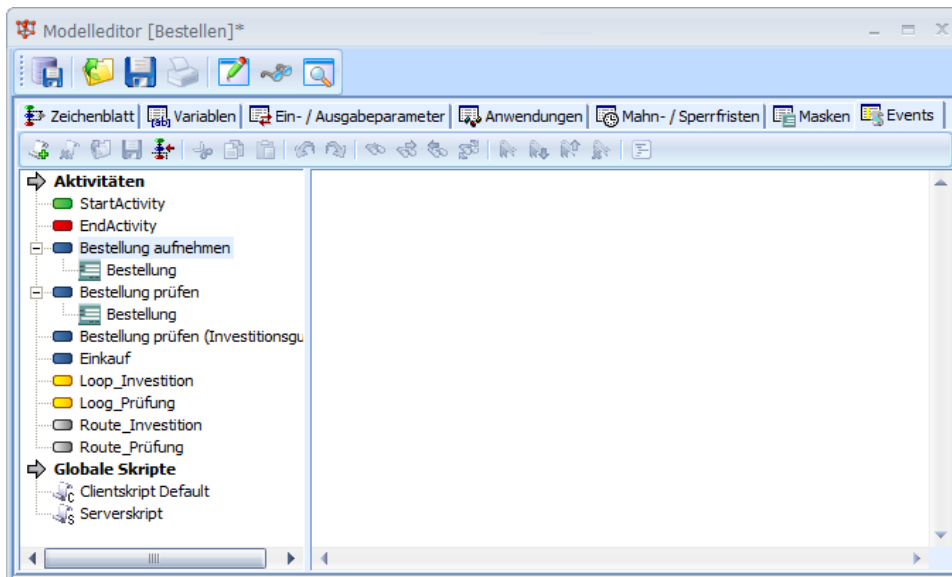
Beide Events können zusätzlich dazu verwendet werden, in der Inhalts/Detailvorschau ein Objekt über die Objekt-ID aufzurufen.

Skriptcode:

```
asfile.ContextObjIdent = ID
```

Einrichten von Eventcode

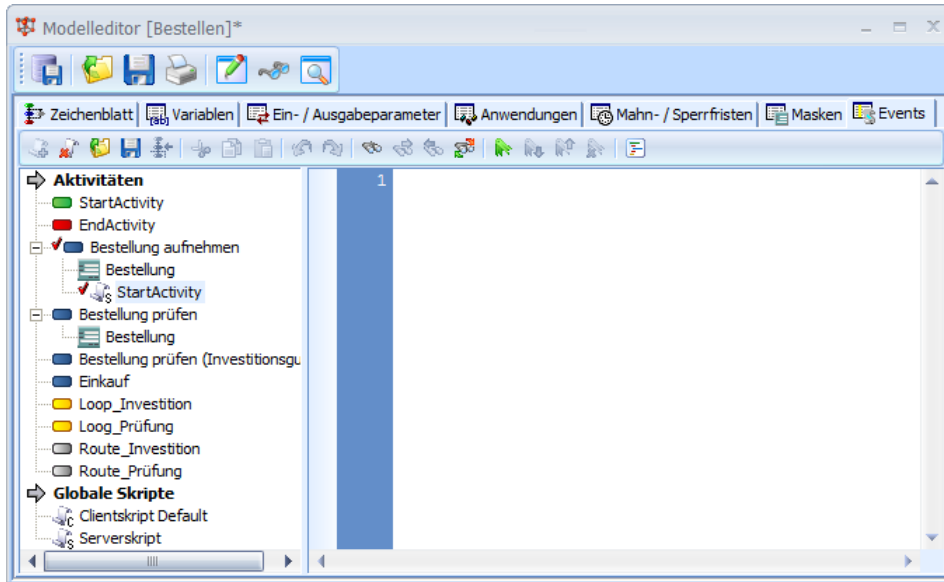
Events werden über den Modelleditor auf der Registerkarte **Events** eingerichtet.



Im Eventfenster werden alle enthaltenen Aktivitäten und deren Anwendungen aufgeführt. Serverevents können mittels Kontextmenü der Aktivität, Clientevents mittels Kontextmenü der Anwendung hinzugefügt werden. Das bedeutet, dass es je nach Anwendung (und den damit assoziierten Clienttypen) unterschiedliche Ausprägungen des jeweiligen Clientevents geben kann. enaio® client erwartet VB-Script, enaio® webclient JavaScript.

Im Editierfenster muss über das Kontextmenü die Skriptsprache eingestellt werden.

Der Eventcode kann im rechten Editierfenster, das nach der Auswahl eines Events aktiviert wird, bearbeitet werden.



Im Editierfenster kann der gewünschte Scriptcode eingegeben werden. Dabei wird durch Syntaxhighlighting und Autovervollständigung dem Anwender eine Hilfestellung zum Editieren gegeben.

Skripte können ebenfalls aus einer Datei geladen oder in einer Datei gespeichert werden.

Durch einen Rückgabewert können einige Events den Ablauf des weiteren Vorgehens beeinflussen. So kann etwa im BeforeForward-Event durch den Rückgabewert gesteuert werden, ob das Weiterleiten wirklich durchgeführt werden soll oder die Daten zunächst noch korrigiert werden sollen.

Des Weiteren können Clientevents Hinweise oder andere Meldungen an den enaio® client zurückgeben und angeben, ob die Meldungen in Messageboxen angezeigt werden sollen.

Ablage der Events

Der Eventcode wird in der Datenbank in folgenden Tabellen gespeichert:

oswfeventtypes	Hier sind alle Eventtypen (z.B. BeforeForward, StartActivity,...) mit einer Typ-ID definiert.
oswfevents	Definiert für ein Modell und eine Aktivität einen Eventtypen. Jedes Event erhält eine eigene Id. Manche Eventtypen können pro Modell/Aktivität mehrfach definiert sein (z.B. ButtonClick). Hier nimmt dann die Spalte 'Params' die Information auf, die für die Eindeutigkeit des Events sorgt (z.B. den Namen des Buttons).
oswfscripsts	In dieser Tabelle sind die eigentlichen Skripte enthalten
oswfevtscrrel	Enthält die Relationen zwischen Events (oswfevents) und Skripten (oswfscripsts). Damit ist es theoretisch möglich, ein Skript mit mehreren Events zu verbinden.

Debugging

Im enaio® client können Benutzer mit entsprechenden Systemrollen den Debug-Modus für Workflowevents einschalten. Im Debug-Modus werden die Events nicht ausgeführt, sondern bei der entsprechenden Aktivität das Skript im Eventeditor geöffnet. Dort kann dann das Skript schrittweise ausgeführt werden. Die aktuellen Werte der Variablen werden angezeigt, die Werte können geändert werden.

Im Debug-Modus kann das Skript nicht editiert werden.

Beispiele für Eventcodes

Nachfolgend werden einige Beispiele aufgeführt, in denen häufig über Eventcode modellspezifische Aktionen ausgeführt werden. Die Beispiele beschreiben das allgemeine Vorgehen. Die beigefügten Codefragmente sollen das Prinzip erläutern und sind im Allgemeinen nicht direkt lauffähig, da zum Zwecke der Übersichtlichkeit auf Initialisierungsfunktionen und Deklarationen verzichtet wurde.

Zugriff auf die COM-Schnittstelle des Clients

Aus einem clientseitigen Workflowevent kann auf die COM-Schnittstelle des enaio® -Clients über das Application - Objekt zugegriffen werden. Dieses Objekt steht in den clientseitigen Workflowevents immer zur Verfügung und muss nicht per Hand erstellt werden.

```
Application.getobjectname oid, oname
```


Soll aus einem beliebigen Scriptcode auf die COM-Schnittstelle des enaio®-Clients zugegriffen werden, so kann dies über die `CreateObject`-Funktion erfolgen. Nach der Bereitstellung des Objektes können die Funktionen zum Zugriff auf das DMS verwendet werden.

```
On Error Resume Next
Dim objax
Set objax = Application
If objax Is Nothing Then
    'Objekt nicht vorhanden
End If
```

Wichtig: In einem clientseitigen Workflowevent sollte auf das `optmial_AS.application` Objekt nicht über `CreateObject` zugegriffen werden, da sonst die Möglichkeit zum Debuggen des Skripts verloren geht.

Zugriff auf die Funktionsbibliothek `oxvbbas.dll`

Die Funktionsbibliothek `oxvbbas.dll` kapselt häufig verwendete Funktionen zum Zugriff auf die COM-Schnittstelle des Clients und erleichtert die Erstellung von Anwendungen. Oft werden in VB-Beispielcodes Routinen aus dieser Bibliothek verwendet.

Vorbelegung von Prozessvariablen beim Starten eines Prozesses

In diesem Beispiel wird angenommen, dass Prozessvariablen schon vor der ersten Aktivität initialisiert, d.h. mit Werten belegt werden sollen. Etwa könnten Werte aus einer externen Datei oder Datenbank gelesen oder einer Variable der Zeitpunkt des Prozessstarts übergeben werden.

In diesem Falle wird ein serverseitiges `StartActivity`-Event für die Startaktivität festgelegt. In unserem Beispiel nehmen wir an, dass eine String-Variable zur Aufnahme der aktuellen Startzeit belegt werden soll. Alternativ können Variablen auch durch eine Vorbelegung initialisiert werden. Dies funktioniert jedoch nur bei festen Werten für die Variable. Soll der Wert berechnet werden, so muss dies durch ein Skript erfolgen.

```
Dim sVarValue
'lesender Zugriff
sVarValue = VarName.value
sVarValue = "hello world"
'schreibender Zugriff
VarName.value = sVarValue
```

Dabei ist `VarName` die Bezeichnung der Variablen. In Serverskripten können Prozessvariablen direkt über ihren Namen referenziert werden. In Clientskripten erfolgt dies über das Hilfsobjekt `wfVariables`. Dieses Hilfsobjekt ist eine Sammlung aller Workflowvariablen.

Die Referenzierung auf die einzelnen Variablen erfolgt folgendermaßen:

```
Dim sVarValue
'lesender Zugriff
sVarValue = VarName.value
sVarValue = "hello world"
'schreibender Zugriff
VarName.value = sVarValue
```

Starten einer Anwendung aus einer Aktivitätenmaske

Ist im Prozess vorgesehen, an einer bestimmten Aktivität eine Anwendung zu starten, etwa um ein Dokument in der Workflowakte anzuzeigen oder ein Fremdsystem anzufragen, so kann dies über das „`ButtonClick`“-Event ausgelöst werden.

Hinzufügen eines Dokumentes zur Workflowakte

Steht an einer Aktivität die Anforderung, zur Bearbeitung des Vorganges ein Dokument der Workflowakte hinzuzufügen, etwa um ein Schreiben zum Geschäftsfall zu erstellen, so kann das Dokument zunächst über die enaio® client-COM-Schnittstelle dem DMS hinzugefügt und dann in die Workflowakte eingefügt werden.

Skripte dieser Art werden im Allgemeinen dem Event „ButtonClick“ zugeordnet. Zu diesem Zwecke muss auf der Aktivitätenmaske eine Schaltfläche vorgesehen sein.

Zunächst wird eine Instanz der enaio® client-COM-Schnittstelle erzeugt. Danach wird ein neues Dokument im DMS angelegt. Wie dies erfolgt, kann der COM-Programmierreferenz entnommen werden. Danach wird das Objekt in die Workflowakte eingefügt. Dies erfolgt über die bereits vorinitialisierte Variable `wfFile` und dessen Methode `wfFile.addfileobject` ist eine Sammlung vom Typ `IWFFFile` zum Zugriff auf die Workflowakte. Eine genaue Beschreibung ist der Programmierreferenz im Anhang zu entnehmen. Nach dem Hinzufügen des Dokumentes wird zunächst das Datenblatt geöffnet und danach das Dokument selbst.

```
'Erzeugen einer Instanz der OS:Client-COM-Schnittstelle
Dim myax
stmpfile = 'Standort der Übergabedatei
Set myax = CreateObject("oxvbscript.function")
'... Initialisierungen des neuen Dokumentes ...
'hier wird das neue Dokument in das DMS eingefügt
myax.InsertIntoDocument stmpfile, newid, newtype
'Hinzufügen des Dokumentes zur Workflowakte
wfFile.AddFileObject newid, newtype, True, 2
Dim mxax
Set mxax = Application
'Datenblatt des Dokuments öffnen
mxax.OpenDataDlg newid, newtype, 1
'Dokument zur Bearbeitung öffnen
mxax.OpenObjectId newid, newtype, 1
```

Plausibilitätsprüfungen vor dem Weiterleiten

Soll vor dem Weiterleiten geprüft werden, ob bestimmte vorgegebene Plausibilitäten eingehalten wurden, so kann dies im Event „BeforeForward“ durchgeführt werden. Dieses Event kann an den enaio® client einen Rückgabewert übergeben, über den gesteuert werden kann, ob die Aktivität wirklich beendet werden soll oder ob der Anwender gegebenenfalls die eingegebenen Daten überarbeiten muss.

In folgendem Codefragment wird geprüft, ob der eingegebene Wert der Bestellung größer als 0 und kleiner als 10000 € ist. Ist dies nicht der Fall, so erscheint eine MessageBox, die zur Änderung der eingegebenen Daten auffordert.

```
If wfVariables.Wert.Value > 10000 Then
    scriptresult.Resultcode = -1
    scriptresult.ResultString = "Artikel können nur bis zum Wert von max. 10000 €
bestellt werden."
End If
If wfVariables.Wert.Value <= 0 Then
    scriptresult.Resultcode = -1
    scriptresult.ResultString = "Bitte geben Sie einen gültigen Wert ein."
End If
If scriptresult.ResultCode = -1 Then
    scriptresult.ShowResultString = True
End If
```

Im nächsten Beispiel wird geprüft, ob an der Aktivität mindestens ein Dokument vom Typ „Geschäftsfall“ angelegt wurde. Ist dies nicht der Fall, so wird eine MessageBox ausgegeben und das Weiterleiten abgebrochen.

```
doctype = 131072
docname = "Geschäftsfall"
Set fileobjs = wfFile.fileobjects
bFound = False
For i = 0 To fileobjs.count-1
    Set obj = fileobjs.item(i)
    If obj.objecttype = doctype And obj.isworkspaceobject Then
```

```

        bFound = True
    End If
Next
If bfound = False Then
    scriptresult.ResultString = "Im Arbeitsbereich der Akte muss mindestens ein
Objekt vom Typ " & docname & " enthalten sein."
    scriptresult.ResultCode = -1
End If
If scriptresult.ResultCode = -1 Then
    scriptresult.ShowResultString = True
End If

```

Übertragung von Objekten in das DMS

Über Scriptcode können Objekte, die sich bisher nur in der Workflowakte befanden, in das DMS übertragen werden. Damit sind diese Objekte auch über eine normale Recherche im enaio® client erreichbar.

```

Dim myax
Set myax = Application
'Objekt vom Typ Geschäftsfall aus der Akte ermitteln
For lq = 0 To wffile.fileobjects.count -1
    lidtmp = wffile.fileobjects(lq).ID
    lTypetmp = wffile.fileobjects(lq).objecttype
    myax.getobjectname lidtmp, lnametmp
    If lcase(lnametmp) = "Geschäftsfall" Then
        lidabl = wffile.fileobjects(lq).ID
        lTypeabl = wffile.fileobjects(lq).objecttype
    End If
Next
'OrdnerID und RegisterID belegen
ordid = 4711
regid = 10234
'Objekt ins DMS verschieben
lRet = myax.moveobject(lidabl, lTypeabl, ordid, regid)

```

Recherche im DMS

Sollen Verknüpfungen zwischen DMS und Workflowakte durchgeführt werden, so ist es mitunter notwendig, zunächst die entsprechenden Objekte im DMS zu identifizieren. Folgender Beispielcode zeigt eine Anfrage auf einen Dokumenttypen. Dabei referenziert myax auf ein Objekt der Client-COM-Schnittstelle.

```

Dim myax
Set myax = Application
oid = 62
sDocName = "Dokumenttyp"
myax.getobjectname oid, oname
TmpDateiName = "c:\temp\hallo.txt"
With oxHelp
    .WriteProfString "ANFRAGE", "SCHRANK", oname, TmpDateiName
    .WriteProfString "ANFRAGE", "SCHRANK-ID", oid, TmpDateiName
    .WriteProfString "ANFRAGE", "DOKUMENT", sDocName, TmpDateiName
    .WriteProfString "ANFRAGE", "FELD1", "holgi", TmpDateiName
End With
sRet=Myax.StartDocRequest(TmpDateiName)
If Trim(sRet) <> vbNullString Then
    If Left(sRet, 5) <> "ERROR" Then
        Set FSO = CreateObject("Scripting.FileSystemObject")
        Set ResultFile = FSO.OpenTextFile(sRet)
        Do While ResultFile.AtEndOfStream = False

```

```

        AktIndex = ResultFile.ReadLine
        If Trim(AktIndex) <> vbNullString Then
            OrdnerAnfragen.add AktIndex
        End If
    Loop
    ResultFile.close
End If
End If

```

Einfügen eines neuen Dokumentes in das DMS

Soll ein neues Dokument im DMS eingefügt werden, so kann dies anhand des folgenden Codes durchgeführt werden. Es wird dabei vorausgesetzt, dass vorher der Standort, also die ID des entsprechenden Ordners ermittelt wurde.

```

Dim myax
Set myax = Application
oid = 24
sDocName = "Dokumenttyp"
sWert = "Hallo"
myax.getobjectname oid, oname
TmpDateiName = "c:\temp\Einfügen.txt"
With oxHelp
    .WriteProfString "EINFÜGEN", "SCHRANK", oname, TmpDateiName
    .WriteProfString "EINFÜGEN", "SCHRANK-ID", oid, TmpDateiName
    .WriteProfString "EINFÜGEN", "DOKUMENT", sDocName, TmpDateiName
    .WriteProfString "EINFÜGEN", "FELD1=Feldname", sWert, TmpDateiName
End With
Dim objax
set objax = createObject("oxvbscript.function")
lRet=objax.insertintodocument(TmpDateiName,did,dtype)
If lRet<>0 Then
    'hier gab es einen Fehler beim Einfügen
End If

```

Festlegen des Folgebearbeiters

Mitunter ist es notwendig, innerhalb eines Workflowprozesses eine dynamische Entscheidung darüber zu treffen, wer als Folgebearbeiter eingetragen werden soll. Dies kann dann auftreten, wenn diese Entscheidung nicht ausschließlich durch Workflowvariablen, sondern auch durch externe Datenquellen bestimmt wird. Weiterhin kann es sein, dass die Modellierung im Prozessmodell einen enormen Aufwand erfordert, wenn für die benötigten Transitionen eine große Anzahl an Bedingungen vorliegt. Über Scriptcode kann der Folgebearbeiter dynamisch festgelegt werden. Dies wird im Event „StartActivity“ festgelegt.

Im folgenden Scriptcode wird als Bearbeiter der Aktivität der Ersteller des Prozesses zugewiesen. Das ist dann sinnvoll, wenn nach der Bearbeitung von Aktivitäten der Ersteller benachrichtigt werden soll oder dieser weitere Entscheidungen zu treffen hat.

```

'Der Bearbeiter dieser Aktivität ist der Ersteller des Prozesses
thisProcess.SetActivityPerformer
thisProcess.CurrentActivity.ActivityId,thisProcess.Creator

```

Alternativ kann auch eine beliebige andere Rolle oder Person als Folgebearbeiter eingesetzt werden. Dazu wird dann die GUID der gewünschten Rolle oder Person als Parameter angegeben.

```

Dim sUserID
sUserID = 'Organisationsobjekt ID des Benutzers
thisProcess.SetActivityPerformer thisProcess.CurrentActivity.ActivityId,sUserID

```

Navigation in der Organisationsstruktur

Über Funktionen zur Ermittlung von Vorgängern und Nachfolgern kann per Scriptcode eine Navigation innerhalb der Organisationsstruktur durchgeführt werden. Dabei behilflich sind Funktionen zum Zugriff auf den Namen der

Organisationseinheit und auf Attribute. Die Ergebnisse der Navigation können beispielsweise zur Festlegung des Folgebearbeiters dienen.

In folgendem Beispiel wird ermittelt, ob die aktuelle Rolle einem „Team“ zugeordnet ist und wenn dies der Fall ist, ob es einen „Teamsprecher“ gibt. Die Suche wird über die Bezeichnungen der Organisationseinheit durchgeführt und hängt vom aktuellen Modell ab. Analog dazu kann nach anderen Organisationseinheiten gesucht werden.

```
set roles = wforganisation.GetOrgObjectsByClassName("Rolle")
for i = 0 To roles.count-1
    set role = roles.item(i)
    'Das Prozentzeichen wird zur DB durchgereicht und dient als Joker
    Set teams = role.GetPredecessorsByClassName("Team%", true)
    For j = 0 To teams.count-1
        'Teamsprecher ermitteln
        Set teams1 = teams.item(j).GetSuccessorsByName("%Teamsprecher%")
        If teams1.count > 0 Then
            r_ts.name.value = teams1.item(0).name
            r_ts.guid.value = teams1.item(0).id
            bFound = True
        Exit For
    End If
Next
Next
```

Ermittlung aller Benutzer und ihrer Rollen

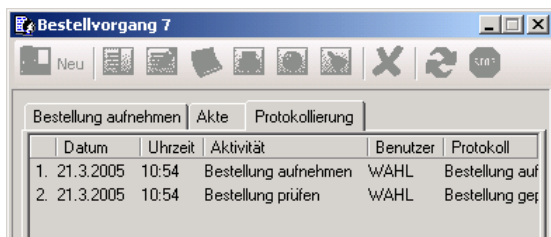
Im folgenden Beispiel sollen alle Benutzer und ihre Rollen ermittelt werden. Zu beachten ist hier, dass den Funktionen GetXYZByName auch Jokerzeichen übergeben werden können. Es werden dann alle Objekte ermittelt, bei denen ein Teil der Bezeichnung mit dem Suchstring übereinstimmen.

```
'Das Prozentzeichen wird zur DB durchgereicht und dient als Joker
Set users = wforganisation.GetOrgObjectsByClassName("Person%")
For i = 0 To users.count -1
    Set username = users.item(i)
    'Eine Variable, hier li_users, vom Typ der li_user Struktur muss vorhanden sein,
    'damit eine Liste von dieser Struktur erstellt werden kann
    Set UserItem = wfvariables.li_users.CreateListItem()
    UserItem.guid.value = username.id
    UserItem.g_name.value = username.name
    Set roles = username.Predecessors
    For j = 0 To roles.count -1
        set role = roles.item(j)
        'In der Struktur li_user muss eine weitere Struktur li_roles vorhanden sein,
        'damit auch von dieser Struktur eine Liste erstellt werden kann
        Set RoleItem = UserItem.li_roles.CreateListItem()
        RoleItem.guid.value = role.id
        RoleItem.g_name.value = role.name
        'Name und ID werden der Liste li_roles hinzugefügt
        itemid = UserItem.li_roles.AddListItem(RoleItem)
    Next
    'Name und ID werden der Liste li_users hinzugefügt
    itemid = wfvariables.li_users.AddListItem(UserItem)
Next
```

Einrichtung des Protokolls

An den Bearbeitungsschritten im enaio® client kann in den bisherigen Verlauf des Prozesses Einsicht genommen werden, wenn dies entsprechend konfiguriert ist. Die Protokollierung ist zur Information der durchlaufenen Aktivitäten gedacht und kann für jedes Prozessmodell separat konfiguriert werden.

Durch die separaten Einstellungen für das Protokoll können durch Scriptcode genau die Einträge erstellt werden, die für den Anwender auch relevant sind.



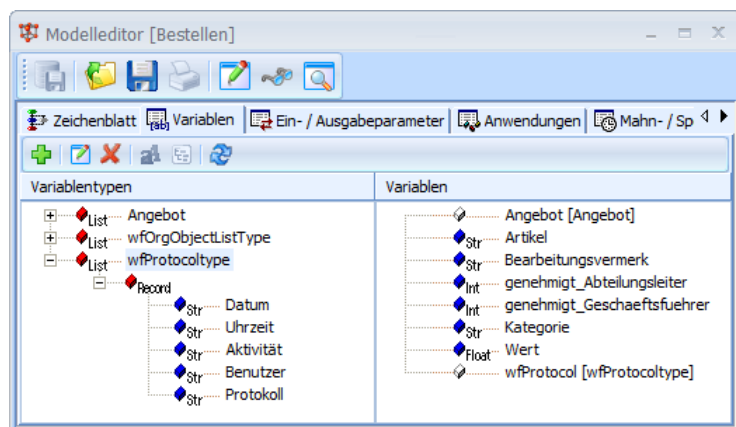
	Datum	Uhrzeit	Aktivität	Benutzer	Protokoll
1.	21.3.2005	10:54	Bestellung aufnehmen	WAHL	Bestellung auf
2.	21.3.2005	10:54	Bestellung prüfen	WAHL	Bestellung ge

Die Einrichtung des Protokolls erfolgt in mehreren Schritten. Zunächst muss ein Variablentyp definiert und dann eine Protokollvariable angelegt werden, die durch den gesamten Workflowprozess geführt wird. Sodann muss an den Aktivitäten, wo eine Protokollierung erfolgen soll, im BeforeForward-Event oder an weiteren Stellen diese Variable gefüllt werden. Die Darstellung des Protokolls an den Aktivitäten erfolgt automatisch.

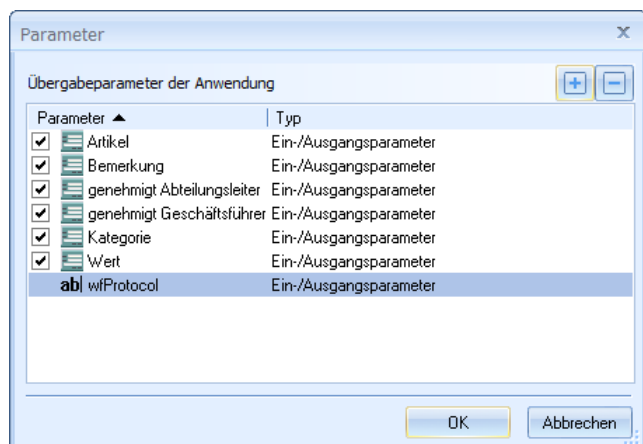
Definition des Protokollvariablentyps

Die Variable zur Aufnahme des Protokolls ist ein zusammengesetzter nutzerdefinierter Typ. Dieses Format wurde gewählt, um im Prozess die Anzahl und Beschriftung der Spalten definieren zu können.

Variable und Variablentyp sind für jedes Modell vordefiniert.



Nachdem die Variable im Prozessmodell bekannt ist, muss sie noch den einzelnen Anwendungen übergeben werden. Dies geschieht bei der Konfiguration der Übergabeparameter der Anwendung. Beim Hinzufügen des Parameters muss der Name „WFProtocol“ angegeben und als Typ „Eingabe- und Ausgangsparameter“ definiert werden. Damit können die Protokolleinträge der bisher abgelaufenen Aktivitäten an den Client übergeben und vor dem Weiterleiten erweitert werden.

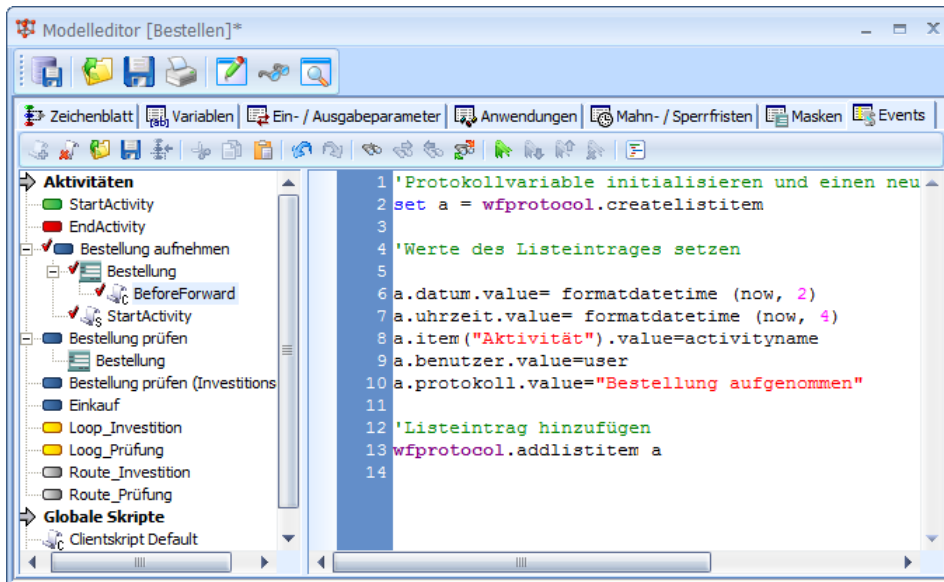


Für jede Aktivität wird dann der Anwendungsparameter 'WFProtocol' der Workflowvariable 'WFProtocol' zugeordnet.

Wurden diese Schritte für alle Anwendungen und Aktivitäten durchgeführt, so steht das Protokoll zunächst an allen Bearbeitungsschritten zur Verfügung. Im nächsten Schritt wird betrachtet, wie die Protokollvariable mit sinnvollen Werten gefüllt werden kann.

Anlegen der Protokollevnts

Wie oben schon angedeutet, müssen die Anwendungen die Protokollvariable selbst füllen. Praktischerweise erfolgt dies mit Hilfe von Eventcode am Einsprungspunkt „BeforeForward“. Selbstverständlich kann dies auch analog dazu an anderen Einsprungspunkten, wie etwa „BeforeOpen“ oder „ButtonClick“ geschehen, wenn aufgezeichnet werden soll, dass ein Anwender eine Aktivität personalisiert oder einen Button betätigt hat.



Der Code ist relativ einfach und wird hier nochmals dargestellt. Zu beachten ist, dass an allen Aktivitäten, an denen die Protokollierung erweitert werden soll, dieser Eventcode hinterlegt werden muss. Der Code kann ergänzend zu anderen Skripten eingefügt werden. Sinnvollerweise fügt man diesen jedoch am Ende des Scriptcodes hinzu, um bei eventuell auftretenden Fehlern im Skript keine ungültige Protokollierung durchzuführen.

```
'Protokollvariable initialisieren und einen neuen Listeeintrag erzeugen
set a = wfprotocol.createlistitem
'Werte des Listeeintrages setzen
a.datum.value= formatdatetime (now, 2)
a.uhrzeit.value= formatdatetime (now, 4)
a.item("Aktivität").value=activityname
a.benutzer.value=user
a.protokoll.value="Bestellung aufgenommen"
'Listeeintrag hinzufügen
wfprotocol.addlistitem a
```

Die einzelnen angesprochenen Felder hängen davon ab, wie oben der Typ der Protokollvariable definiert wurde. So können weitere Felder angesprochen oder der Eintrag in der Spalte „Protokoll“ weiter formatiert werden.

Wie hier am Beispiel zu sehen ist, müssen Variablen, die Umlaute enthalten, über das Konstrukt: `Variable.item(„<Variablenname>“).Value` angesprochen werden. Zum Setzen des aktuellen Datums oder der Uhrzeit kann die VBScript Funktion `formatdatetime` verwendet werden.

Integration der Digitalen Signatur und Schrittautorisierung

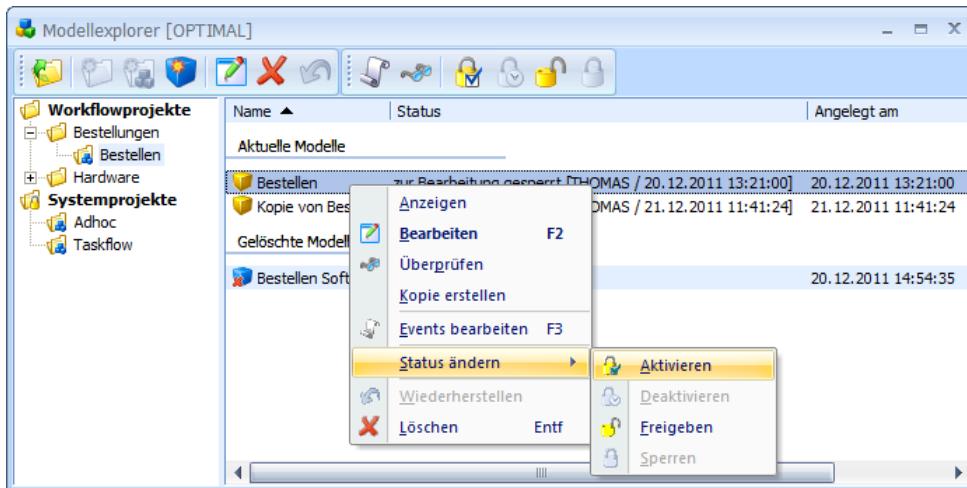
Durch die Integration der Digitalen Signatur können Dokumente innerhalb des Workflows signiert oder eine vorhandene Signatur überprüft werden. Die Einbindung der Digitalen Signatur in den integrierten Workflow erlaubt die rechtssichere Autorisierung einzelner Vorgangsschritte und Eingaben.

Alternativ kann auch eine Schrittautorisierung durch einen Passwortdialog vorgesehen werden. Dabei wird der Anwender vor dem Weiterleiten von Aktivitäten aufgefordert, zunächst sein Login-Passwort in einen Dialog einzugeben.

Die Einstellung, ob und welche Art von Autorisierung durchgeführt werden soll, wird in einem Extended Attribute zu den Aktivitäten vorgenommen.

Aktivierung des Modells

Workflowmodelle durchlaufen bis zur Aktivierung unterschiedliche Stadien. Dies ist dadurch begründet, dass die Workflow-Engine nur gültige Modelle verarbeiten kann und während des Designs möglicherweise nicht alle Schritte bis zu Ende definiert sind. Aus diesem Grunde hat jedes Modell einen bestimmten Status, der im Workfloweditor eingestellt werden kann. Gleichzeitig wird dadurch sichergestellt, dass nicht mehrere Bearbeiter ein Modell bearbeiten und damit die Änderungen gegenseitig überschreiben.



Folgende Arten von Statusinformationen können für die einzelnen Modelle gesetzt werden:

Status	Bedeutung
Sperren	Modell ist zur Bearbeitung gesperrt (vom Bearbeiter ausgecheckt)
Freigeben	Modell ist zur Bearbeitung freigegeben (vom Bearbeiter eingeccheckt)
Verwenden	Modell ist am Server bekannt gegeben, es können aber keine neuen Prozesse gestartet werden. Prozesse, die mit diesem Modell gestartet wurden, laufen bis zum Ende weiter.
Aktivieren	Modell ist am Server aktiviert, es können neue Prozesse gestartet werden.

Zu beachten ist, dass pro Workflowfamilie immer maximal ein aktives Modell vorhanden sein kann. Werden in einem Workflowprojekt mehrere abhängige Prozessmodelle verwaltet, so ist der Engine immer ein aktuelles Modell bekannt, das zur Bearbeitung neuer Prozesse verwendet werden kann.

Bei einem Versionswechsel von Modellen kann man also das bisher laufende Modell auf den Status „in Benutzung“ setzen und ein neues Modell aktivieren. Bereits gestartete Prozesse laufen dann mit dem alten Modell bis zu Ende weiter, neue Prozesse verwenden hingegen das aktivierte Modell.

Modelle können nicht aktiviert oder auf den Status „In Benutzung“ gesetzt werden, solange sie syntaktisch inkorrekt sind, d.h. wenn etwa Aktivitäten keine Anwendungen zugeordnet sind. Diese Prüfung erfolgt während des Umschaltens des Status in der Workflow-Engine. Falls die Prüfung fehlschlägt, wird eine Fehlermeldung ausgegeben und der bisherige Status wird beibehalten.

Sollen bereits gestartete Prozesse nicht weiter ausgeführt werden, so müssen diese mit Hilfe des Workflow-Administrators beendet werden. Dies ist im Kapitel *Betrieb und Administration* weiter unten beschrieben.

Zu beachten ist ebenfalls, dass nur Workflowmodelle der aktiven Organisation verwendet werden können, obwohl Workflowmodelle nicht aktiver Organisationen den Status 'Aktiviert' erhalten können.

Import und Export von Workflowprojekten

Der Import und Export von Workflowprojekten dient dem Austausch von Modellen, Organisationsstrukturen und Masken zwischen Entwicklungs-, Test- und Produktivsystemen. Da das Design der Prozessmodelle einen relativ hohen zeitlichen Aufwand in Anspruch nimmt, ist es notwendig, dass Änderungen und Erweiterungen komplett in andere Zielsysteme übernommen werden können, ohne dass diese im Einzelnen nachvollzogen werden müssen.

Zu diesem Zwecke steht die Import-/Exportfunktionalität des Workfloweditors zur Verfügung. Organisationen, Projekte, Familien und Modelle können über das Kontextmenü im Workflow-Editor exportiert und importiert werden.

Im Regelfall wird die Aufwärtskompatibilität von Konfigurationen in den verschiedenen enaio®-Versionen angestrebt, sodass Konfigurationen, die in einer früheren Systemversion angelegt wurden, auch in späteren Systemversionen verwendet werden können.

Im Gegensatz dazu ist die Abwärtskompatibilität nicht sichergestellt, da die Weiterentwicklung der enaio®-Plattform neue Funktionen und damit Konfigurationserweiterungen umfasst. Konfigurationen aus späteren Systemversionen können also in früheren Systemversionen nicht verwendet werden.

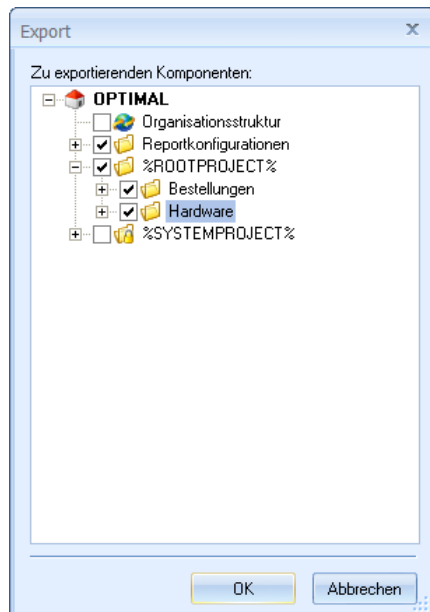
Mit großer Wahrscheinlichkeit gehen beim Versuch, Konfigurationen in frühere Systemversionen zu importieren, Einstellungen für neue Funktionen verloren und es kann zu Formatinkompatibilitäten und/oder Programmfehlern kommen. Deshalb muss beim Betrieb von Test-, Entwicklungs- und Produktivsystemen auf eine Harmonisierung der Versionen geachtet werden und es darf auf keinen Fall der Versuch unternommen werden, Konfigurationen in spätere Systemversionen zu importieren oder zu aktivieren. Das betrifft alle Konfigurationsdateien, insbesondere aber Objektdefinitionen und Workflowmodelle.

Der Austausch von Modellen sollte nur zwischen Systemen mit gleicher Version erfolgen. Ein Austausch von Modellen aus unterschiedlichen Versionen erfordert genaue Tests.

Exportieren

Exportiert werden Komponenten, die einer Organisation zugeordnet sind. Der Exportdialog listet die Organisationsstruktur und alle zugeordneten Projekte, Familien, Modelle und Reportkonfigurationen auf.

Die Daten aller Objekte, die markiert sind, werden in eine XML-Datei exportiert.



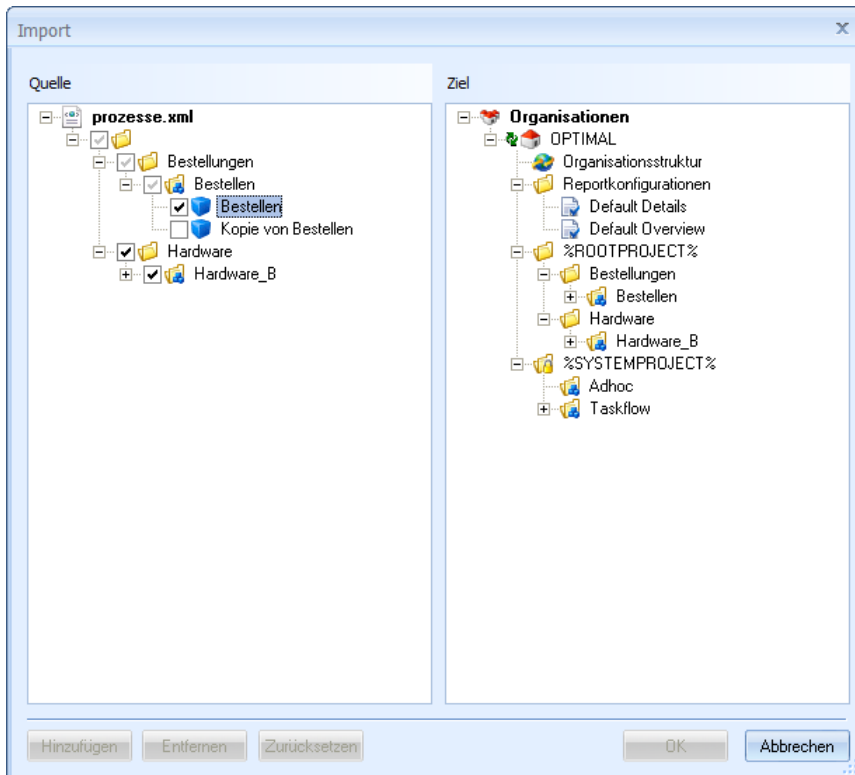
Ein einzelnes Modell, das im Modelleditor geöffnet ist, kann ebenfalls in eine XML-Datei exportiert werden.

Importieren

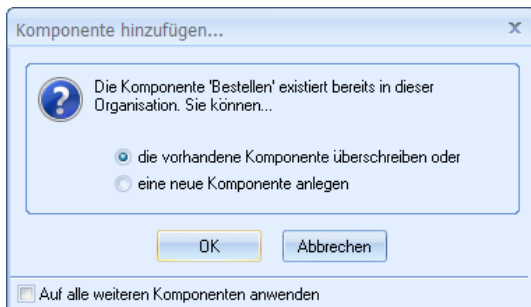
Für den Import werden Organisationsstruktur, Reportkonfigurationen, Projekte, Familien oder Modelle einer bereits eingerichteten Organisation zugeordnet.

Der Importdialog listet alle Objekte aus der gewählten XML-Export-Datei auf und eingerichteten Organisationen, Projekte, Familien und Modelle.

Objekte aus der XML-Export-Datei können markiert und per Drag&Drop auf der entsprechenden Ebene einer Organisation eingefügt werden.



Alle importierten Modelle erhalten den Status 'zur Bearbeitung freigegeben'.
Gibt es bereits Komponenten mit der gleichen GUID, wird ein Bestätigungsdialog geöffnet.



Der Import kann vorhandene Komponenten überschreiben oder importierten Komponenten eine neue GUID geben. Die Bezeichnungen bleiben erhalten.

Haben Sie ein Modell im Modelleditor geöffnet, können Sie die Daten aus einer XML-Datei importieren, die über den Modelleditor exportiert wurden.

Der Standard Ad-hoc-Workflow und der Taskflow

Um die Möglichkeiten des Workflows schnell und einfach nutzen zu können, beinhaltet enaio® Modelle, mit dem in wenigen Minuten ein System mit Workflow-Funktionen produktiv gehen kann, ohne ein tiefer gehendes Modell-Design durchzuführen. Zwei Modelle können mit geringer Anpassung eingebunden werden:

§ Taskflow

Im Unternehmensalltag bestehen sehr viele Prozesse nur aus einem Schritt, einer einfachen Zuteilung von Aufgaben. Eine solche Aufgabe kann im enaio® client auf vielfältiger Weise ausgeführt werden, hier als Taskflow gibt es die vielen Vorteile eines Workflows dazu, beispielsweise die Stellvertretungen, Personalisierungen und die Protokollierungen.

§ Standard Ad-hoc-Workflow

Dieses Modell beinhaltet häufig anzutreffende Arbeitsschritte, aus denen sich Abläufe in Organisationen zusammensetzen – Kenntnissnahmen, Freigaben, Abstimmung, Rückfrage, Mitarbeit, Überarbeiten.

Standard Ad-hoc-Workflow

Dieses Modell beinhaltet häufig anzutreffende Arbeitsschritte, aus denen sich Abläufe in Organisationen zusammensetzen – Kenntnisnahmen, Freigaben, Abstimmung, Rückfrage, Mitarbeit, Überarbeiten.

Das Standard Ad-hoc-Modell verfolgt das Ziel, in Installationen sofort einen deutlichen Mehrwert an Produktivität und Prozesssicherheit zu schaffen. Gleichzeitig kann durch das Standardmodell in einer Einführungsphase für standardisierte Workflow-Prozesse eine Evaluierung der Möglichkeiten vorgenommen werden, um so die bisher bestehenden informellen Prozesse elektronisch abzubilden, ohne die Arbeitsabläufe der Anwender in festen Strukturen zu verankern. So können dann Schritt für Schritt immer wiederkehrende Teilabläufe zunächst in Lauflisten gespeichert und dann in definierte Modelle überführt werden.

Das Standard Ad-hoc-Modell ist auf sehr verbreitete Anwendungsfälle ausgerichtet und nutzt demzufolge nicht alle technischen Möglichkeiten des enaio®-Workflows. Änderungen an dem Modell durch den Betreiber sind nicht vorgesehen, um die breite Verwendbarkeit und Wartungsfreundlichkeit des Modells sicher zu stellen. Gleichzeitig kann aber durch den Betreiber oder im Projekt ein analoges Modell entwickelt werden, um spezifische Anforderungen genauer umzusetzen.

Installation

Bei der Installation von enaio® wird das Workflowmodell `osdefaultmodels.xml` in das Verzeichnis `\clients\admin\` kopiert. Dieses Modell wird über den enaio® editor-for-workflow importiert und beinhaltet alle für den Taskflow und für den Standard Ad-hoc-Workflow notwendigen Daten.

Vergewissern Sie sich, dass die WF-Jobs `WFM::WorkItemNoti`, `WFM::WorkerJob`, `WFM::SpoolerJob` und `WFM::CheckJob` in enaio® enterprise-manager aktiviert sind.

Zusätzlich zu diesem Modell steht Ihnen für den Standard Ad-hoc-Workflow eine Objektdefinitionsdatei zur Verfügung, über die Sie einen Schrank, einen Registertyp und Dokumenttypen einrichten können, um so Protokolldateien der Ad-hoc-Vorgänge in enaio® automatisiert abzulegen. Diese Protokollverwaltung wird über eine Konfigurationsdatei gesteuert.

Alternativ können Protokolldateien als Bilddateien erzeugt und in der Ablage der Prozessverantwortlichen gespeichert werden. Für diese Lösung ist keine Protokollkonfiguration notwendig.

Konfiguration

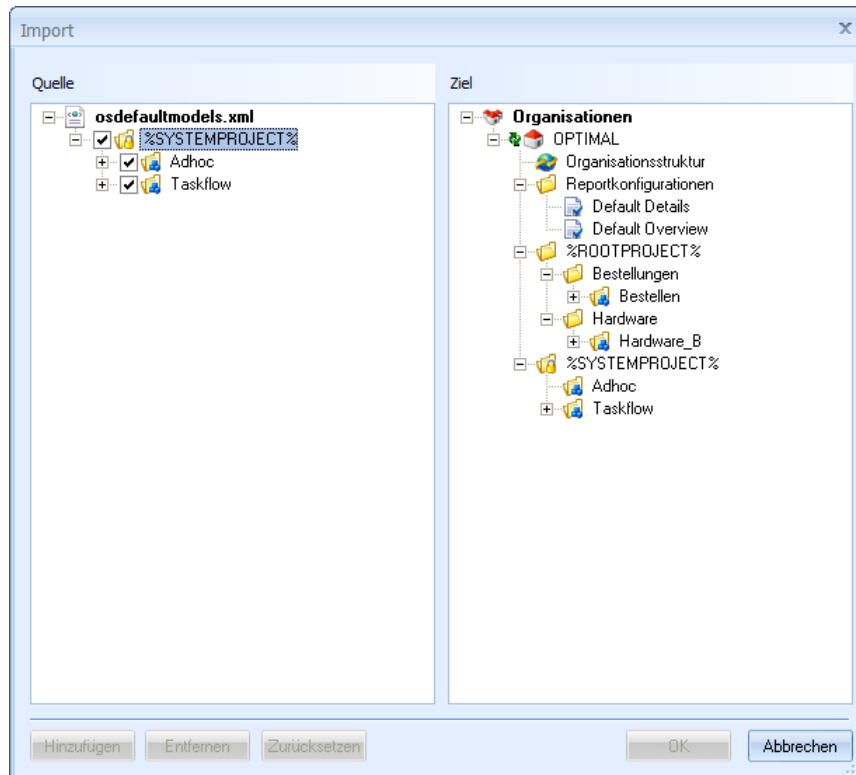
Die Konfiguration des Standard Ad-hoc-Modells erfolgt in drei Schritten:

- § Importieren und Aktivieren des Modells `osdefaultmodels.xml`.
- § Zuordnen von Personen über das Organisationsmodell.
- § Optional ist das Einrichten von Objekttypen für die Protokollierung und das Konfigurieren der Protokollablage.

Import des Workflowmodells

Das Workflowmodell importieren Sie über den enaio® editor-for-workflow.

- § Starten Sie den enaio® editor-for-workflow.
- § Wählen Sie die Funktion **Organisation importieren** aus der Symbolleiste oder dem Menü **Datei**.
- § Öffnen Sie über den Dateiauswahldialog die Datei `osdefaultmodels.xml` aus dem Verzeichnis `\clients\admin\`.
- § Markieren Sie im Bereich **Quelle** den Eintrag 'Systemprojekte' und im Bereich **Ziel** 'Organisation'. Sie können ebenfalls mit der Maus den Eintrag 'Systemprojekte' auf den Eintrag 'Organisation' ziehen.

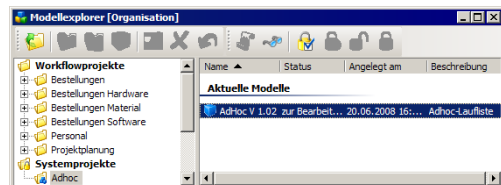


Klicken Sie dann auf den Button **Hinzufügen**. Das Projekt wird importiert und der Dialog geschlossen.

Statt der beiden Projekte Taskflow und Standard Ad-hoc-Modell in jeweils drei Sprachversionen können Sie auch nur die Einzelmodelle, die Sie einsetzen wollen, importieren. Die gewählte Sprachversion legt ausschließlich die Sprache fest, die der enaio® Server für die Kommunikation und Protokollierung wählt.

Die Workflowschritte liegen unabhängig davon in jeder Sprachversion in jeder der drei Sprachen Deutsch, Englisch und Französisch vor.

- § Öffnen Sie den Modellexplorer. Das neue Systemprojekt wird angezeigt.



Die übernommenen Modelle sind schreibgeschützt und können nicht bearbeitet werden.

- § Markieren Sie die gewünschte Sprachversion des Modells und aktivieren Sie es über die entsprechende Funktion aus dem Kontextmenü oder der Symbolleiste.

Sie erhalten bei der automatischen Überprüfung den Hinweis, dass kein Aktenverantwortlicher eingetragen ist. Diese Funktion wird über Events automatisch durch den Benutzer übernommen, der einen Prozess startet. Nach dem Ende des Vorgangs werden alle Dokumente aus der Workflow-Akte, die noch keinen Standort haben, in die persönliche Ablage übertragen.

Das Modell ist damit eingebunden und aktiviert. Im folgenden Schritt binden Sie Rollen und Personen ein.

Konfigurieren der Organisationsstruktur

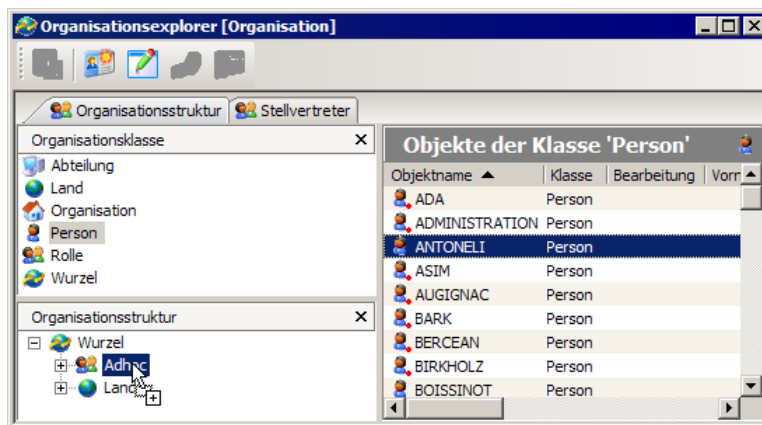
Mit der Installation von enaio® wird automatisch die Rolle 'Adhoc' angelegt und innerhalb der Organisationsstruktur auf der obersten Ebene eingefügt. Alle Personen, die Sie als Unterobjekte der Rolle 'Adhoc' in die Organisationsstruktur einfügen, können Ad-hoc-Prozesse und Taskflows im enaio® client starten, falls sie über die Systemrolle 'Client: Workflow benutzen' verfügen.

Für die Ad-hoc-Prozesse können alle Personen und Rollen aus der Organisationsstruktur im enaio® client bei der Konfiguration der Laufliste eines Ad-hoc-Prozesses als Teilnehmer eingetragen werden.

Die Organisationsstruktur bearbeiten Sie über den enaio® editor-for-workflow:

- § Starten Sie den enaio® editor-for-workflow.
- § Öffnen Sie über den Arbeitsbereich den Organisationsexplorer.

§ Ordnen Sie Objekte der Klasse 'Person' der Rolle 'Adhoc' zu.



§ Speichern Sie die Änderungen.

Mit dem Abschluss dieses Schritts können Ad-hoc-Vorgänge im enaio® client gestartet und ausgeführt werden.

Importieren Sie Benutzer im Organisationsexplorer, können Sie diesen gleich die Rolle 'Adhoc' zuordnen.

Optional können Sie für Ad-hoc-Vorgänge weitere Rollen anlegen und einbinden. Diesen Rollen werden über die Events aus dem Kontext laufender Vorgänge Benutzer, die eine Funktion im Ablauf hatten, zugeordnet.

Die folgenden Rollen sind möglich:

Rolle	Bedeutung
%Erste Empfänger der Aktivität	Ist eine Aktivität mindestens ein zweites Mal in der Laufliste aufgeführt, werden durch diesen Eintrag, diejenigen Personen oder Rollen als Teilnehmer angegeben, die für die Aktivität beim ersten Mal als Teilnehmer eingetragen wurden.
%Erste Bearbeiter der Aktivität	Ist eine Aktivität mindestens ein zweites Mal in der Laufliste aufgeführt, werden durch diesen Eintrag, diejenigen Personen oder Rollen als Teilnehmer angegeben, die die Aktivität beim ersten Mal ausgeführt haben.
%Vorherige Empfänger der Aktivität	Ist eine Aktivität mindestens ein zweites Mal in der Laufliste aufgeführt, werden durch diesen Eintrag, diejenigen Personen oder Rollen als Teilnehmer angegeben, die für die Aktivität beim vorherigen Mal als Teilnehmer eingetragen wurden.
%Vorherige Bearbeiter der Aktivität	Ist eine Aktivität mindestens ein zweites Mal in der Laufliste aufgeführt, werden durch diesen Eintrag, diejenigen Personen oder Rollen als Teilnehmer angegeben, die die Aktivität beim vorherigen Mal ausgeführt haben.
%Letzte Bearbeiter	Derjenige Teilnehmer, der zuletzt einen Vorgangsschritt weitergeleitet hat.
%Prozessinitiator	Derjenige, der den Vorgang gestartet hat.

Diese Rollen können Sie als Objekte der Klasse 'Rolle' mit dem Organisationsexplorer anlegen und der Organisationsstruktur zuordnen.

Legen Sie diese Rollen an, werden Sie gegebenenfalls auch in anderen Workflowvorgängen dem Benutzer zur Auswahl angeboten. Dort haben sie allerdings ohne die Einbindung durch Events keine Funktion.

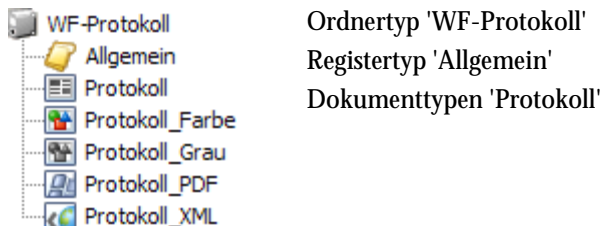
Konfiguration einer Protokollierung

Bestandteil des Modells 'Adhoc' ist die Aktivität 'Protokoll', die, gesteuert durch Events, alle Daten der bereits abgeschlossenen Aktivitäten in einem Protokoll zusammenstellt. Die Aktivität wird, wenn Sie in eine Laufliste aufgenommen wurde, automatisch durchlaufen. Das Protokoll befindet sich danach als Dokument in der Workflowakte. In der folgenden Aktivität kann das Protokoll einem Dokumenttyp zugeordnet, verschlagwortet und an

einem Standort gespeichert werden. Geschieht dies nicht, wird das Protokoll nach Ablauf des Vorgangs automatisch in die persönliche Ablage des Benutzers gelegt, der den Vorgang gestartet hat.

Über eine Konfiguration kann das Format für das Protokoll festgelegt werden und ein Standort konfiguriert werden, an dem das Protokoll automatisch gespeichert wird. Diese Konfiguration ist optional, ohne Konfiguration wird das Protokoll als Bilddokument in die Workflowablage gelegt.

In der Objektdefinitionsdatei sind folgende Objekttypen für die Ad-hoc-Workflowprotokollierung definiert:



Die Konfigurationsdatei ist auf diese Objektdefinition abgestimmt. Erzeugt wird für jedes Jahr ein Ordner, für jeden Monat ein Register und dort ein Dokument mit den Protokolldaten.

Vorschlagwortet wird das Dokument mit dem Prozessnamen, der Prozess-ID und mit dem Betreff aus den Vorgangsschritten:

Übernahme der Objektdefinition

Mit dem enaio® editor öffnen Sie die Objektdefinitionsdatei `AdHoc_asobjdef.xml` aus dem Verzeichnis `\clients\admin\`.

Sie übernehmen den Schrank, das Register und mindestens einen Dokumenttyp in die System-Objektdefinition. Sie benötigen nur den Dokumenttyp, der dem gewünschten Protokollformat entspricht.

Danach speichern Sie die Objektdefinition und passen die Tabellen an.

Informationen zum enaio® editor finden Sie im entsprechenden Handbuch.

Für die weitere Konfiguration benötigen Sie die ID der Objekttypen. Diese ID finden Sie als ersten Eintrag in der Spalte 'Objekttyp' des Eigenschaften-Dialogs.

Beispiel:

Der Dokumenttyp 'Protokoll' hat die ID 131115

Datenbank	
Bezeichnung	Protokoll
Tabellenname	object103
Objekttyp	131115 / 2 / 43
Interner Name	WF_Protokoll_TIF

Benutzer mit der Rolle 'Adhoc' benötigen im Sicherheitssystem die Zugriffsrechte, um die hier konfigurierten Ordner, Register und Dokumente für die Protokollierung anlegen zu können.

Protokoll-Konfigurationsdatei

Über die Protokoll-Konfigurationsdatei `AdHoc_Config.xml` legen Sie fest, in welchem Dateiformat die Protokolle erstellt werden und wohin sie gespeichert werden.

Die Datei wird in das Verzeichnis `\etc` des Datenverzeichnisses ihrer enaio®-Installation installiert. Ändern Sie die Datei nicht, werden Protokolldateien im PDF-Format erstellt. Löschen Sie die Konfigurationsdatei, wird nur eine Bilddatei ohne Dokumenttyp erzeugt.

Voraussetzungen für Protokolle im PDF-Format sind ein JRE ab der Version 1.3 auf jedem beteiligten enaio® Server, das Setzen der Umgebungsvariable `JAVA_HOME` auf den Java-Installationspfad `... \java\jdk\` und aktivierte Konvertierungsengines.

Die Protokoll-Konfigurationsdatei hat folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<Protocol Type="NONE">
```

Als Protokoll-Typ geben Sie 'XML', 'TIF', 'PDF' oder 'NONE' an. Wollen Sie keine automatische Übernahme an einen Standort, sondern nur ein Protokoll in der Akte, geben Sie 'NONE' an. Dann wird aus der folgenden Konfiguration nur der Dokumenttyp für die Protokolldatei übernommen.

Wollen Sie ein Bilddokument, geben Sie 'TIF' an. Das Dateiformat für die Ablage - JPG oder TIFF - wird automatisch durch die Zuordnung zu einem Bild-Dokumenttyp festgelegt. Wollen Sie ein XML-Dokument, geben Sie 'XML' an.

```
<Folder ObjectId="0" InternalName="WF_Protokoll">
```

Sie geben die Ordner-ID und den internen Namen des Ordnerstyps an.

```
<Fields>
```

```
<Family InternalName="WF_Family" />
```

```
<Year InternalName="WF_Year" />
```

```
</Fields>
```

Sie geben die internen Feldbezeichnungen für die Ordnerfelder an, denen der Modellname und das Jahr zugeordnet werden.

```
</Folder>
```

```
<Register ObjectId="6488064" InternalName="WF_General">
```

Sie geben die Registertyp-ID und den internen Namen des Registertyps an.

```
<Fields>
```

```
<Month InternalName="WF_Month" />
```

```
</Fields>
```

Sie geben die internen Feldbezeichnungen für das Registerfeld an, dem der Monat zugeordnet wird.

```
</Register>
```

```
<Document ObjectId="458752" InternalName="WF_Protocol_PDF">
```

Sie geben die Dokumenttyp-ID und den internen Namen des Dokumenttyps an, den Sie in der Objektdefinition für das gewünschte Dateiformat eingerichtet haben.

```
<Fields>
```

```
<ProcessName InternalName="WF_ProcessName_PDF" />
```

```
<ProcessId InternalName="WF_ProcessId_PDF" />
```

```
<Comment InternalName="WF_Comment_PDF" />
```

```
</Fields>
```

Sie geben die internen Feldbezeichnungen für die Dokumenttypfelder an, denen Prozessname, Prozess-ID und Betreff zugeordnet werden.

```
</Document>
```

```
</Protocol>
```

Sie können diese Konfiguration ebenfalls an andere Archivobjekttypen anpassen. Eine Struktur aus Ordnerstyp, Registertyp und Dokumenttyp ist notwendig. Die Zuordnung erfolgt immer über Objekttyp-ID und interne Namen.

Wählen Sie das XML-Format für die Protokollierung und den zugeordneten Dokumenttyp 'Protokoll_XML', können Sie das Stylesheet `adhoc.xml` aus dem Verzeichnis `\etc\templates\` des Datenverzeichnisses über einen Eintrag in der Datei `as.cfg` aus dem Verzeichnis `\etc\` des Datenverzeichnisses dem Dokumenttyp für die Ansicht im enaio® client zuordnen.

Ein enaio®-Update überschreibt weder die Konfigurationsdatei `AdHoc_Config.xml` aus dem Verzeichnis `\etc` des Datenverzeichnisses noch die Stylesheets `adhoc.xml`, `adhocFO.xml` oder deren anderssprachigen Äquivalente aus dem Verzeichnis `\etc\templates`. Dadurch bleiben individuelle Anpassungen auch nach einem Update erhalten. Ein Update ersetzt jedoch immer alle entsprechenden Default-Dateien (`AdHoc_Config_default.xml`, `adhoc_default.xml`, `adhocFO_default.xml` etc.), die ebenfalls in den Verzeichnissen `\etc` und `\etc\templates` des Datenverzeichnisses zu Verfügung stehen und Ihnen als stets aktuelle Vorlage für eigene Anpassungen dienen.

Informationen zu XML-Dokumenttypen finden Sie im Handbuch enaio® administrator.

Standard Ad-hoc-Workflow - Ablauf

Der Standard Ad-hoc-Workflow bietet das Modell für Vorgänge, in die sich die Benutzer schnell und einfach einarbeiten können, die aber zugleich komplex genug sind, um einen großen Teil täglich wiederkehrender Arbeitsprozesse zu organisieren.

Der Standard Ad-hoc-Workflow besteht aus einem Bereich mit sieben aufeinander abgestimmten Aktivitäten, die - im Unterschied zum strukturierten Workflow - nicht durch vorgegebene Transitionswege verbunden sind, sondern aus denen erst im Vorgang ein flexibler Ablauf durch die Benutzer erstellt wird. Mit diesen Aktivitäten werden drei unterschiedliche Prozesse ausgeführt - Abstimmungsprozesse, Freigabeprozesse und Informationsprozesse. Diese Prozesse können beliebig kombiniert, wie auch mehrfach ausgeführt werden. Beendet ist ein Ablauf, sobald in der Laufliste keine Aktivitäten mehr vorgesehen sind.

Die Akte des Standard Ad-hoc-Workflows kann von jedem Teilnehmer in jedem Schritt bearbeitet werden. Auf der Registerkarte **Protokollierung** ist in jedem Schritt der technische Ablauf nachvollziehbar. Jeder Schritt bietet einen Bereich, in dem der Benutzer Informationen für eine Notizhistorie eingeben kann. Diese ist ebenfalls für jeden Benutzer in jedem Schritt auf der entsprechenden Registerkarte einsehbar.

Der Standard Ad-hoc-Workflow kann sowohl im enaio® client wie auch im enaio® webclient ausgeführt werden. Alle Schritte liegen in Deutsch, Englisch und Französisch vor.

Im Folgenden werden die einzelnen Aktivitäten mit ihren Beziehungen zu folgenden und vorangehenden Aktivitäten beschrieben.

Eine allgemeine Beschreibung des Workflows, der Workflow-Akte und der Lauflisten finden Sie im Handbuch 'enaio® client'.

Initialisierung

Ein Ad-hoc-Vorgang wird aus dem enaio® client über den Korb 'Startbare Vorgänge' gestartet. Derjenige, der einen Vorgang startet, wird zu dessen Initiator. Ist ein Vorgang beendet und gibt es in der Workflowakte noch Dokumente und Protokolle, die keinen Standort in enaio® haben, werden diese in die persönliche Ablage des Initiators gelegt.

Der erste Vorgangsschritt des gestarteten Vorgangs ist immer die 'Initialisierung'.

Die Workflowmaske des Vorgangsschritts hat auf der Registerkarte **Allgemein** drei bearbeitbare Bereiche:

§ **Betreff**

Das Feld **Betreff** ist ein Pflichtfeld, muss also ausgefüllt werden. Der Betreff kennzeichnet alle Schritte des Vorgangs im Eingangskorb und kann für die Verschlagwortung von Protokolldateien verwendet werden. Deshalb ist es sinnvoll, in das Feld **Betreff** eine aussagekräftige Beschreibung für den Vorgang einzutragen. Der Eintrag im Feld **Betreff** kann nur in einem weiteren Vorgangsschritt 'Initialisierung' geändert werden.

§ **Bemerkung**

Das Bemerkungsfeld wird in allen Schritten angezeigt, und kann ebenfalls nur in einem weiteren Vorgangsschritt 'Initialisierung' geändert werden. Ein Eintrag in diesem Feld wird in allen folgenden Schritten angezeigt und wird auch in die Notizhistorie übertragen.

§ **Abstimmung**

Über das Feld **Abstimmung** können Optionen für eine Abstimmung in einem Folgeschritt 'Abstimmung' über eine Tabelle angegeben werden. Abstimmungsoptionen können in einem Folgeschritt 'Überarbeiten' oder 'Initialisierung' geändert werden. Wurde bereits abgestimmt, werden bei Änderungen bereits vorliegende Abstimmungsergebnisse immer zurückgesetzt.

Die hier eingetragenen Abstimmungsoptionen werden im Schritt 'Abstimmung' in einer Liste angezeigt, aus der die Teilnehmer die gewünschte Option wählen. Automatisch wird die Abstimmungsoption 'keine Meinung' ergänzt.

Eine Abstimmung ist eine Multi-Instanz-Aktivität, kann also von mehreren Teilnehmern ausgeführt werden. Die Abstimmung wird automatisch beendet, wenn sie von allen Teilnehmern ausgeführt wurde oder wenn die Abstimmungen für eine Option den angegebenen prozentualen Wert überschritten haben.

Das Ergebnis einer Abstimmung wird in einem Folgeschritt 'Kenntnisnahme' und auf der Registerkarte **Protokollierung** angezeigt.

Das Feld **Lauflistenbemerkung** zeigt den Eintrag an, der in der Laufliste als Bemerkung für diese Aktivität angegeben wurde.

Aktivität	Aufgabe	Teilnehmer	Eskalation	fallig am	um			Bemerkung
-----------	---------	------------	------------	-----------	----	--	--	-----------

1. Abstimmung

Abstimmung	Abstimmung	WENZ,...	Mahnmail	06.03.2009	13:11:22	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Zurück"/>
------------	------------	----------	----------	------------	----------	--------------------------	--------------------------	---------------------------------------

Lauflistenbemerkungen sind optional. Im ersten Initialisierungsschritt ist dieses Feld noch leer.

Beim Weiterleiten erhält der Benutzer einen Hinweis, falls keine Objekte in die Akte gelegt wurden.

Alle Folgeschritte, außer einer weiteren Initialisierung, zeigen auf der Registerkarte **Notizhistorie** in den Feldern **Initiator** und **Startdatum**, welcher Teilnehmer an welchem Datum den Prozess initialisiert hat.

Die Notizhistorie auf der entsprechenden Registerkarte ist im ersten Schritt noch leer.

Überarbeiten

Im Schritt 'Überarbeiten' finden Sie in den Feldern **Betreff** und **Bemerkung** die entsprechenden Einträge aus der Initialisierung und im Feld **Lauflistenbemerkung** den Eintrag, der in der Laufliste als Bemerkung für diese Aktivität angegeben wurde. Diese drei Felder sind schreibgeschützt.

Einträge im Feld **Notiz** werden in die Notizhistorie übertragen und sind dort in allen Folgeschritten sichtbar.

Auf der Registerkarte **Abstimmungsoptionen überarbeiten** können, wie bei der Initialisierung, in einer Tabelle Abstimmungsoptionen angegeben oder überarbeitet werden. Wurde bereits in einem vorherigen Schritt abgestimmt, werden diese Abstimmungsdaten gelöscht und sind nur noch zugänglich, wenn sie in einem vorangehenden Protokollierungsschritt gespeichert wurden.

Mitarbeit / Rückfrage

Im Schritt 'Mitarbeit / Rückfrage' finden Sie ebenfalls in den schreibgeschützten Feldern **Betreff** und **Bemerkung** die entsprechenden Einträge aus der Initialisierung und im Feld **Lauflistenbemerkung** den Eintrag, der in der Laufliste als Bemerkung für diese Aktivität angegeben wurde.

Einträge im Feld **Antwort** werden in die Notizhistorie übertragen und sind dort in allen Folgeschritten sichtbar.

Abstimmung

Die Maske des Schritts 'Abstimmung' stellt die Abstimmungsoptionen, die bei der Initialisierung oder Überarbeitung über eine Tabelle angegeben werden, als Auswahlliste dar.

Optionen

Optionen
1. investieren
2. nicht investieren
3. später investieren

Bearbeitung

Abstimmung

Stimmen Sie hier ab. Bitte beachten Sie für ihre Abstimmung auch die Bemerkung des Initiators, sowie die Notizhistorie der anderen Workflow-Teilnehmer.

Notiz

Geben Sie hier eine Notiz ein. Sie ist für nachfolgende Lauflistenteilnehmer sichtbar.

<Keine Meinung>

investieren

nicht investieren

später investieren

Zusätzlich zu den eingegeben Abstimmungsoptionen hat der Teilnehmer immer die Möglichkeit, die Option 'Keine Meinung' zu wählen. Das Feld **Abstimmung** ist ein Pflichtfeld und muss ausgefüllt werden.

Abstimmungen sind Multi-Instanz-Aktivitäten und können von mehreren Teilnehmern ausgeführt werden.

Eine Abstimmung wird automatisch beendet, wenn sie von allen Teilnehmern ausgeführt wurde oder wenn die Stimmen für eine Option den angegebenen prozentualen Wert überschritten haben.

Das Ergebnis einer Abstimmung wird im Schritt 'Kenntnisnahme' auf der Registerkarte **Abstimmungsergebnis** als Tabelle angezeigt. Über die Registerkarte 'Protokollierung' kann eingesehen werden, wie die einzelnen Teilnehmer abgestimmt haben. Der Inhalt dieser Protokollierungstabelle kann über das Kontextmenü oder **Strg+E** in die Zwischenablage kopiert werden.

Die Anzahl der 'Keine Meinung'-Stimmen wird in der Tabelle nicht angezeigt.

Freigabe erteilen

Die Felder **Betreff** und **Bemerkung** enthalten die entsprechenden Einträge aus der Initialisierung, das Feld **Lauflistenbemerkung** den Eintrag, der in der Laufliste als Bemerkung für diese Aktivität angegeben wurde.

Im Schritt 'Freigabe erteilen' kann eine Option aus dem Bereich **Entscheidung** markiert werden – **Offen**, **Keine Freigabe** oder **Freigabe erteilt**.

Die gewählte Option wird protokolliert und kann in allen Folgeschritten auf der Registerkarte **Protokollierung** eingesehen werden.

Folgt ein weiterer Schritt 'Freigabe erteilen', wird dort die zuvor gewählte Option voreingestellt.

Einträge im Feld **Notiz** werden in die Notizhistorie übertragen und sind dort in allen Folgeschritten sichtbar.

Kenntnisnahme

Auch im Schritt 'Kenntnisnahme' finden Sie in den schreibeschützten Feldern **Betreff** und **Bemerkung** die entsprechenden Einträge aus der Initialisierung und im Feld **Lauflistenbemerkung** den Eintrag, der in der Laufliste als Bemerkung für diese Aktivität angegeben wurde.

Einträge im Feld **Notiz** werden in die Notizhistorie übertragen und sind dort in allen Folgeschritten sichtbar.

Auf der Registerkarte **Abstimmungsergebnis** finden Sie, falls eine Abstimmung voranging, das Ergebnis als Tabelle. Über die Registerkarte 'Protokollierung' kann eingesehen werden, wie die einzelnen Teilnehmer abgestimmt haben.

Informationen zu Freigaben finden Sie auf der Registerkarte **Protokollierung**.

Protokoll

Der Schritt 'Protokoll' hat keine Teilnehmer. Er wird also von keinem Teilnehmer gestartet, sondern automatisch durchlaufen. Wenn der Schritt in die Laufliste aufgenommen ist, wird eine Protokolldatei mit den Daten aller bis zu diesem Zeitpunkt durchgeführter Schritte erzeugt und in die Workflow-Akte gelegt. Diese Protokolldatei kann in jedem folgenden Schritt eingesehen werden.

Der Schritt 'Protokoll' kann am Ende eines Vorgangs stehen. Folgt kein weiterer Schritt, wird eine Protokolldatei, der nicht bereits über eine Konfiguration ein Standort und Dokumenttyp zugeordnet ist, wie alle Dokumente ohne Standort aus der Workflow-Akte in die Ablage des Initiators gelegt.

Neben dieser Protokollierung wird automatisch eine technische Ablaufprotokollierung auf der Registerkarte **Protokollierung** geführt. Diese Protokollierung beinhaltet beispielsweise Freigabeinformationen, nicht aber die Notizen aus den einzelnen Schritten. Diese können über die Notizhistorie eingesehen werden.

Der Inhalt der Registerkarte **Protokollierung** kann über das Kontextmenü oder **Strg+E** in die Zwischenablage kopiert werden.

Eskalation für Standard Ad-hoc-Aktivitäten

In einer Laufliste kann einer Aktivität eine im Modell eingerichtete Eskalation zugeordnet werden. Für alle Aktivitäten des Standard Ad-hoc-Workflows sind die folgenden Eskalationen möglich:

§ Erinnerungs-E-Mail

Ist ein Vorgangsschritt im Eingangskorb eines Benutzers eingetroffen und nicht innerhalb eines Zeitraums weitergeleitet, erhält er eine E-Mail mit entsprechendem Hinweis. Über die Felder **fällig am** und **um** wird der Zeitraum angegeben.

Vorausgesetzt ist, dass die E-Mail-Adresse in der Workflow-Benutzerverwaltung angegeben ist.

§ Automatische Weiterleitung

Der Vorgangsschritt wird automatisch an die Folgeaktivität weitergeleitet, sobald das Ende eines Zeitraums erreicht ist. Weitergeleitet werden sowohl personalisierte wie auch nicht personalisierte Schritte. Der voreingestellte Zeitraum von zwei Tagen kann über die Felder **fällig am** und **um** geändert werden.

§ Zuweisen an Stellvertreter

Ist ein Vorgangsschritt im Eingangskorb eines Benutzers eingetroffen und nicht innerhalb eines Zeitraums weitergeleitet, wird der Vorgangsschritt automatisch dem Stellvertreter zugewiesen. Über die Felder **fällig am** und **um** wird der Zeitraum angegeben.

Vorausgesetzt ist, dass ein Stellvertreter in der Workflow-Benutzerverwaltung eingerichtet ist.

Taskflow

Der Standard Ad-hoc-Workflow kann auch für einfache Zuweisungen von Aufgaben eingesetzt werden, ist aber dazu auf Grund der komplexen Möglichkeiten der Lauflistenverwaltung für viele Benutzer zu sperrig.

Der Taskflow dagegen besteht nur aus einem Schritt, der schnell ausgeführt und an einen Empfänger weitergeleitet werden kann. Der Empfänger bearbeitet diesen Schritt. Durch das Weiterleiten gelangt dann der Schritt wieder zum Absender. Der Empfänger kann weitere Empfänger mit einbeziehen und wird dann in diesem Kontext zum Absender, an den der Schritt wieder zurück geleitet wird.

Installation und Konfiguration

Der Taskflow ist zusammen mit dem Standard Ad-hoc-Workflow Teil der Datei `osdefaultmodels.xml` aus dem Verzeichnis `\clients\admin\` und wird über den Import dieser Datei installiert (vgl. 'Installation').

Für die Konfiguration müssen ebenfalls wie beim Standard Ad-hoc-Workflow Personen über das Organisationsmodell der Rolle 'Adhoc' zugeordnet werden. Jede Person mit dieser Rolle wird gleichberechtigter Teilnehmer des Taskflows.

Taskflow - Ablauf

Der Taskflow besteht aus einem Schritt.

Innerhalb dieses Schrittes kann die Aufgabe angegeben und über Optionen spezifiziert werden: **zur Bearbeitung**, **zur Prüfung**, **zur Freigabe** und **zur Kenntnisnahme**. Es kann ein Fälligkeitsdatum angegeben werden. Das Fälligkeitsdatum entspricht einer Mahnfrist im Modell.

Als Empfänger kann jeder Benutzer ausgewählt werden, der Berechtigter des Modells ist. Der Empfänger muss über das AddOn ausgewählt werden.

Der Ablauf kann strukturiert werden, indem der Schritt jeweils einem Empfänger zur Prüfung, Freigabe oder Kenntnisnahme vorgelegt wird und gegebenenfalls vom Empfänger nochmal an einen weiteren Empfänger weitergeleitet werden kann.

Jeder Schritt kann dabei vom Empfänger kommentiert werden. Die Kommentare werden im Verlaufsbereich automatisch protokolliert.

Jeder Schritt wird automatisch im Ablauf wieder an den Absender und letztlich wieder an den Initiator zurück geleitet, der so immer über den Abschluss des Taskflows informiert wird.

Betrieb und Administration

Lizenzierung und Berechtigungen

enaio® Workflow ist Bestandteil von enaio® und setzt die Installation von enaio® client voraus. Die einzelnen Module verwenden unterschiedliche Lizenzschlüssel, die nachfolgend aufgelistet sind.

Lizenzschlüssel	Bedeutung	Geprüft durch
ASW	Starten der Workflow-Engine	Workflow-Engine im Applikationsserver (oxjobwfm.dll)
WFA	Starten des Workflowadministrators	Workflowadministrator (axwfadm.exe)
WFE	Workfloweditor: Organisationsstruktur	Workfloweditor (axwfedit.exe)
WFG	Workfloweditor: Masken- und Prozessdesign Folgende Aktionen sind ohne 'WFG' möglich: § Importieren, § Anlegen, Bearbeiten und Löschen von Projekten und Familien, § Umbenennen, Löschen und Ändern des Status von Modellen, aber nicht das Sperren.	Workfloweditor (axwfedit.exe)
MWC	Workflow-Client	enaio® client (ax.exe)
AVG	Automatischer Start von Prozessen	Workflow-Startkomponente in enaio® capture (axwfstart.exe) und als separates Programm (axwfscrip.exe)
WAU	Starten der Workflow-Autostation	Workflow-Autostation
WAH	Ad-hoc-Workflow	Ad-hoc-Workflow

Berechtigungen

Da der Einsatz eines Workflowmanagementsystems mitunter unternehmenskritische Bereiche betrifft, ist es notwendig, die Administration und das Prozessdesign vor unbefugtem Zugriff zu schützen.

Aus diesem Grund müssen Anwender, die den Workflow-Administrator oder den Workflow-Editor benutzen, über entsprechende Systemrollen verfügen.

Die folgenden Systemrollen können für den Workflow vergeben werden:

- ☐ OSIECM - WF-Admin: Starten
- ☒ OSIECM - WF-Editor: Starten
 - ☐ OSIECM - WF-Editor: Organisation bearbeiten
 - ☐ OSIECM - WF-Editor: Benutzer an-/abwesend melden
 - ☐ OSIECM - WF-Editor: Modell erstellen
- ☐ OSIECM - WF-Prozesse: Per Import starten
- ☐ OSIECM - WF-Simulation: Starten
- ☐ OSIECM - WF-Script: Starten

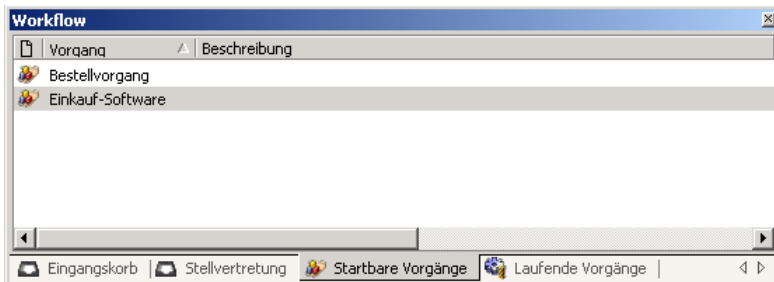
Workflowkomponenten im enaio® client

Die wichtigsten Anwendungen zur Nutzung des Workflows sind der enaio® client und enaio® webclient. In diesen Clients erfolgen alle interaktiven Eingaben zur Bearbeitung der im Workflowprozess festgelegten Aktivitäten.

enaio® client und enaio® webclient sind analog aufgebaut.

Workflowbereich

Der Workflowbereich ist im enaio® client ähnlich wie die Abonnement- oder Wiedervorlagefenster realisiert. Das Fenster kann durch den Anwender beliebig andockt oder verschoben werden, wobei die letzten Einstellungen gespeichert werden.



Der Workflowbereich ist in verschiedene Körbe unterteilt, die vom Anwender selbst konfiguriert werden können. In den Körben werden Prozesse bzw. Aktivitäten oder Modelle verwaltet.

Die wichtigsten Körbe sind der Eingangskorb und die Liste der Startbaren Vorgänge.

Eingangskorb

Im Eingangskorb erhält der Anwender Benachrichtigungen über anstehende Aktivitäten. Wurde also von der Workflow-Engine festgestellt, dass eine Aktivität durch Anwender einer bestimmten Rolle bearbeitet werden soll, so wird die Rolle nach den betreffenden Benutzern aufgelöst. Alle in der Rolle enthaltenen Benutzer erhalten dann eine Benachrichtigung über das Vorliegen der neuen Aktivität und zeigen diese im Eingangskorb an.



Durch entsprechende Symbole wird im Eingangskorb dargestellt, ob eine Aktivität personalisiert wurde oder ob eine vorgegebene Mahnfrist abgelaufen ist.

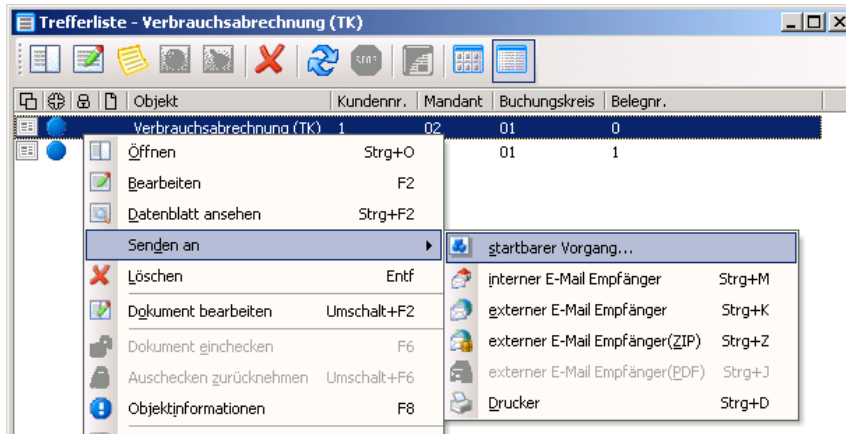
Zusätzlich kann über den enaio® enterprise-manager eine pauschale Benachrichtigung per E-Mail eingerichtet werden und für jede einzelne Aktivität konfiguriert werden. Dabei stehen wie bei den Mahnfristen Platzhalter zur Verfügung.

Personalisierung

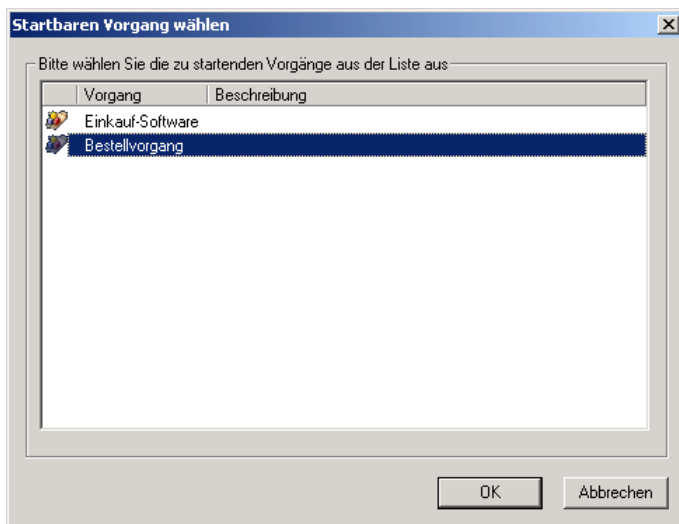
Nachdem der erste Anwender die Aktivität durch einen Doppelklick im Eingangskorb angenommen hat, wird diese personalisiert. Damit verschwindet die Aktivität aus den Eingangskörben der anderen Rollenmitglieder und im Eingangskorb des Bearbeiters wird die Aktivität durch das Symbol  markiert. Gewöhnlich erscheint nun die im Modell festgelegte Aktivitätenmaske und der Anwender kann die vorgesehenen Aktionen ausführen.

Startbare Vorgänge

In der Liste der startbaren Vorgänge sind alle aktivierten Modelle aufgelistet, die durch den Anwender, d.h. durch eine der Rollen, in der er sich befindet, gestartet werden können. Der Start erfolgt gewöhnlich über einen Doppelklick auf einen entsprechenden Eintrag. Ist im Modell hinterlegt, dass der Prozess mit einem Dokument oder sonstigen DMS-Objekt gestartet werden kann oder muss, so kann der Prozess durch Ziehen des Objektes auf den startbaren Vorgang initiiert werden. Alternativ dazu steht im **Senden an...** Menü des Clients der Eintrag **Startbarer Vorgang**.



Danach erscheint ein Dialog, in dem alle „Startbaren Vorgänge“ aufgelistet sind.



Ist ein Vorgang gestartet, wird der entsprechende Vorgangsschritt im Eingangskorb aller Teilnehmer eingetragen.

Über eine Sofortbenachrichtigung kann dieser Vorgangsschritt automatisch geöffnet werden, wenn derjenige, der den Vorgang startet ebenfalls Teilnehmer des ersten Vorgangsschritts ist. Die Sofortbenachrichtigung wird über den Benachrichtigungs-Dialog der Aktivität eingeschaltet.

Zusätzlich müssen Sie über die Konfigurationsdatei `as.cfg` aus dem Verzeichnis `\etc` des Datenverzeichnisses einen Zeitraum angeben, innerhalb dessen die Sofortbenachrichtigung erfolgen soll:

[WORKFLOW]

`AUTOSTARTWAITTIME=Zeitraum in Sekunden`

Dieser Zeitraum sollten Sie so bemessen, dass der Server in der Lage ist, alle Daten zu verarbeiten und der Benutzer in der Lage ist, die Abfolge und Zuordnung der Schritte zu realisieren. In der Regel ist ein Zeitraum von 20 Sekunden angemessen. Nach Ablauf dieses Zeitraums wird ein Vorgangsschritt nicht mehr automatisch geöffnet.

Die Sofortbenachrichtigung tritt ebenfalls ein, wenn ein Benutzer einen Vorgangsschritt weiterleitet und ebenfalls Teilnehmer des folgenden Vorgangsschritts ist.

Aktivitätenmaske

Die Bearbeitung der Aktivitäten erfolgt durch Aktivitätenmasken, die durch das Prozessdesign vorgegeben sind. An den einzelnen Schritten soll nun der Anwender die vorgesehenen Daten eingeben.

Die Funktion der einzelnen Dialogelemente entsprechen denen auf Verschlagwortungsmasken im enaio® client. Benutzer schließen eine Aktivität ab, indem Sie sie **Weiterleiten**. Auf der Maske kann ein Feld mit dem Workflow-Empfänger-AddOn verbunden werden. Ist dieses mit einer folgenden Aktivität verbunden, dann werden die darüber ausgewählten Personen oder Rollen zu zusätzlichen Teilnehmern dieser Aktivität.

Wird eine Aktivität geschlossen, können Änderungen übernommen oder verworfen werden.

Über eine Sofortbenachrichtigungsfunktion kann nach dem Weiterleiten der folgende Vorgangsschritt automatisch geöffnet werden, wenn derjenige, der den Vorgang weiterleitet ebenfalls Teilnehmer des nächsten Vorgangsschritts ist.

Laufende Vorgänge

Vorgänge bleiben für die Benutzer, die diese Vorgänge gestartet haben, im enaio® client sichtbar. Im Workflowbereich werden auf der Registerkarte **Laufende Vorgänge** die aktuellen Vorgangsschritte und, falls diese personalisiert sind, die Benutzer angezeigt. Zu einem Vorgangsschritt kann die Akte eingesehen werden. Vorgangsschritte können anderen Benutzern zugewiesen werden.

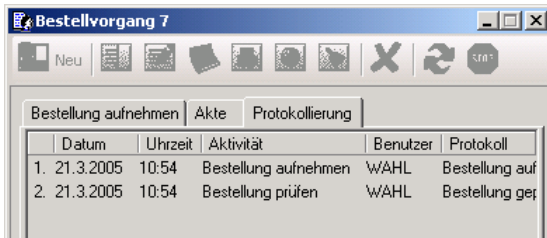
Sind 'Verantwortliche' für ein Modell angegeben, können diese ebenfalls auf der Registerkarte die Vorgangsschritte einsehen.

Vorgangsschritt	Vorgang	Benutzer	gestartet am	fällig am
Bestellung prüfen	Bestellvorgang 7		21.03.2005 10:54:14	
Bestellung aufnehmen	Bestellvorgang 8		21.03.2005 10:55:49	23.03.2005 10:55:50
Bestellung aufnehmen	Bestellvorgang 9	Wahl	21.03.2005 10:55:51	23.03.2005 10:55:53

Über das Kontextmenü kann auf Prozessdetails zugegriffen werden.

Protokollierung

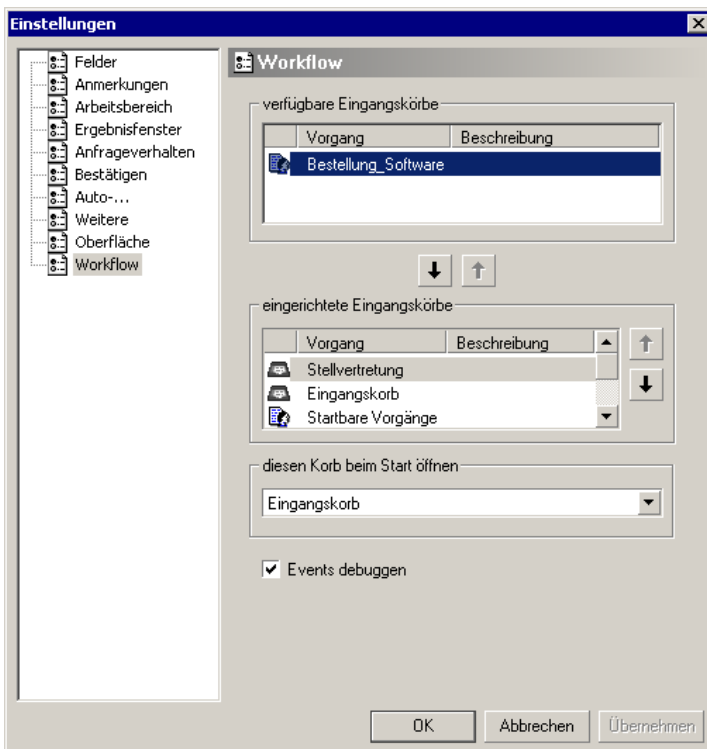
Wenn das Protokoll für den Workflow eingerichtet ist, dann erscheint auf den Aktivitätenmasken eine zweite Seite mit der Protokollierung. Die jeweils dargestellten Spalten hängen von den Einstellungen des Protokolls ab und können für jedes Modell differenziert angezeigt werden. Die Einstellungsmöglichkeiten für das integrierte Prozessprotokoll sind im Kapitel *Prozessdesign* beschrieben.



Datum	Uhrzeit	Aktivität	Benutzer	Protokoll
1. 21.3.2005	10:54	Bestellung aufnehmen	WAHL	Bestellung auf
2. 21.3.2005	10:54	Bestellung prüfen	WAHL	Bestellung ges

Einstellungen

Der Anwender kann in seinem Einstellungsdialog das Aussehen und Verhalten der Workflowkörbe selbst beeinflussen. Dazu steht ein Bereich zur Verfügung, den der Anwender über den üblichen Einstellungsdialog im enaio® client öffnet.



Über die Verwaltung der Körbe können die im Workflowbereich anzuzeigenden Körbe verwaltet werden. Standardmäßig sind folgende Körbe aktiv:

- Eingangskorb** der Anwender erhält hierüber die neuen Aktivitäten, die den Rollen, die er innehat, zugeteilt sind.
- Startbare Vorgänge** Prozessmodelle, die von den Rollen des Anwenders gestartet werden können.
- Stellvertretung** Eingangskorb für Aktivitäten, die der Anwender als Stellvertreter für andere Benutzer erhält.
- Laufende Vorgänge** Alle aktuellen Aktivitäten von Vorgängen, die ein Benutzer gestartet hat, sind hier aufgelistet.

Darüber hinaus kann der Anwender für die für ihn gültigen Prozessmodelle eigene Körbe einrichten, in denen dann nur Prozesse des jeweiligen Modells gesammelt werden.

Stellvertretung

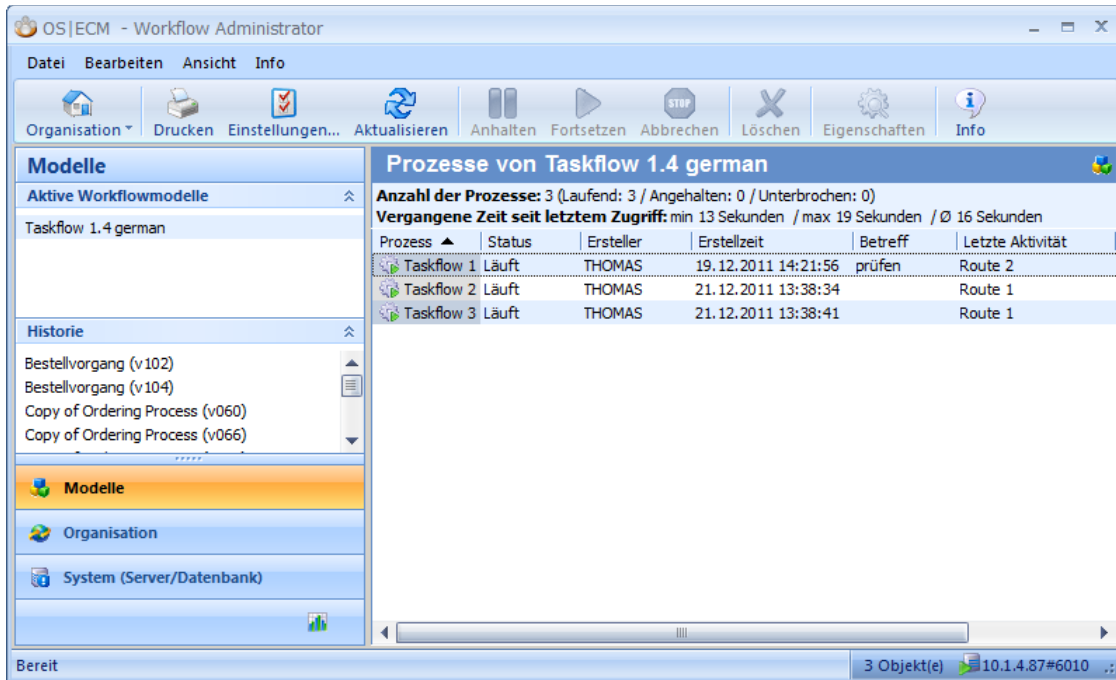
Die Stellvertretung arbeitet im Zusammenhang mit der Anwesenheit der Benutzer. Der Anwender kann sich-etwa im Falle von Urlaub-selbst als abwesend eintragen. Sind alle Anwender einer Rolle, für die eine Aktivität vorgesehen ist, als abwesend eingetragen, so wird die entsprechende Aktivität den Stellvertretern zugestellt. Dies gilt nur für nicht personalisierte Aktivitäten.

Um den Status der Anwesenheit zu setzen, kann im Workflowbereich des enaio® clients der gewünschte Eintrag im Kontextmenü ausgewählt werden.

Die Stellvertreter selbst werden im Organisationexplorer benannt.

Workflow-Administrator

Der enaio®-Workflow Administrator ist ein Tool zur Überwachung und Steuerung bereits laufender Prozesse. In ihm werden alle gestarteten Prozesse in verschiedenen Gruppierungen dargestellt. Aus der Prozessansicht kann Einblick in die Details, insbesondere in die Belegung der Workflowvariablen genommen werden. Des Weiteren können laufende Prozesse angehalten, fortgesetzt und abgebrochen werden. Zusätzlich können hier Informationen über den Status der Datenbank und der Workflow-Engine abgerufen werden.



Im Workflow Administrator können zur besseren Übersichtlichkeit über verschiedene Bereiche Sichten auf die Prozesse gewählt werden. Ein weiterer Bereich bietet Einblick in den Zustand der Datenbank und der Workflow-Engine, über den Bereich Statistik werden Reports erstellt.

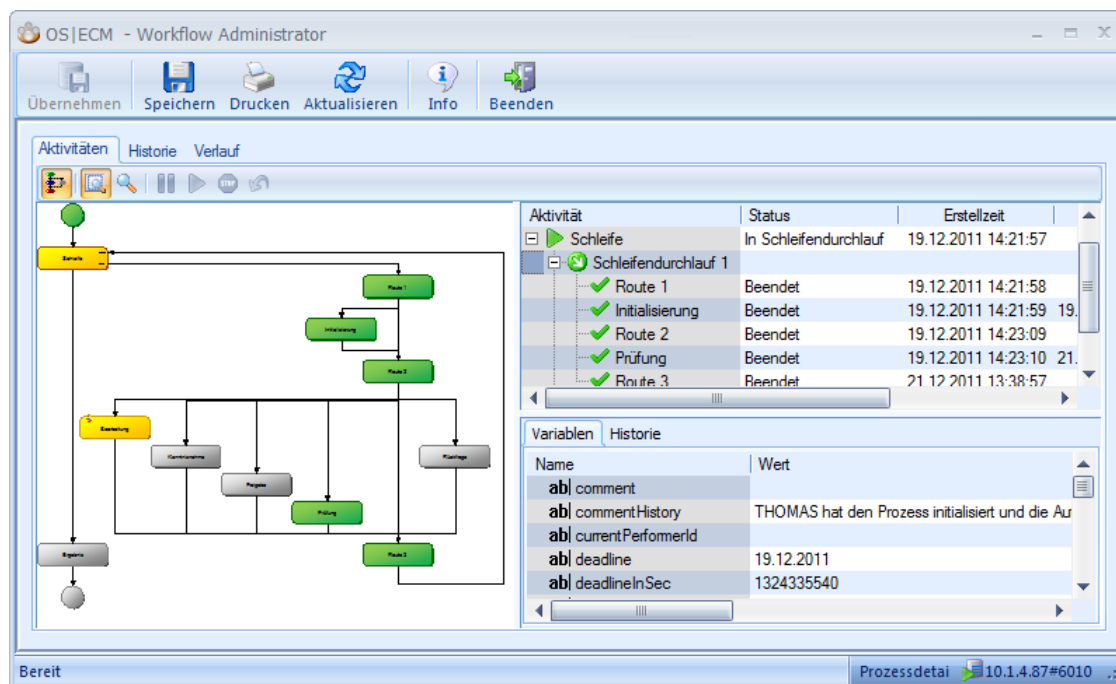
Modelle	Im Arbeitsbereich werden alle Vorgangsmodelle angezeigt, die aktiv oder in Benutzung sind.
Organisation Rollen	Im Arbeitsbereich werden alle Rollen angezeigt, zu denen noch Aktivitäten in laufenden Prozessen ausstehen. Sind Aktivitäten mehrere Rollen zugeordnet oder sind mehrere Aktivitäten eines Prozesses aktiv, so werden die Zuordnungen zu allen betroffenen Rollen dargestellt.
Organisation Benutzer	Im Arbeitsbereich werden Benutzer dargestellt, die sich in Rollen befinden, zu denen ausstehende Aktivitäten vorhanden sind. Im Listenfenster erscheinen dann alle zu Benutzern vorhandene Aktivitäten und deren zugehörige Prozesse.
System (Server/ Datenbank)	Es lassen sich Informationen über Sperren auf Tabellen und Prozesse sowie den Status des Servers (die Workerqueue) abrufen.
Statistik	Es können Konfigurationen erstellt werden, über die Übersichts-Reports oder Detail-Reports erstellt werden.

Folgende Daten werden in der Modell-Ansicht angezeigt:

Spalte	Bedeutung
Statussymbol	Durch das Symbol wird der Status des laufenden Prozesses symbolisiert.
Prozess	Name des Vorgangsmodells
Status	Status des Prozesses
Ersteller	Anwender, der den Prozess gestartet hat
Erstellzeit	Zeitpunkt der Erstellung des Prozesses
Betreff	Für das Modell festgelegte Betreff-Zeile
Letzte Aktivität	Zuletzt absolvierte Aktivität des Prozesses

Zeitpunkt	Zeitpunkt der Beendigung der letzten Aktivität
-----------	--

Über das Kontextmenü eines Prozesses kann dieser Prozess angehalten, fortgesetzt und gelöscht werden. Es lassen sich weitere Eigenschaften in einer Detailansicht abrufen. Diese Detailansicht öffnen Sie über den Button **Eigenschaft**, das Kontextmenü, die Funktionstaste **F2** oder durch Doppelklick auf einen Prozess:



Diese Detailansicht bietet wiederum viele verschiedene Sichten auf den Einzelprozess und ebenfalls die Möglichkeiten, den Prozess anzuhalten, fortzusetzen und Variablen zu ändern.

In der Historien-Ansicht des Modellbereichs werden alle Modelle mit ihrer automatisch vergebenen Versionsnummer angezeigt. Bei jeder Änderung an einem Modell wird die Versionsnummer hochgesetzt. Die Historien-Ansicht enthält die bereits beendeten Prozesse. Bis auf die Spalte **Letzte Aktivität** werden die gleichen Spalten wie in der Modell-Ansicht gezeigt.









Werden die Aktivitäten nach Rollen oder Benutzern gruppiert, so werden Aktivitäten und deren zugehörige Prozesse aufgelistet. In diesem Fall kann es vorkommen, dass für einen laufenden Prozess mehrere Aktivitäten dargestellt werden, und zwar genau dann, wenn die mehrere parallele Zweige im Prozess vorgesehen sind.

Folgende Daten werden in der Rollen- und Benutzer-Ansicht angezeigt:

Spalte	Bedeutung
Statussymbol	Durch das Symbol wird der Status des laufenden Prozesses symbolisiert.
Prozess	Bezeichnung des Prozessmodells
Status	Status des Prozesses
Ersteller	Name des Anwenders, der den Prozess erstellt hat
Erstellzeit	Zeitpunkt der Erstellung des Prozesses
Aktivität	Bezeichnung der Aktivität
Aktivitätenstatus	Status der Aktivität
Zeitpunkt	Zeitpunkt der Erstellung der Aktivität
Personalisiert	Zeitpunkt der Personalisierung
Durch	In der Rollen-Ansicht der Name des Anwenders, der die Aktivität personalisiert hat.
Mahnfrist	Anzeige einer vorgegebenen Mahnfrist für die Aktivität

Prozess- und Aktivitätenstatus

Prozesse können unterschiedliche Statuswerte annehmen, die in der folgenden Tabelle aufgelistet sind:

Symbol	Prozessstatus
	Initialisierung
	Läuft
	Läuft, mindestens eine Aktivität ist angehalten
	Angehalten
	Aktiv
	Abgebrochen
	Beendet
	Unterbrochen

Aktivitäten können unterschiedliche Statuswerte annehmen, die in der folgenden Tabelle aufgelistet sind:

Symbol	Aktivitätenstatus
	Angelegt
	Gestartet
	Ausführung Skript
	Überprüfung Schleifenbedingung
	In Schleifendurchlauf
	Arbeitsschritt angelegt
	Arbeitsschritt personalisiert
	Warten aufgrund einer Sperrfrist
	Warten aufgrund eines Subflows
	Ausgeführt
	Weitergeleitet
	Abgeschlossen
	Angehalten
	Beendet
	Fehler

Prozess-Eigenschaften

Abhängig von der gewählten Sicht im Arbeitsbereich werden die zugeordneten Prozesse aufgelistet. Für jeden Prozess kann über das Kontextmenü der Eigenschaftsdialog geöffnet werden.

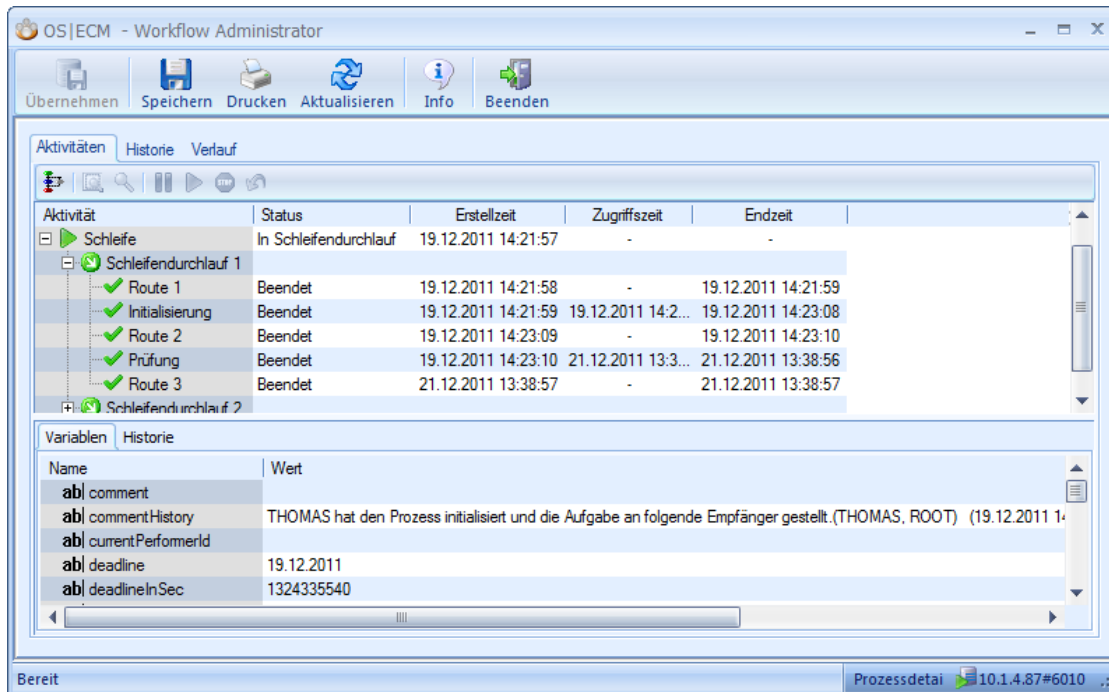
Die Daten werden über drei Registerkarten dargestellt:

§ Aktivitäten

Alle Aktivitäten, die bereits durchlaufen wurden und die aktuelle Aktivität werden angezeigt. Die Werte der Variablen einer markierten Aktivität werden angezeigt und können editiert werden.

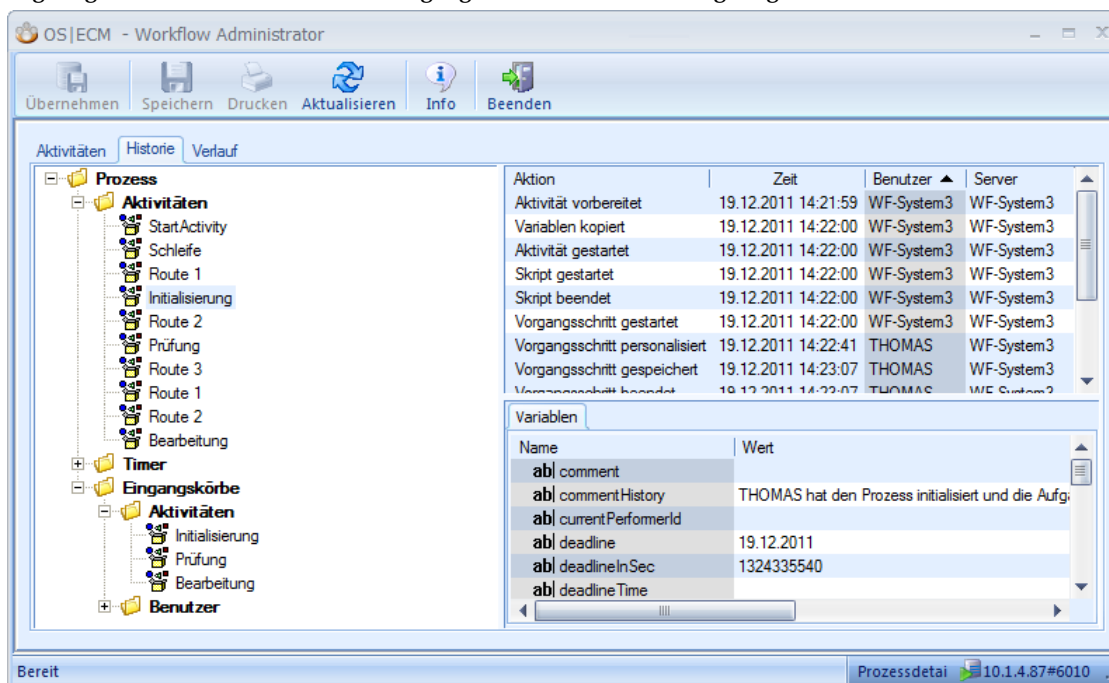
Über das Kontextmenü einer aktuellen Aktivität kann diese Aktivität einem Benutzer oder einer Rolle zugewiesen werden. Ist die Aktivität personalisiert, wird dazu die Personalisierung aufgehoben. Außerdem können Aktivitäten angehalten und fortgesetzt werden. Für angehaltene Prozesse ist es möglich den Prozess zu einer Aktivität zurückzusetzen.

Für abgeschlossene Prozesse wird diese Registerkarte über die Historien-Ansicht nicht angezeigt.



§ Historie

Auf der Registerkarte werden alle Detail-Aktionen zu Aktivitäten angezeigt und der Wert der Variablen. Ebenfalls angezeigt werden kann, in welchen Eingangskörben Aktivitäten angezeigt wurden.



§ Verlauf

Über diese Registerkarte wird ein Verlaufsprotokoll angezeigt. Das Protokoll kann gespeichert oder ausgedruckt werden.

Über den Reiter **Aktivitäten** ist es für angehaltene Prozesse möglich, den Prozess zu einer Aktivität zurückzusetzen. Alle nachfolgenden Aktivitäten würden gelöscht und Variablenwerte wieder entsprechend zurückgesetzt. Laufende Fristen oder die Workflowakte werden dabei nicht berücksichtigt.

Es werden, abhängig davon, welchen Status die Aktivität bereits erreicht hat, bis zu drei mögliche Aufsatzpunkte angeboten:

Aufsatzpunkt	Erklärung
--------------	-----------

Aktivität erstellen	Die Aktivität wird zu dem Zeitpunkt vor der Variablenübernahme zurückgesetzt.
EndActivity-Event ausführen	Die Aktivität wird zu dem Zeitpunkt vor dem Ausführen des EndActivity-Events zurückgesetzt.
Aktivität weiterleiten	Die Aktivität wird zu dem Zeitpunkt zurückgesetzt, an dem ihre Ausführung beendet wurde.

Organisation

Im Bereich **Organisation** erhält man einen Überblick über die laufenden Arbeitsschritte, unterteilt nach Rollen und Benutzer. Über das Kontextmenü der Einträge ist es möglich, Arbeitsschritte anzuhalten / fortzuführen und Arbeitsschritte Benutzern oder Rollen zuzuordnen. Je nach Aktivität und Status sind diese Aktionen auch für mehrere markierte Arbeitsschritte auf einmal möglich.

Prozess	Status	Ersteller	Erstellzeit	Betreff	Aktivität	Aktivitätenstatus
Taskflow 1	Läuft	THOMAS	19.12.2011 14:21:56	prüfen	Prüfung	Angehalten
Taskflow 2	Läuft	THOMAS	21.12.2011 13:38:34		Initialisierung	Vorgangsschritt ar
Taskflow 3	Läuft	THOMAS	21.12.2011 13:38:41		Initialisierung	Vorgangsschritt ar
Taskflow 4	Läuft	THOMAS	04.01.2012 15:13:05		Initialisierung	Vorgangsschritt pe

Server- und Datenbank-Eigenschaften

Im Bereich **Server/Datenbank** erhält man einen Überblick über die Sperren auf Tabellen und Prozessen sowie über den aktuellen Zustand der Workerqueue (serverinterner Mechanismus zur Bearbeitung von Prozessen) der Workflow-Engine. Sperren lassen sich über das Kontextmenü aufheben.

Tabelle	Status	Sperzeitpunkt	Server ID	Thread ID
osorganization	0	-	0	0
osspooler	0	-	0	0
oswfactivities	0	-	0	0
oswfprocesses	0	-	0	0
oswfprocesses	0	-	0	0
oswfworkers	0	-	0	0
oswfworkerqueue	0	-	0	0

Statistik-Reports

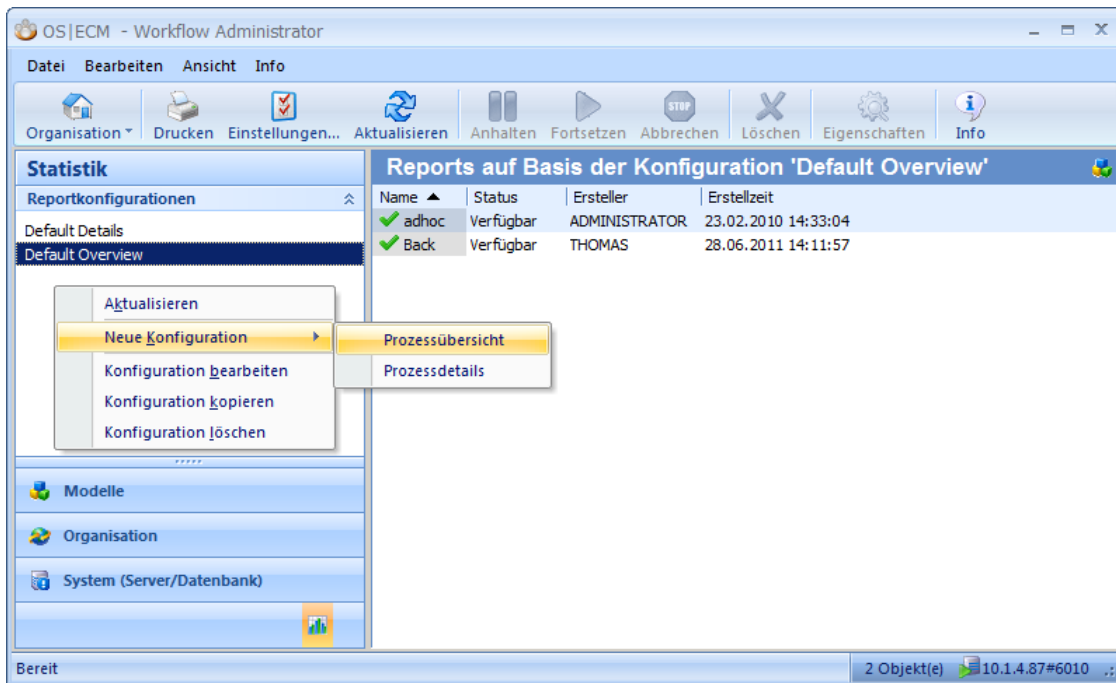
Im Bereich **Statistik** legen Sie Konfigurationen für Reports an. Über Reports dokumentieren sie, je nach Konfiguration, den Zustand eines Prozesses im Detail oder den Zustand aller Prozesse im Überblick zu einem festgelegten Zeitpunkt.

Auf der Grundlage einer Detail- oder Überblick-Konfiguration kann ein Report über den Bereich **Statistik** oder aus dem enaio® client erstellt und eingesehen werden. Konfigurationen können nur über den Bereich **Statistik** angelegt werden, Report können aus dem enaio® client nur von Benutzern erzeugt werden, die über den Eigenschaftsdialog einer Workflowfamilie dazu berechtigt werden.

Reports werden als XML-Dateien erzeugt. Im enaio® client und über den Bereich **Statistik** werden die Dateien formatiert oder als Quelltext angezeigt. Die Struktur dieser XML-Daten ist im Anhang (vgl. Report XML) dokumentiert. Diese Dokumentation soll Ihnen ermöglichen, die Daten für externe Auswertungstools zur Verfügung zu stellen.

Report-Konfigurationen

Reports können nur auf der Grundlage einer Reportkonfiguration erstellt werden. Konfigurationen erstellen Sie über den Bereich **Statistik**.



Sie können Übersicht- oder Detail-Konfigurationen erstellen. Eine Detail-Konfiguration ermöglicht, Reports für den Einblick in alle Aktivitäten eines Vorgangs zu erstellen, eine Übersichts-Konfiguration Reports für den Überblick über viele laufende Prozesse.

Prozessübersichts-Konfiguration

Für die Übersichts-Konfiguration geben Sie einen **Namen** an, legen fest, ob die Prozesse aller **Familien** oder nur einzelner Familien einbezogen werden und geben die **Detailltiefe** an: Übersicht, Einzelaufstellung pro Familie oder Einzelaufstellung pro Familie und Aktivität.

Dazu können Sie detailliert die Aufnahme einzelner Daten ein-/ausschalten (true/false).

The 'Reportkonfiguration' dialog box is shown with the 'Allgemein' tab selected. The configuration is as follows:

Allgemein	
Name	
Typ	Prozessübersicht
Ersteller	THOMAS
Erstellzeitpunkt	21/12/2011
ID	B4A98A32E87B44B6853762D434FB5849
Familien	Keine
Detailltiefe	0 (Übersicht)

Prozess	
Gesamtanzahl der Prozesse	Ja
Anzahl fehlerhafter Prozesse	Ja
Anzahl pausierter Prozesse	Ja
Laufzeit	Ja; Ja; Ja

Aktivitäten	
Gesamtanzahl der Aktivitäten	Ja
Anzahl beendeter Aktivitäten	Ja
Anzahl nicht beendeter Aktivitäten	Ja
Anzahl personalisierter Aktivitäten	Ja
Anzahl fehlerhafter Aktivitäten	Ja
Anzahl pausierter Aktivitäten	Ja
Laufzeit	Ja; Ja; Ja
Laufzeit beendeter Aktivitäten	Ja; Ja; Ja
Laufzeit nicht beendeter Aktivitäten	Ja; Ja; Ja
Laufzeit personalisierter Aktivitäten	Ja; Ja; Ja

Eskalationen (abgelaufene Fristen)	
Anzahl für Prozess	Ja
Gesamtanzahl	Ja

Buttons: OK, Abbrechen

Prozessdetails-Konfiguration

Für die Detail-Konfiguration geben Sie einen **Namen** an, legen fest, ob die Prozesse aller **Familien** oder nur einzelner Familien einbezogen werden und geben die **Detailltiefe** an: Prozesse oder Prozesse und Aktivitäten.

Dazu können Sie detailliert die Aufnahme einzelner Daten ein-/ausschalten (true/false).

Im Bereich **Datenfelder** geben Sie Variablen an, deren Wert mit ausgegeben werden soll.

The 'Datenfelder' dialog box is shown with the 'Datenfeld hinzufügen/bearbeiten' sub-dialog open. The sub-dialog has the following fields:

- Anzeigename:
- Variablen ID:
- Workflowfamilie:

Buttons: OK, Abbrechen

Die Variablen ID entnehmen Sie der Modellübersicht im enaio® editor-for-workflow.

The 'Reportkonfiguration' dialog box is shown with the 'Allgemein' tab selected. The configuration is as follows:

Allgemein	
Name	
Typ	Prozessdetails
Ersteller	THOMAS
Erstellzeitpunkt	21/12/2011
ID	FE05E339F09448AB82FD3D1954F9CF16
Familien	Keine
Detailltiefe	0 (Prozesse)

Prozess	
ID	Ja
Name	Ja
Betreff	Ja
Startzeitpunkt	Ja
Laufzeit	Ja
Status	Ja
Modell-ID	Ja
Modellname	Ja
Familien-ID	Ja
Familienname	Ja
Gesamtanzahl der Aktivitäten	Ja
Anzahl beendeter Aktivitäten	Ja
Anzahl nicht beendeter Aktivitäten	Ja
Anzahl personalisierter Aktivitäten	Ja
Anzahl fehlerhafter Aktivitäten	Ja
Anzahl pausierter Aktivitäten	Ja

Aktivitäten	
ID	Ja

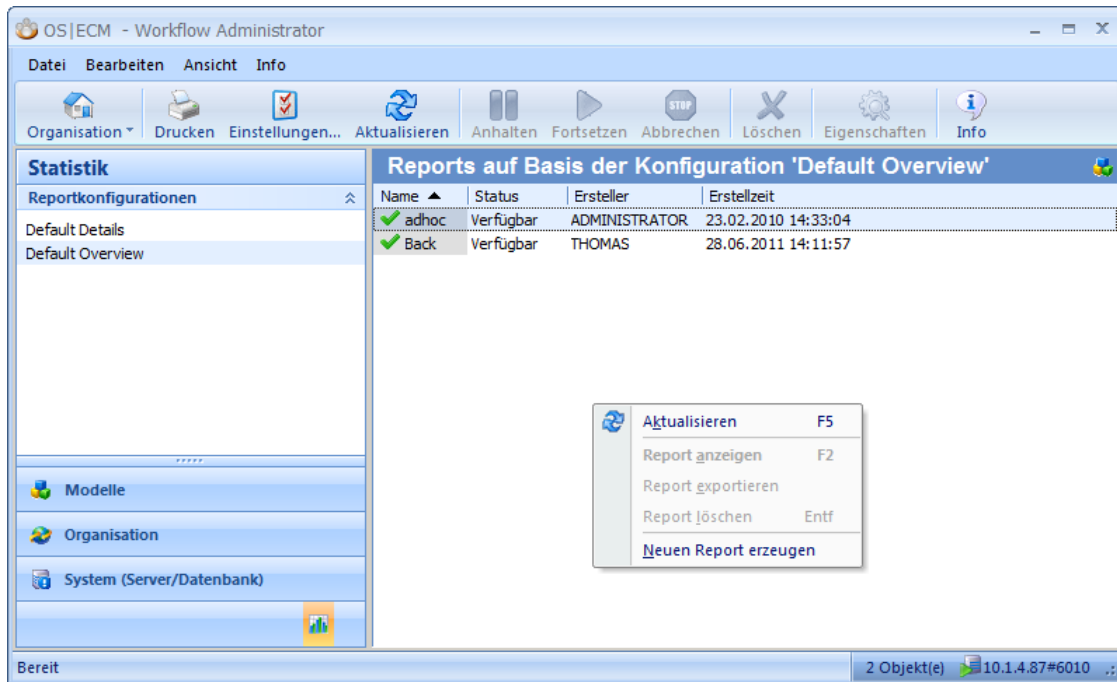
Datenfelder

Buttons: OK, Abbrechen

Reports erzeugen

Reports können nur auf der Grundlage einer Reportkonfiguration erstellt werden. Sie können über den enaio® client von Benutzern erstellt werden, die die entsprechende Berechtigung erhalten haben, oder über den enaio® administrator-for-workflow.

Sie markieren die gewünschte Reportkonfiguration und wählen aus dem Kontextmenü über dem Reportbereich den Eintrag **Neuen Report erzeugen**.



Für den Report geben Sie einen **Namen** ein und den **Ausführungszeitpunkt**.

Über den periodischen Job 'Workflow SpoolerJob' ist festgelegt, in welchem Rhythmus nach vorliegenden Report-Anforderungen gesucht wird. Diesen Job konfigurieren Sie über den enaio® enterprise-manager. Voreingestellt ist ein Rhythmus von einer Minute.

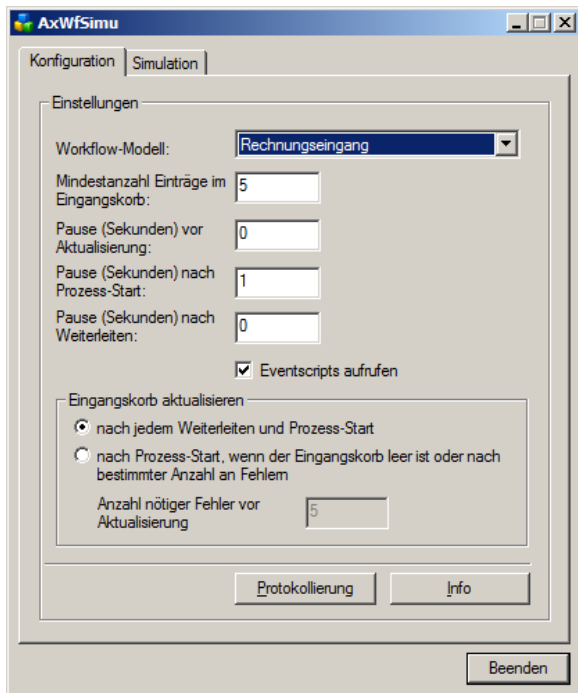
Workflow Simulation

Die Anwendung `axwfsimu.exe` erlaubt das automatische Testen von Workflowprozessen. Die Datei befindet sich im Installationsverzeichnis `...clients\admin`. Im Namen des angemeldeten Benutzers werden Prozesse eines ausgewählten Workflowmodells gestartet und deren Arbeitsschritte automatisiert weitergeleitet. Die Benutzerinteraktion während des Prozessverlaufs über Masken wird dabei gegebenenfalls mittels eines besonderen Eventskripts simuliert. Um mehrere Workflowbenutzer zu simulieren kann `axwfsimu.exe` mehrfach parallel ausgeführt werden.

Die so ausgeführten Prozesse werden vom System wie alle anderen Workflowprozesse behandelt und können daher z.B. über den Workflow Administrator überwacht und statistisch ausgewertet werden.

Um die Anwendung zu starten, führen Sie die Datei `axwfsimu.exe` aus. Der enaio® client wird automatisch gestartet. Nach der Anmeldung am enaio® client erscheint der Konfigurationsdialog.

Konfiguration



Auf der Registerkarte **Konfiguration** von `axwfsimu.exe` können folgende Einstellungen vorgenommen werden:

Mit dem Eintrag **Workflow-Modell** kann aus der Menge der startbaren Vorgänge des angemeldeten Benutzers das zu simulierende Modell ausgewählt werden. Bei der Simulation werden ausschließlich Vorgangsschritte dieses Modells angezeigt und automatisiert abgearbeitet.

Über **Mindestanzahl Einträge im Eingangskorb** geben Sie an, wie viele Einträge der Eingangskorb des Benutzers mindestens enthalten soll, bevor eines dieser Elemente bearbeitet wird. Sind weniger Elemente im Eingangskorb als hier eingestellt ist, so wird stattdessen ein neuer Prozess gestartet. Soll während der Simulation nie ein Prozess gestartet werden sondern ausschließlich eine Abarbeitung des Eingangskorbes erfolgen, so muss dieser Wert auf 0 gesetzt sein.

Um den Server während der Simulation nicht durch pausenlose Anfragen zu überlasten, können verschiedene Wartezeiten definiert werden. Die jeweiligen Werte entsprechen der Pause in Sekunden:

Pause vor Aktualisierung	Bevor der Eingangskorb über eine Serveranfrage aktualisiert werden soll, kann eine Pause in der Simulation erzwungen werden.
Pause nach Prozessstart	Nachdem ein neuer Prozess gestartet wurde, kann eine Pause in der Simulation erzwungen werden.
Pause nach Weiterleiten	Nachdem ein Arbeitsschritt weitergeleitet wurde, kann eine Pause in der Simulation erzwungen werden.

Aktivieren Sie die Option **Eventscripts aufrufen**, um festzulegen, ob die Client-seitigen Skripte "BeforeOpen", "SimulateMaskEdit" und "BeforeForward" bei der Bearbeitung eines Vorgangsschrittes ausgeführt werden sollen.

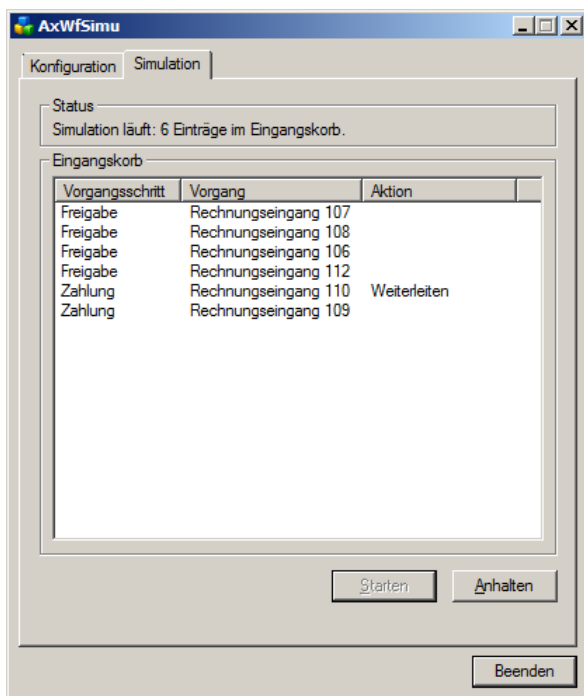
Im Bereich **Eingangskorb aktualisieren** legen Sie fest, wann der Eingangskorb über eine Serveranfrage aktualisiert werden soll:

Wählen Sie **Nach jedem Weiterleiten und Prozess-Start** wird der Eingangskorb immer aktualisiert. Allerdings hat das eine höhere Auslastung des Servers zur Folge.

Wird **Nach Prozess-Start, wenn der Eingangskorb leer ist oder nach bestimmter Anzahl an Fehlern** gewählt, verringert sich die Last auf dem Server, da weniger Aktualisierungen ausgeführt werden. Jedoch nimmt man dafür in Kauf, dass der Eingangskorb nicht immer mit seinem tatsächlichen Zustand bekannt ist. So sind möglicherweise noch Arbeitsschritte sichtbar, die schon durch einen anderen Benutzer bearbeitet wurden. Wird ein solcher Arbeitsschritt geöffnet, wird dies als Fehler gewertet. Es kann hier konfiguriert werden, nach wie vielen solcher Fehler der Eingangskorb aktualisiert werden soll.

Beim Beenden von `axwfsimu.exe` werden die vorgenommenen Einstellungen automatisch gespeichert, so dass sie bei einem Neustart wieder zur Verfügung stehen.

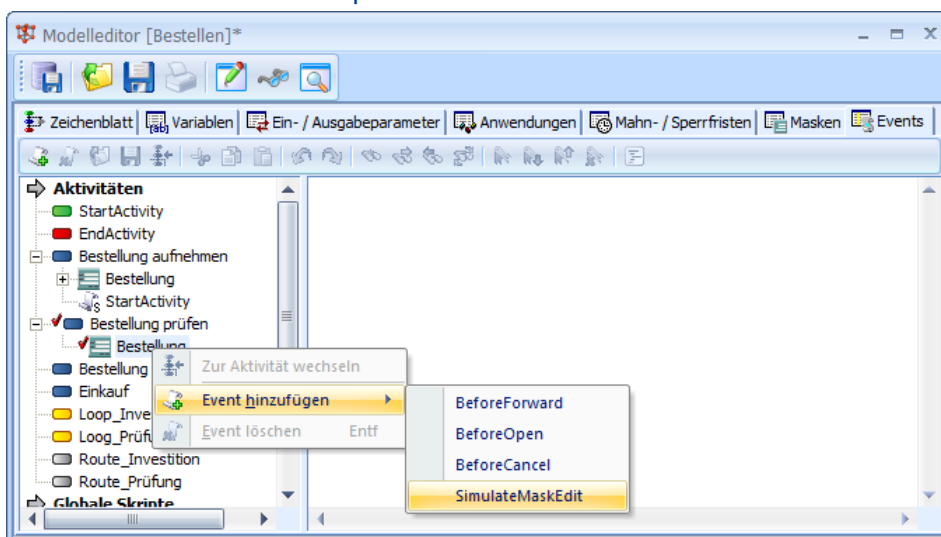
Simulation



Auf der Registerkarte **Simulation** kann die Simulation über entsprechende Schaltflächen gestartet und angehalten werden. Es kann zwar jederzeit zwischen den Registerkarten **Simulation** und **Konfiguration** gewechselt werden, allerdings kann die Konfiguration nicht verändert werden, solange die Simulation läuft.

Über den Eingangskorb ist zur Simulationslaufzeit ersichtlich, welche Arbeitsschritte von Prozessen des konfigurierten Modells der angemeldete Benutzer momentan sieht und welche Aktion gerade simuliert wird.

SimulateMaskEdit – Skripte



Da die Simulation voll automatisch durchlaufen wird, können keine Workflowmasken geöffnet werden. Trotzdem sollen möglicherweise auch die Benutzerinteraktionen simuliert werden. Dazu existiert ein eigenes Event "SimulateMaskEdit", das wie alle anderen Events über den Workfloweditor eingerichtet wird. Hier können nun z.B. Variablenwerte des Prozesses manipuliert werden.

"SimulateMaskEdit" wird nach dem Event "BeforeOpen" und vor dem Event "BeforeForward" ausgeführt, sofern die Eventskripte über die Konfiguration aktiviert wurden.

Kommandozeilenparameter

Die Anwendung `axwfsimu.exe` kann zusätzlich mit folgenden Kommandozeilenparametern gestartet werden:

Name	Parameter / Erläuterung
------	-------------------------

-wf	Setzt die 32-stellige ID des zu simulierenden Workflowmodells.
-me	Setzt Mindestanzahl der Einträge im Eingangskorb.
-es	Wenn auf 1 gesetzt, werden Eventskripte ausgeführt, wenn auf 0 gesetzt nicht.
-ps	Setzt die Pause in Sekunden nach Prozessstart.
-pf	Setzt die Pause in Sekunden nach dem Weiterleiten eines Arbeitsschrittes.
-pr	Setzt die Pause in Sekunden vor der Aktualisierung des Eingangskorbes.
-ra	Wenn auf 1 gesetzt, wird der Eingangskorb nach jedem Weiterleiten eines Arbeitsschrittes oder nach Prozessstart aktualisiert. Wenn auf 0 gesetzt, wird der Eingangskorb nach Prozessstart, bei leerem Eingangskorb oder einer bestimmten Anzahl an Fehlern aktualisiert.
-rf	Setzt die Anzahl der Fehler, nach der der Eingangskorb aktualisiert wird.
-up	Setzt Benutzer und Passwort um den Login-Dialog zu verhindern, z.B. user@password.
-ao	Wenn auf 1 gesetzt, wird die Simulation sofort gestartet.

Beispiel:

```
axwfsimu -up user1@pw -es 0 -me 5 -ao 1 -wf 350DD718818F4482B9E708BC6088B61B
```

Startet `axwfsimu.exe` für Benutzer "user1" mit dem Passwort "pw". Es sollen keine Eventskripte ausgeführt werden und die Mindestanzahl an Elementen im Eingangskorb soll 5 sein. Es werden Prozesse des Modells mit der ID "350DD718818F4482B9E708BC6088B61B" bearbeitet. Die Simulation wird sofort mit Start von `axwfsimu.exe` begonnen.

Starten Sie `axwfsimu.exe` mit dem Parameter `-ao 1` wird die Simulation sofort gestartet, ohne dass der **AxWfsimu** Dialog angezeigt wird.

Automatisiertes Starten von Workflowprozessen

Neben dem Start von Workflowprozessen aus dem enaio® client ist es möglich, im Anschluss an eine Massenindexierung oder durch einen Import Workflowprozesse zu initiieren und dabei Dokumente zu übernehmen.

Weiterhin können Prozesse über Scriptcode, der in einer Workflow-Startkomponente läuft, gestartet werden, allerdings ohne Übernahme von Dokumenten.

Ebenfalls möglich ist das Starten von Workflowprozessen mit der Übernahme von E-Mail-Daten aus Microsoft Outlook.

Integration über den Importassistenten

enaio® capture ist die Massenindexierungskomponente innerhalb der Produktfamilie enaio®. Sie dient dazu, Dokumente in Batches digital zu erfassen, eine automatische oder manuelle Verschlagwortung durchzuführen und danach die indexierten Dokumente in das Dokumentenmanagementsystem zu importieren. Die Zuordnung der erfassten Daten und Dokumente zu Objekten in enaio® übernimmt der Importassistent. Dieser kann für eine Konfiguration einer automatischen Importaktion oder für enaio® capture die erfassten Daten und Dokumente an Workflowprozesse weiterleiten.

Ein typisches Beispiel für diesen Einsatz aus enaio® capture ist ein Posteingangsszenario. Hier werden in einer zentralen Poststelle Dokumente (z.B. Lieferscheine, Bestellungen oder sonstige Belege) erfasst und nach der Aufnahme der Verschlagwortungsdaten entsprechende Prozesse zur Bearbeitung gestartet. Dies kann je nach Art der Belege eine Bestellung, eine Lieferung oder ein sonstiger Prozess sein, der die Teilnahme mehrerer Rollen in unterschiedlichen Abteilungen erfordert. Typischerweise werden die Belege nach Absender indexiert und das Geschäftszeichen eingegeben und danach an den zuständigen Sachbearbeiter oder die zuständige Abteilung weitergeleitet. Hier erfolgt dann die eigentliche Bearbeitung des gewünschten Geschäftsprozesses, der dann im zugehörigen Modell designet ist.

Workflow-Skript-Komponente

Die Workflow-Skript-Komponente ist ein Kommandozeilenprogramm, das zum Ausführen von Scriptcode geeignet ist. Dem Scriptcode werden bereits initialisierte COM-Objekte (`wfclient` und `assystem`) übergeben, das im Skript zum Starten von Prozessen benutzt werden kann. Die Workflow-Start-Komponente wird im Admin-Verzeichnis installiert und heißt `axwfscrip.exe`.

Das Programm kann mit unterschiedlichen Kommandozeilenparametern gestartet werden, die nachfolgend erläutert sind:

Parameter	Typ	Bedeutung	Bemerkung
-u	Optional	Benutzername@Passwort	Wenn Benutzername und Passwort angegeben sind, wird der Anmeldedialog übersprungen. Ansonsten wird eine normale Anmeldung (automatisch / Dialog) ausgeführt
-s	Required	Script	Dateiname des Scriptcodes, der ausgeführt werden soll.
-h, -?, /h /?, kein Parameter		Help	Hilfetext mit Angabe der möglichen Argumente
-q	Optional	Quite output	Wenn angeschaltet, erfolgt kein Output auf der Konsole.
-t + Sekundenangabe	Optional	Intervall	Wenn angegeben, dann wird ein Timer gesetzt, der im angegebenen Intervall das Skript ausführt. Ansonsten wird das Programm nach Ausführung des Skripts beendet.
-srv	Optional	Server#Port	Wenn angegeben, wird die Serverkonfiguration aus der asinit.cfg ignoriert und stattdessen der über den Parameter spezifizierte Server angesprochen.

Über den -t-Parameter ist es somit möglich, das Programm im Dauerbetrieb laufen zu lassen und dabei periodisch Skripte auszuführen. Über diese Option kann folgendes Szenario realisiert werden:

Über einen Webserver werden Anfragen (Bestellungen, Rechnungen) aus dem Internet empfangen. Die dazugehörigen Daten werden in Dateien abgespeichert. Im Scriptcode wird dieses Verzeichnis überwacht. Wenn eine Datei gefunden wird, dann werden die Informationen gelesen und in Form von Inputparametern an einen neu zu startenden Prozess übergeben. Damit erreicht man eine automatische Anbindung von Subsystemen an das Workflowsystem und hat gleichzeitig die Flexibilität, die Daten über VBScript anzupassen.

Beispiel

Über nachfolgende Kommandozeile wird das Skript c:\test.vbs alle 30 Sekunden unter dem Account des Benutzers root ausgeführt. Ein Output auf die Konsole erfolgt nicht.

```
Axwfscrip. Exe -u root@masterkey -s c:\test.vbs -q -t 30
```

Beispielcode zum Starten eines Prozesses

Zum Starten von Prozessen kann etwa folgender Code verwendet werden. Dieser prüft in der Menge der startbaren Vorgänge, ob es ein Prozessmodell mit dem Namen „Bestellung“ gibt und wenn das der Fall ist, dann werden die Startparameter des Prozesses mit Werten belegt.

```
'Startbare Prozesse holen
Set wfitems = wfclient.wfitems
For i =0 To wfitems.count -1
    Set wfitem = wfitems.item(i)
    If wfitem.name = "Bestellung" Then
'Startvariablen festlegen
        Set inputvars = wfitem.inputvariables
        inputvars.Item("Variable1").value = "abx"
        inputvars.item("Variable2").value = "12434"
'Vorgang starten
        iRet = wfitem.startprocess (Nothing, inputvars, varid, varerror)
        If iRet <> 0 Then
```

```

    msgbox "Fehler bei StartProcess " & iRet
Else
    msgbox "Prozess gestartet "
End If
End If
Next

```

Das Workflow-Skript-Programm wird Arbeitsplatz bezogen mit dem Lizenzschlüssel 'AVG' lizenziert.

enaio® document-storage

Workflowprozesse können aus E-Mail-Clients wie Microsoft Outlook, Novell GroupWise und Lotus Notes mit der Übergabe von E-Maildaten gestartet werden. Diese Funktion stellt enaio® document-storage zur Verfügung.

Der Benutzer startet aus einem E-Mail-Client über eingebundene Menüs die verfügbaren Workflows mit der Übergabe von E-Mails oder E-Maildaten.

Über die Taskleiste wird enaio® document-storage zur Konfiguration gestartet. Hier können Zuordnungen zwischen E-Mail-Daten und Workflowvariablen festgelegt werden.

Informationen dazu finden Sie im Handbuch 'enaio® mail-management'.

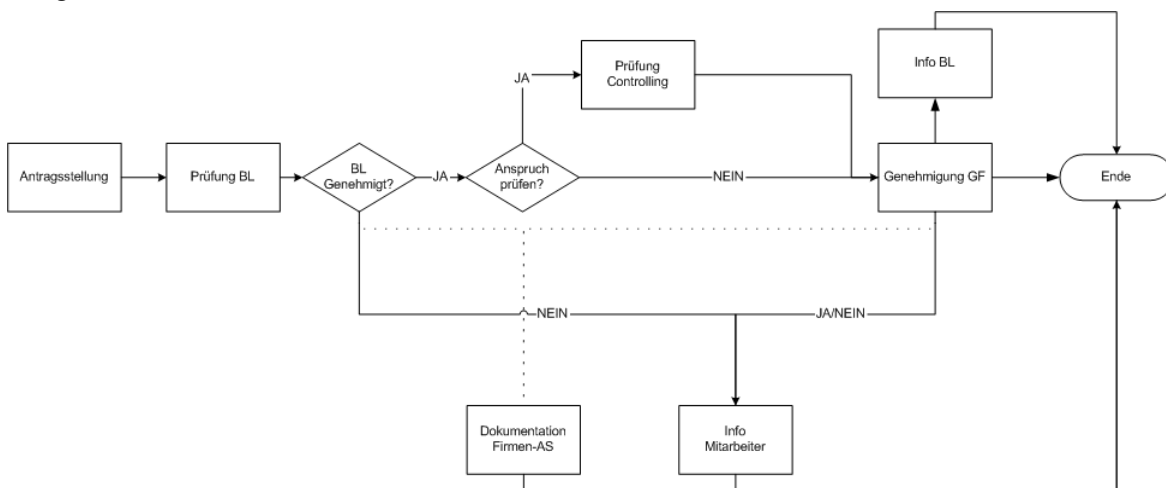
Beispiele für Workflowszenarien

Urlaubsworkflow

Der Prozess beginnt mit der Antragsstellung des Mitarbeiters. Zuvor wurde der Urlaubsanspruch für alle Mitarbeiter des Unternehmens in einer separaten Datenbank erfasst. Der Mitarbeiter wählt bei der Antragstellung aus, ob Urlaub oder Arbeitszeitausgleich beantragt werden soll. Der Unterschied der beiden Modi besteht darin, dass bei Arbeitszeitausgleich keine Anrechnung auf den Urlaubsanspruch erfolgt. Der Mitarbeiter wählt den Beginn und das Ende des Zeitraums aus. Es muss die Möglichkeit bestehen, halbe Tag zu beantragen. Bei Weiterleiten des Urlaubsantrags erfolgt automatisch die Berechnung des verbleibenden Urlaubsanspruchs.

Der Mitarbeiter hat des Weiteren die Möglichkeit, eine Prüfung seines derzeitigen Urlaubsanspruchs durch das Controlling einzufordern. Dies geschieht durch Setzen der Checkbox 'Urlaubsanspruch prüfen'.

Der Antrag wird dann an den zuständigen Bereichsleiter (BL) weitergeleitet. Dieser prüft den Antrag dahingehend, ob der Mitarbeiter zum beantragten Zeitraum frei nehmen kann. Die Ermittlung des zuständigen BL sollte über die Organisationsstruktur erfolgen. Ist dies aus funktionalen Gründen nicht möglich, sollte in der Urlaubsdatenbank der Bereich, zu dem der Mitarbeiter gehört, vermerkt werden. Die Auswahl der Zielrolle muss auf jeden Fall dynamisch erfolgen.



Bei Antrag auf Arbeitszeitausgleich prüft der BL den Anspruch auf Arbeitszeitausgleich. Der BL genehmigt den Antrag durch Setzen einer Checkbox. Bei Genehmigung wird geprüft, ob der Mitarbeiter eine Prüfung des Urlaubsanspruchs verlangt hat. Ist das so, erfolgt einer Weiterleitung des Antrages an das Controlling. Ist keine Prüfung verlangt, erfolgt die Weiterleitung an den Geschäftsführer. Bei Ablehnung durch den BL erfolgt eine Benachrichtigung des Mitarbeiters (Aktivität Info Mitarbeiter) sowie die Dokumentation des Vorgangs im DMS. Der Geschäftsführer bearbeitet den Antrag abschließend. Bei Genehmigung des Antrages erfolgt die Benachrichtigung des Mitarbeiters, des Bereichsleiters, die Dokumentation im DMS sowie die Eintragung im allgemeinen Outlook-Kalender des Unternehmens. Bei Ablehnung erfolgen die gleichen Benachrichtigungen sowie die Dokumentation im DMS.

Bestellworkflow

In diesem Beispiel handelt es sich um eine komplexer Bearbeitung einer Bestellanforderung (BANF) mit einer zusätzlichen Prüfung der Kompetenzrichtlinien in Abhängigkeit von Kostenstelle, Position des Mitarbeiters und Bestellwert.



Eine Bedarfsmeldung darf durch jeden Mitarbeiter des Unternehmens erfolgen. Der Prozess kann gestartet werden, wenn die Leistungsbeschreibung feststeht. Diese hat formlos schriftlich zu erfolgen und wird dem Prozess beigelegt. Etwaige Angebote, die bereits durch den Bedarfsmelder eingeholt wurden, können der Bedarfsmeldung beigelegt werden. (*Definition: Bedarfsmelder ist derjenige Mitarbeiter, der einen BANF-Prozeß startet.*). Der Bedarfsmelder definiert die zu belastende Kostenstelle. Des Weiteren definiert er die Bedarfsart (*Definition: Die Bedarfsart klassifiziert den Gegenstand der Bestellung. Aufgrund der Bedarfsart wird die entsprechende Kompetenzlinie ausgewählt.*). Im Workflow werden alle Bedarfsarten der Kompetenzrichtlinie abgebildet.

Rollen: Bedarfsmelder

Bei Zustimmung der einzelnen Hierarchiestufen, erfolgt eine Weiterleitung laut Kompetenzregelung. Nach der Zustimmung durch die Hierarchiestufe TS (Teamsprecher) erfolgt eine informelle Mitbeteiligung des Bereichscontrollings. Hier wird geprüft, ob ein entsprechendes Budget vorhanden ist. Die Information hierüber wird im Workflow abgebildet. Danach durchläuft der Prozess die weitere Zustimmung laut Kompetenzrichtlinie. Bei Bedarfarten, die eine Beteiligung von TS und AL (Abteilungsleiter) am Zustimmungsprozess nicht vorsehen, ist das Bereichscontrolling direkt nach der Bedarfsmeldung einbezogen.

Rollen:

- ## Sonderregelung nach Bedarfsart

Wenn die Ware am Lager ist, dann erfolgt eine Rückmeldung an den Bedarfsmelder, wann und wo er die Ware abholen kann. Handelt es sich bei der bestellten Ware um ein Anlagegut (Bestellwert>410€, muss die Anlagenbuchhaltung bei FI informiert werden. Dies erfolgt durch das Formular „Veränderungsmeldung für Anlagegüter“. Das Formular wird durch den Lageristen ausgefüllt.

Im Anschluss erfolgt die Weiterleitung an FI.

- enaio®

ALLER bisher am Prozess Beteiligten. Der Prozess wird beendet. Eine Wiederaufnahme dieses Bestellprozesses durch Änderung der Bestellung durch den Bedarfsmelder ist nicht vorgesehen. Der Bedarfsmelder muss in diesem Fall einen neuen BANF-Prozess anstoßen. Er hat aber die Möglichkeit, auf die bisher angefallenen Dokumente im Bestellarchiv zuzugreifen und muss somit z.B. die Leistungsbeschreibung nicht nochmals vornehmen.

Wenn Ja (Bestellung fachlich-inhaltlich korrekt), erfolgt eine Prüfung, ob die Ware am Lager ist. Wenn die Ware am Lager ist, dann erfolgt eine Rückmeldung an den Bedarfsmelder, wann und wo er die Ware abholen kann. Handelt es sich bei der bestellten Hard- und Software um ein Anlagegut (Bestellwert > 410€), muss die Anlagenbuchhaltung informiert werden. Dies erfolgt durch das Formular „Veränderungsmeldung für Anlagegüter“. Das Formular wird durch den IT-Facheinkaufsverantwortlichen ausgefüllt. Zusätzlich hat eine Benachrichtigung eines Mitarbeiters der Rolle: IT-Einkauf zu erfolgen. Ist die Ware nicht am Lager, wird dies im Workflow dokumentiert. Es erfolgen keine weiteren Handlungen durch den IT-Facheinkaufsverantwortlichen. Im Anschluss erfolgt die Weiterleitung an die Rolle FI.

Rollen:

1. Lagerist
2. Anlagenbuchhalter
3. IT-Facheinkaufsverantwortlicher
4. IT-Einkauf

FI-Finanzbuchhaltung

An diesem Arbeitsschritt wird durch den Sachbearbeiter FI ein Sachkonto in die BANF eingetragen. Handelt es sich um ein Anlagegut, wird die Anlagenklasse in die BANF eingetragen und im SAP ein Anlagenstammsatz angelegt.

Der Sachbearbeiter dokumentiert seine ordnungsgemäßen Handlungen im Workflow indem er einen Haken in das Kontrollkästchen *Kontierung durchgeführt* setzt.

Rollen:

Kontierung (FI)

Controlling

An diesem Prozessschritt erfolgt eine Prüfung durch das zentrale Controlling. Die Zustimmung bzw. Ablehnung wird durch eine entsprechende Checkbox im Workflow dokumentiert. Erfolgt eine Ablehnung durch das Controlling, ist diese in einer entsprechenden Anmerkung/Aktennotiz zu begründen bzw. zu dokumentieren. In diesem Falle erfolgt eine Benachrichtigung ALLER bisher am Prozess beteiligten. Der Prozess wird danach beendet. Erfolgt eine Zustimmung durch das Controlling, erfolgt die Weiterleitung an den Einkauf.

Rollen:

zentrales Controlling

Einkauf

Durch den Einkauf erfolgt die Festlegung der Lieferanten. Dies wird nicht im Workflow dokumentiert bzw. durch den Workflow unterstützt. Im SAP werden Angebotsanforderungen an die ausgewählten Lieferanten erstellt. Diese werden über den SAP ArchiveLink im Bestellarchiv abgelegt. SAP vergibt jeweils eine Angebotsanforderungsnummer. Diese wird im Workflow/Bestellarchiv dokumentiert.

Im Weiteren wird auf den Eingang der Angebote gewartet. Jedes Angebot wird zunächst zur inhaltlichen Prüfung an den Bedarfsmelder gegeben. Stimmt dieser dem Angebot inhaltlich zu, erfolgt eine kaufmännische kommerzielle Prüfung durch den Einkauf. Die Angebotsprüfungen sind durch den Workflow zu unterstützen.

Der Einkauf entscheidet sich zu einem bestimmten Zeitpunkt für eines der eingegangenen Angebote und führt die Bestellung aus.

Rollen:

Einkauf<Bedarfsart> (z.B. Einkauf-Hard- und Software)

Bestellung

Die Bestellung wird in SAP generiert. Die dabei generierte Bestellnummer ist im Workflow zu dokumentieren. Es wird eine Benachrichtigung des Bedarfsmelders mit der Bestellnummer und dem voraussichtlichen Liefertermin erzeugt. Die Bestellung sollte über den SAP ArchiveLink im Bestellarchiv abgelegt werden.

Rollen:

Einkauf<Bedarfsart> (z.B. Einkauf-Hard- und Software)

Prozessabschluss

Es wird automatisch ein Protokoll mit allen relevanten Ablaufdaten erzeugt und im Bestellarchiv abgelegt.

- § Kombierter strukturierter und Ad-Hoc Workflow zur flexiblen Bearbeitung von Vorgängen bei Genehmigungsverfahren
- § Posteingangsszenario (zentrale Erfassung von Dokumenten in der Poststelle und automatische Weiterleitung an die entsprechenden Abteilungen und Sachbearbeiter bis hin zur Freigabe von Antwortschreiben und Ablage im DMS)
- § Rechnungseingang (zentrale Erfassung von Rechnungen in der Poststelle, inhaltliche, administrative und finanzielle Freigabe durch unterschiedliche Abteilungen bis zur Übergabe an das Rechnungswesen)

Anhang

Analysebogen für Geschäftsprozesse

Verbale Beschreibung des Geschäftsprozesses

Teilnehmer am Geschäftsprozess

Zusammenfassung der Aktivitäten

Nr.	Aktivität	Beschreibung	Beteiligte Rollen
1	Startaktivität	<diese Aktivität ist in allen Modellen vorhanden>	<hier sind die Rollen einzutragen, die den Prozess starten dürfen>

Eingabemasken

Aktivität 1:

Aktivität n:

Übergänge zwischen den Aktivitäten (Transitionen)

Nr.	Von Aktivität	Nach Aktivität	Bedingung

Parallel auszuführende Aktivitäten

Nr.	Aktivität	Beschreibung	Beteiligte Rollen

Workflowakte

An welchen Aktivitäten steht eine Workflowakte zur Verfügung?

Nr.	Aktivität	Zugelassene Dokumenttypen	Minimale Anzahl	Maximale Anzahl

Können Vorgänge aus dem Client mit einem Objekt gestartet werden?

Was soll am Ende des Prozesses mit den Dokumenten der Workflowakte passieren?

Clientskripte

Welche Events (Initialisierungen, Plausibilitätschecks etc.) sollen bei der Bearbeitung von Aktivitäten ausgeführt werden?

Nr.	Aktivität	Eventtyp	Beschreibung

Programmierreferenz

Umgebung

Für die Programmierung von Eventcode und anderen Skripten in der Workflow-Engine für den enaio® client wird die Programmiersprache Microsoft Visual Basic Script verwendet. Diese Programmiersprache ist auf allen Windows Systemen verfügbar und stellt umfangreichen Möglichkeiten zum Zugriff auf die Betriebssystemumgebung, auf das Dateisystem, Datenbanken und Objekte zur Verfügung. Eine genaue Beschreibung der Programmiersprache ist der MSDN innerhalb von Microsoft Entwicklungsumgebungen oder dem Internet unter

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vbscripttoc.asp> zu entnehmen.

Hier sollen nur kurz die wichtigsten Aspekte der Sprache beschrieben werden. Die Skriptsprache stellt einige einfache Funktionen zur Verfügung. Weitere Funktionen sind durch Referenzierung auf betriebssysteminterne Objekte, wie das File System Object oder das Active Data Object, sowie auf zusätzlich erstellte skriptfähige Objekte möglich.

Informationen zu Events im enaio® webclient finden Sie im Handbuch 'Workflow Event-Code für OSWEB'. Das Dokument befindet sich im Verzeichnis 'Win32\Disk1\OSWEB\Doc' auf der Installations-DVD.

Standardfunktionen von VBScript

Standardmäßig werden in VBScript folgende Funktionen angeboten. Der Zugriff kann direkt aus dem Scriptcode ohne vorherige Initialisierung eines Objektes erfolgen.

Abs	Array	Asc	Atn
CBool	CByte	CCur	CDate
CDBl	Chr	CInt	CLng
Conversions	Cos	CreateObject	CSng
Date	DateAdd	DateDiff	DatePart
DateSerial	DateValue	Day	Derived Maths
Eval	Exp	Filter	FormatCurrency
FormatDateTime	FormatNumber	FormatPercent	GetLocale
GetObject	GetRef	Hex	Hour
InputBox	InStr	InStrRev	Int, Fixs
IsArray	IsDate	IsEmpty	IsNull
IsNumeric	IsObject	Join	LBound
LCase	Left	Len	LoadPicture
Log	LTrim; RTrim; and Trims	Maths	Mid
Minute	Month	MonthName	MsgBox
Now	Oct	Replace	RGB
Right	Rnd	Round	ScriptEngine
ScriptEngineBuildVersion	ScriptEngineMajorVersion	ScriptEngineMinorVersion	Second
SetLocale	Sgn	Sin	Space
Split	Sqr	StrComp	String
Tan	Time	Timer	TimeSerial
TimeValue	TypeName	UBound	UCase
VarType	Weekday	WeekdayName	Year

Eine genaue Erläuterung der einzelnen Funktionen und die benötigten Parameter sind den oben genannten Quellen zu entnehmen.

Active Data Object

Über das Active Data Object kann ein Zugriff auf beliebige Datenquellen erfolgen, zu denen eine ODBC-oder eine OLEDB-Verbindung besteht.

```
On Error Resume Next
'Hier Connectstring eintragen
'Beispiel: sConnect="PROVIDER=SQLNCLI;DSN=acm520;UID=sysadm;PWD=freund;"
Set db = CreateObject("ADODB.Connection")
If db Is Nothing Then
    'kein ADO verfügbar
Else
With db
    .CursorLocation = adUseClient
    .Open sConnect
    If .State <> 1 Then
        'fehler im Connect
        msgbox "Fehler bei der Datenbankverbindung" & vbCrLf & err.description
    Else
        msgbox "OK"
    End If
End With
End If
```

Über die Methode `Execute` kann dann ein SQL-Statement ausgeführt werden. Handelt es sich um eine SQL-Select-Anweisung, so erhält man ein Recordset mit den Datensätzen. Eine genaue Beschreibung der Objekte und Methoden zu ADO sind der MSDN im Internet unter <http://msdn.microsoft.com> zu entnehmen.

File System Object

Die File System Object-Schnittstelle ist Bestandteil des Betriebssystems Microsoft Windows. Über diese Schnittstelle können fast beliebige Dateioperationen, wie das Kopieren, Umbenennen und Löschen von Dateien und Verzeichnissen durchgeführt werden. Weiterhin können hierüber Textdateien gelesen und beschrieben werden.

Folgendes Beispiel verdeutlicht den Zugriff auf das File System Object.

```
Dim fso, MyFile
Set fso = CreateObject("Scripting.FileSystemObject")
Set MyFile = fso.CreateTextFile("c:\testfile.txt", True)
MyFile.WriteLine("This is a test.")
MyFile.Close
```

Eine komplette Referenz der Methoden der File System Object-Schnittstelle finden Sie unter:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/sgfilesystemobjects.asp>

Darin enthalten sind auch Beispielpcodes und ausführliche Beschreibungen der Funktionen.

Schnittstellenobjekte

Für Server- und Client-Events werden COM-Schnittstellen bereitgestellt. Diese erlauben den Zugriff auf die Workflowmodelle, Organisationsstrukturen, Prozesse und Vorgangsschritte sowie Workflowakten und Dokumente. Eine vollständige technische Dokumentation der Objekte und Methoden mit Beispielcode finden Sie im Dokument 'COM-Objekte für enaio®-Workflow'.

Reportkonfiguration XML

Allgemein

StatisticReportConfiguration Tag

Jede Reportkonfiguration wird von einem StatisticReportConfiguration Tag umschlossen. Dieses hat folgende Attribute:

Id	Kennung des Reports
Name	Name des Reports
Type	Typ des Reports (Prozessübersicht = 0, Prozessdetails = 1)
OrganisationId	Id der Organisation für die der Report gilt
CreatorId	Id des Erstellers der Konfiguration
CreatorName	Name des Erstellers der Konfiguration
CreationTime	Zeitpunkt der Erstellung der Konfiguration (Zeitstempel)
Version	Hauptversion des Reports (z.B. 550)
SubVersion	SubVersion des Reports (z.B. 1)
MinProcessVersion	Hauptversion die zum Zeitpunkt der Erstellung eines Prozesses mindestens laufen musste, damit der Prozess in die Statistik eingeht. Z.B. 550.
MinProcessSubVersion	Subversion die zum Zeitpunkt der Erstellung eines Prozesses mindestens laufen musste, damit der Prozess in die Statistik eingeht. Z.B. 0.

Prozessübersichtreportkonfiguration

Rows Tag, DrillDownType Tag, FamilyIds und FamilyId Tag

Das Rows Tag ist im StatisticReportConfiguration Tag enthalten und besitzt selbst keine Attribute. Es enthält die Tags DrillDownType und FamilyIds.

Das DrillDownType Tag kann folgende Werte enthalten:

0	Erzeugung einer Gesamtübersicht, keine Auflösung nach Familien und Aktivitäten.
1	Erzeugung einer Gesamtübersicht und Einzelaufstellung pro Familie.
2	Erzeugung einer Gesamtübersicht sowie Einzelaufstellung pro Familie und Aktivität.

FamilyIds gruppiert FamilyId Tags, diese enthalten die Kennungen der Familien, deren Prozesse in Reports, die mit dieser Konfiguration erstellt werden, einfließen sollen. Wird als Familienkennung FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF (32mal F) verwendet, so bedeutet dies, dass alle Familien der Organisation verwendet werden sollen.

Columns Tag

Das Columns Tag ist im StatisticReportConfiguration Tag enthalten und besitzt selbst keine Attribute. Innerhalb des Columns Tags erscheinen die im Folgenden aufgeführten Tags, die jeweils ein Attribut namens Active enthalten, welches die entsprechende Option aktiviert (Active = 1) bzw. deaktiviert (Active = 0). Folgende Optionen für die Reports werden als Tags innerhalb des Columns Tag aufgeführt:

NumberOfProcesses	Gesamtanzahl der Prozesse
NumberOfActivities	Gesamtanzahl der Aktivitäten
NumberOfActivitiesUnfinished	Anzahl nicht beendeter Aktivitäten
NumberOfActivitiesFinished	Anzahl beendeter Aktivitäten
NumberOfActivitiesPersonalized	Anzahl personalisierter Aktivitäten
NumberOfProcessesError	Anzahl fehlerhafter Prozesse

NumberOfActivitiesError	Anzahl fehlerhafter Aktivitäten
NumberOfProcessesPause	Anzahl pausierter Prozesse
NumberOfActivitiesPause	Anzahl pausierter Aktivitäten
MinProcessRunTime	Minimale Prozesslaufzeit
MeanProcessRunTime	Durchschnittliche Prozesslaufzeit
MaxProcessRunTime	Maximale Prozesslaufzeit
MinActivityRunTime	Minimale Aktivitätenlaufzeit
MeanActivityRunTime	Durchschnittliche Aktivitätenlaufzeit
MaxActivityRunTime	Maximale Aktivitätenlaufzeit
MinActivityRunTimeFinished	Minimale Laufzeit beendeter Aktivitäten
MeanActivityRunTimeFinished	Durchschnittliche Laufzeit beendeter Aktivitäten
MaxActivityRunTimeFinished	Maximale Laufzeit beendeter Aktivitäten
MinActivityRunTimeUnfinished	Minimale Laufzeit nicht beendeter Aktivitäten
MeanActivityRunTimeUnfinished	Durchschnittliche Laufzeit nicht beendeter Aktivitäten
MaxActivityRunTimeUnfinished	Maximale Laufzeit nicht beendeter Aktivitäten
MinActivityRunTimePersonalized	Minimale Laufzeit personalisierter Aktivitäten
MeanActivityRunTimePersonalized	Durchschnittliche Laufzeit personalisierter Aktivitäten
MaxActivityRunTimePersonalized	Maximale Laufzeit personalisierter Aktivitäten
NumberOfEscalationsProcess	Gesamtanzahl an Eskalationen (abgelaufene Fristen)
NumberOfEscalationsActivity	Gesamtanzahl an Eskalationen (abgelaufene Fristen) von Aktivitäten
NumberOfEscalationsActivityUnfinished	Anzahl an Eskalationen (abgelaufene Fristen) von nicht beendeten Aktivitäten
NumberOfEscalationsActivityFinished	Anzahl an Eskalationen (abgelaufene Fristen) von beendeten Aktivitäten
NumberOfEscalationsActivityPersonalized	Anzahl an Eskalationen (abgelaufene Fristen) von personalisierten Aktivitäten

Prozessdetailreportkonfiguration

Rows Tag, DrillDownType Tag, FamilyIds und FamilyId Tag

Das Rows Tag ist im StatisticReportConfiguration Tag enthalten und besitzt selbst keine Attribute. Es enthält die Tags DrillDownType und FamilyIds.

Das DrillDownType Tag kann folgende Werte enthalten:

0	Es werden nur Einträge für Prozesse erstellt, es erfolgt keine Aufstellung der einzelnen Aktivitäten.
1	Es werden Einträge für Prozesse und Aktivitäten erstellt.

FamilyIds gruppiert FamilyId Tags, diese enthalten die Kennungen der Familien, deren Prozesse in Reports, die mit dieser Konfiguration erstellt werden, einfließen sollen. Wird als Familienkennung FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF (32mal F) verwendet, so bedeutet dies, dass alle Familien der Organisation verwendet werden sollen.

Columns Tag

Das Columns Tag ist im StatisticReportConfiguration Tag enthalten und besitzt selbst keine Attribute. Innerhalb des Columns Tags erscheinen die im Folgenden aufgeführten Tags, die jeweils ein Attribut namens Active enthalten, welches

die entsprechende Option aktiviert (Active = 1) bzw. deaktiviert (Active = 0). Folgende Optionen für die Reports werden als Tags innerhalb des Columns Tag aufgeführt:

Prozessdaten:

ProcessId	Kennung des Prozesses
ProzessName	Name des Prozesses
ProcessSubject	Betreff des Prozesses
FamilyId	Kennung der Familie
FamilyName	Name der Familie
ModelId	Kennung des Modells
ModelName	Name des Modells
ProcessStartTime	Startzeitpunkt des Prozesses
ProcessRunTime	Laufzeit des Prozesses
ProcessState	Status des Prozesses
NumberOfActivities	Gesamtanzahl der Aktivitäten
NumberOfActivitiesUnfinished	Anzahl nicht beendeter Aktivitäten
NumberOfActivitiesFinished	Anzahl beendeter Aktivitäten
NumberOfActivitiesPersonalized	Anzahl personalisierter Aktivitäten
NumberOfActivitiesError	Anzahl fehlerhafter Aktivitäten
NumberOfActivitiesPause	Anzahl pausierter Aktivitäten
NumberOfEscalationsProcess	Anzahl an Eskalationen (abgelaufene Fristen) im Prozess
NumberOfEscalationsActivity	Gesamtanzahl an Eskalationen (abgelaufene Fristen) von Aktivitäten
NumberOfEscalationsActivityUnfinished	Anzahl an Eskalationen (abgelaufene Fristen) von nicht beendeten Aktivitäten
NumberOfEscalationsActivityFinished	Anzahl an Eskalationen (abgelaufene Fristen) von beendeten Aktivitäten
NumberOfEscalationsActivityPersonalized	Anzahl an Eskalationen (abgelaufene Fristen) von personalisierten Aktivitäten

Aktivitätendaten (für DrillDownType = 1)

ActivityId	Kennung der Aktivität
ModActivityId	Kennung der Aktivität im Modell
ActivityName	Nam der Aktivität
ActivityStartTime	Startzeitpunkt der Aktivität
ActivityRunTime	Laufzeit der Aktivität
ActivityState	Status der Aktivität
NumberOfEscalationsActivity (das selbe Tag wie bei den Prozessdaten)	Anzahl der Eskalationen (abgelaufene Fristen) in der Aktivität

ProcessDatafields Tag, ProcessDatafield Tag

Das ProcessDatafields Tag befindet sich im Columns Tag und hat selbst keine Attribute. Es kann mehrere ProcessDatafield Tags enthalten, die globale Prozessvariablen identifizieren, deren Werte im Report ausgegeben werden. Das ProcessDatafieldTag hat folgende Attribute:

Id	Kennung der Prozessvariablen
FamilyId	Kennung der Id der Familie für die die Prozessvariable im Report ausgegeben werden soll
Name	Anzeigenname für die Prozessvariable in der gegebenen Familie

Report XML

Die Reports werden für den Anwender als HTML mit teilweise grafischen Elementen zur Anzeige aufbereitet. Es besteht zusätzlich die Möglichkeit, die generierten Reports als XML abzuspeichern, um die entsprechenden statistischen Daten detaillierter analysieren zu können. Im Folgenden wird dieses XML Format dokumentiert.

SR Tag (StatisticReport)

Jeder Report wird von einem SR Tag umschlossen. Es enthält immer genau ein StatisticReportConfiguration Tag, das die zu Grunde liegende Konfiguration beschreibt (siehe oben) sowie genau ein D Tag, welches die ermittelten statistischen Daten enthält (siehe unten). Das SR Tag hat folgende Attribute:

Id	Kennung des Reports
N (Name)	Name des Reports
CrT (CreationTime)	Zeitstempel der Erstellung des Reports
CrY (CreationYear)	Jahr der Erstellung des Reports
CrM (CreationMonth)	Monat der Erstellung des Reports
CrD (CreationDay)	Tag der Erstellung des Reports
CrH (CreationHour)	Stunde der Erstellung des Reports
CrU (CreationMinute)	Minute der Erstellung des Reports
CrS (CreationSecond)	Sekunde der Erstellung des Reports
CrI (CreatorId)	Kennung des Erstellers des Reports
CrN (CreatorName)	Name des Erstellers des Reports

Prozessübersichtsreport

D Tag (Data)

Das D Tag liegt im SR Tag und enthält selbst keine Attribute. Es enthält immer ein OD Tag und je nach „DrillDownType“ der Reportkonfiguration auch ein AD und/oder FD Tag.

OD Tag (OverviewData), Fls Tag (FamilyIds), FI Tag (Family Id)

Das OD Tag liegt im D Tag und liefert zusammenfassende Informationen zu den laufenden Prozessen. Zudem enthält es FI Tags (gruppiert über ein Fls Tag) die Kennungen aller Familien, deren Prozesse in dem Report betrachtet werden. Das OD Tag hat folgende Attribute (insofern die jeweiligen Optionen in der Reportkonfiguration aktiviert sind):

NP (NumberOfProcesses)	Gesamtanzahl laufender Prozesse
NPS (NumberOfProcessesPause)	Anzahl pausierter Prozesse
NPE (NumberOfProcessError)	Anzahl fehlerhafter Prozesse
sP (EscalationsProcesses)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Prozesse
iPR (MinimumProcessRuntime)	Minimale Prozesslaufzeit
ePR (MeanProcessRunTime)	Durchschnittliche Prozesslaufzeit
aPR (MaximumProcessRunTime)	Maximale Prozesslaufzeit
NA (NumberOfActivities)	Gesamtanzahl aller Aktivitäten laufender Prozesse
NAF (NumberOfActivitiesFinished)	Anzahl beendeter Aktivitäten
NAE (NumberOfActivitiesError)	Anzahl fehlerhafter Aktivitäten
NAS (NumberOfActivitiesPause)	Anzahl pausierter Aktivitäten
NAU (NumberOfActivitiesUnfinished)	Anzahl nicht beendeter Aktivitäten
NAP (NumberOfActivitiesPersonalized)	Anzahl personalisierter Aktivitäten
sa (EscalationsActivities)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Aktivitäten

sF (EscalationsActivitiesFinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) beendeter Aktivitäten
sU (EscalationsActivitiesUnfinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) nicht beendeter Aktivitäten
sAP (EscalationsActivitiesPersonalized)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) personalisierter Aktivitäten
iAR (MinActivityRunTime)	Minimale Aktivitätenlaufzeit
iAF (MinActivityFinishedRunTime)	Minimale Laufzeit beendeter Aktivitäten
IAU (MinActivityUnfinishedRunTime)	Minimale Laufzeit nicht beendeter Aktivitäten
iAP (MinActivityPersonalizedRunTime)	Minimale Laufzeit personalisierter Aktivitäten
eAR (MeanActivityRunTime)	Durchschnittliche Aktivitätenlaufzeit
eAF (MeanActivityFinishedRunTime)	Durchschnittliche Laufzeit beendeter Aktivitäten
eAU (MeanActivityUnfinishedRunTime)	Durchschnittliche Laufzeit nicht beendeter Aktivitäten
eAP (MeanActivityPersonalizedRunTime)	Durchschnittliche Laufzeit personalisierter Aktivitäten
aAR (MaxActivityRunTime)	Maximale Aktivitätenlaufzeit
aAF (MaxActivityFinishedRunTime)	Maximale Laufzeit beendeter Aktivitäten
AAU (MaxActivityUnfinishedRunTime)	Maximale Laufzeit nicht beendeter Aktivitäten
aAP (MaxActivityPersonalizedRunTime)	Maximale Laufzeit personalisierter Aktivitäten

FD Tag (FamilyData), FE Tag (FamilyDataEntry)

Das FD Tag liegt im D Tag, gruppiert alle FE Tags und enthält selbst keine Attribute. Das FD Tag existiert nur wenn dies in der Konfiguration über den „DrillDown“-Typ entsprechend festgelegt ist. Das FE Tag liefert Informationen zu einer Prozessfamilie und hat dazu folgende Attribute (insofern die jeweiligen Optionen in der Reportkonfiguration aktiviert sind):

FI (FamilyId)	Kennung der Familie
FN (FamilyName)	Name der Familie
NP (NumberOfProcesses)	Gesamtanzahl laufender Prozesse
NPS (NumberOfProcessesPause)	Anzahl pausierter Prozesse
NPE (NumberOfProcessError)	Anzahl fehlerhafter Prozesse
sP (EscalationsProcesses)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Prozesse
iPR (MinimumProcessRuntime)	Minimale Prozesslaufzeit
ePR (MeanProcessRunTime)	Durchschnittliche Prozesslaufzeit
aPR (MaximumProcessRunTime)	Maximale Prozesslaufzeit
NA (NumberOfActivities)	Gesamtanzahl aller Aktivitäten laufender Prozesse
NAF (NumberOfActivitiesFinished)	Anzahl beendeter Aktivitäten
NAE (NumberOfActivitiesError)	Anzahl fehlerhafter Aktivitäten
NAS (NumberOfActivitiesPause)	Anzahl pausierter Aktivitäten
NAU (NumberOfActivitiesUnfinished)	Anzahl nicht beendeter Aktivitäten
NAP (NumberOfActivitiesPersonalized)	Anzahl personalisierter Aktivitäten
sA (EscalationsActivities)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Aktivitäten
sF (EscalationsActivitiesFinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) beendeter Aktivitäten
sU (EscalationsActivitiesUnfinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) nicht beendeter Aktivitäten
sAP (EscalationsActivitiesPersonalized)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) personalisierter Aktivitäten

iAR (MinActivityRunTime)	Minimale Aktivitätenlaufzeit
iAF (MinActivityFinishedRunTime)	Minimale Laufzeit beendeter Aktivitäten
iAU (MinActivityUnfinishedRunTime)	Minimale Laufzeit nicht beendeter Aktivitäten
iAP (MinActivityPersonalizedRunTime)	Minimale Laufzeit personalisierter Aktivitäten
eAR (MeanActivityRunTime)	Durchschnittliche Aktivitätenlaufzeit
eAF (MeanActivityFinishedRunTime)	Durchschnittliche Laufzeit beendeter Aktivitäten
eAU (MeanActivityUnfinishedRunTime)	Durchschnittliche Laufzeit nicht beendeter Aktivitäten
eAP (MeanActivityPersonalizedRunTime)	Durchschnittliche Laufzeit personalisierter Aktivitäten
aAR (MaxActivityRunTime)	Maximale Aktivitätenlaufzeit
aAF (MaxActivityFinishedRunTime)	Maximale Laufzeit beendeter Aktivitäten
aAU (MaxActivityUnfinishedRunTime)	Maximale Laufzeit nicht beendeter Aktivitäten
aAP (MaxActivityPersonalizedRunTime)	Maximale Laufzeit personalisierter Aktivitäten

AD Tag (ActivityData), AE Tag (ActivityDataEntry)

Das AD Tag liegt im D Tag, gruppiert alle AE Tags und enthält selbst keine Attribute. Das AD Tag existiert nur wenn dies in der Konfiguration über den „DrillDown“-Typ entsprechend festgelegt ist. Das AE Tag liefert Informationen zu einer Aktivität und hat dazu folgende Attribute (insofern die jeweiligen Optionen in der Reportkonfiguration aktiviert sind):

FI (FamilyId)	Kennung der Familie
FN (FamilyName)	Name der Familie
AI (ActivityId)	Kennung der Aktivität im Modell
AN (ActivityName)	Name der Aktivität
NA (NumberOfActivities)	Gesamtanzahl aller Aktivitäten laufender Prozesse
NAF (NumberOfActivitiesFinished)	Anzahl beendeter Aktivitäten
NAE (NumberOfActivitiesError)	Anzahl fehlerhafter Aktivitäten
NAS (NumberOfActivitiesPause)	Anzahl pausierter Aktivitäten
NAU (NumberOfActivitiesUnfinished)	Anzahl nicht beendeter Aktivitäten
NAP (NumberOfActivitiesPersonalized)	Anzahl personalisierter Aktivitäten
sA (EscalationsActivities)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Aktivitäten
sF (EscalationsActivitiesFinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) beendeter Aktivitäten
sU (EscalationsActivitiesUnfinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) nicht beendeter Aktivitäten
sAP (EscalationsActivitiesPersonalized)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) personalisierter Aktivitäten
iAR (MinActivityRunTime)	Minimale Aktivitätenlaufzeit
iAF (MinActivityFinishedRunTime)	Minimale Laufzeit beendeter Aktivitäten
iAU (MinActivityUnfinishedRunTime)	Minimale Laufzeit nicht beendeter Aktivitäten
iAP (MinActivityPersonalizedRunTime)	Minimale Laufzeit personalisierter Aktivitäten
eAR (MeanActivityRunTime)	Durchschnittliche Aktivitätenlaufzeit
eAF (MeanActivityFinishedRunTime)	Durchschnittliche Laufzeit beendeter Aktivitäten
eAU (MeanActivityUnfinishedRunTime)	Durchschnittliche Laufzeit nicht beendeter Aktivitäten
eAP (MeanActivityPersonalizedRunTime)	Durchschnittliche Laufzeit personalisierter Aktivitäten
aAR (MaxActivityRunTime)	Maximale Aktivitätenlaufzeit
aAF (MaxActivityFinishedRunTime)	Maximale Laufzeit beendeter Aktivitäten

aAU (MaxActivityUnfinishedRunTime)	Maximale Laufzeit nicht beendeter Aktivitäten
aAP (MaxActivityPersonalizedRunTime)	Maximale Laufzeit personalisierter Aktivitäten

Prozessdetailreport

D Tag (Data) im SR Tag (SR/D)

Das D Tag liegt im SR Tag und enthält selbst keine Attribute. Es enthält immer ein Ds Tag und ein P Tag. Abhängig von der Reportkonfiguration kann es auch ein AD Tag enthalten.

Ds Tag (DataFields) im D Tag (SR/D/DS)

Das Ds Tag gruppiert D Tags und enthält selbst keine Attribute. Ein D Tag im Ds Tag ordnet Kennungen von Prozessvariablen, die im Report auftauchen können, Namen zu, die zur Anzeige verwendet werden sollen. Ein D Tag im Ds Tag hat folgende Attribute:

FI (FamilyId)	Kennung der Familie, in der der Name verwendet werden soll
Id	Kennung der Prozessvariable
N (Name)	Name der Prozessvariable zur Anzeige

P Tag (ProcessData), E Tag (ProcessEntry) (SR/D/P/E)

Das P Tag liegt im SR Tag und enthält selbst keine Attribute. Es enthält je ein E Tag pro laufenden Prozess. E Tags können Ds Tags enthalten und haben folgende Attribute (insofern die jeweiligen Optionen in der Reportkonfiguration aktiviert sind):

Id	Kennung des Prozesses
N (Name)	Name des Prozesses
S (Subject)	Betreff des Prozesses
FI (FamilyId)	Kennung der Familie
FN (FamilyName)	Name der Familie
MI (ModelId)	Kennung des Modells
MN (ModelName)	Name des Modells
C (CreationTime)	Zeitstempel der Erstellung des Prozesses
CY (CreationYear)	Jahr der Erstellung des Prozesses
CM (CreationMonth)	Monat der Erstellung des Prozesses
CD (CreationDay)	Tag der Erstellung des Prozesses
CH (CreationHour)	Stunde der Erstellung des Prozesses
CU (CreationMinute)	Minute der Erstellung des Prozesses
CS (CreationSecond)	Sekunde der Erstellung des Prozesses
R (RunTime)	Laufzeit des Prozesses
St (State)	Status des Prozesses: 0x00 unbekannt 0x01 initialisiert 0x02 laufend 0x04 angehalten 0x08 aktiv 0x10 abgebrochen 0x20 beendet 0x40 fehlerhaft, vom System angehalten
NA (NumberOfActivities)	Gesamtanzahl aller Aktivitäten des Prozesses

NAF (NumberOfActivitiesFinished)	Anzahl beendeter Aktivitäten
NAU (NumberOfActivitiesUnfinished)	Anzahl nicht beendeter Aktivitäten
NAP (NumberOfActivitiesPersonalized)	Anzahl personalisierter Aktivitäten
NAS (NumberOfActivitiesPaused)	Anzahl pausierter Aktivitäten
NAE (NumberOfActivitiesError)	Anzahl fehlerhafter Aktivitäten
sP (EscalationsProcess)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) des Prozesses
sA (EscalationsActivity)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) aller Aktivitäten des Prozesses
sU (EscalationsActivitiesUnfinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) nicht beendeter Aktivitäten
sF (EscalationsActivitiesFinished)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) beendeter Aktivitäten
sAP (EscalationsActivitiesPersonalized)	Gesamtanzahl an Eskalationen (abgelaufene Fristen) personalisierter Aktivitäten

Ds Tag (DataFields) im P Tag (SR/D/P/Ds/D)

Das Ds Tag gruppiert D Tags und enthält selbst keine Attribute. Ein D Tag enthält die Werte von Prozessvariablen hat dazu folgendes Attribut

Id	Kennung der Prozessvariable
----	-----------------------------

Das D Tag enthält je nach Typ der Prozessvariablen selbst ein weiteres Tag, welches wiederum den Wert der Prozessvariablen enthält. Folgende Tags sind innerhalb des D Tags erlaubt:

S (String)	Für Zeichenketten
B (Boolean)	Für Wahrheitswerte
L (Long)	Für ganzzahlige Werte
F (Float)	Für Fließkommazahlen
D (Date)	Für Datumsangaben

AD Tag (ActivityDate), AE Tag (ActivityDataEntry) (SR/D/AD/AE)

Das AD Tag liegt im D Tag, gruppiert alle AE Tags und enthält selbst keine Attribute. Das AD Tag existiert nur wenn dies in der Konfiguration über den „DrillDown“-Typ entsprechend festgelegt ist. Das AE Tag liefert Informationen zu einer Prozessaktivität und hat dazu folgende Attribute (insofern die jeweiligen Optionen in der Reportkonfiguration aktiviert sind):

Id (ProcessId)	Kennung des Prozesses
AI (ActivityId)	Kennung der Aktivität
MA (ModelActivityId)	Kennung der Aktivität im Modell
AN (ActivityName)	Name der Aktivität
T (StartTime)	Startzeitpunkt (Zeitstempel)
TY (StartYear)	Jahr des Startzeitpunkts
TM (StartMonth)	Monat des Startzeitpunkts
TD (StartDay)	Tag des Startzeitpunkts
TH (StartHour)	Stunde des Startzeitpunkts
TU (StartMinute)	Minute des Startzeitpunkts
TS (StartSecond)	Sekunde des Startzeitpunkts
R (RunTime)	Laufzeit

S (State)	<p>Status:</p> <p>0x00000000 unbekannt</p> <p>0x00000001 erstellt</p> <p>0x00000002 gestartet</p> <p>0x00000004 StartActivityEvent ausgeführt</p> <p>0x00000008 EndActivityEvent ausgeführt</p> <p>0x00000010 Schleifenbedingung geprüft</p> <p>0x00000020 Schleifenkörper ausführen</p> <p>0x00000040 Arbeitsschritt angelegt</p> <p>0x00000080 Arbeitsschritt personalisiert</p> <p>0x00000100 Warten auf Sperrfrist</p> <p>0x00000200 Warten auf Subflow</p> <p>0x00000400 ausgeführt</p> <p>0x00000800 weitergeleitet</p> <p>0x00001000 beendet</p> <p>0x00002000 angehalten</p> <p>0x00004000 beendet</p> <p>0x10000000 fehlerhaft</p>
E (NumberOfEscalations)	Anzahl Eskalationen (Abgelaufene Fristen)