

@enterprise 8.0

Systemadministration

Dezember 2017

Groiss Informatics GmbH

Groiss Informatics GmbH

Strutzmannstraße 10/4 9020 Klagenfurt Austria

Tel: +43 463 504694 - 0 Fax: +43 463 504594 - 10 Email: support@groiss.com

Dokumentversion 8.0.22989

Copyright © 2001 - 2017 Groiss Informatics GmbH. Alle Rechte vorbehalten.

Die Informationen in diesem Dokument können jederzeit geändert werden. Falls Sie Fehler in der Dokumentation finden, bitte melden Sie diese an uns. Die Groiss Informatics GmbH gibt keine Garantie dafür ab, dass die Dokumente fehlerfrei sind.

Jede Art der Vervielfältigung oder Weitergabe dieser Materialien, ob elektronisch oder mechanisch, ist ohne die explizite schriftliche Erlaubnis der Groiss Informatics GmbH untersagt.

@enterprise ist eine eingetragene Marke der Groiss Informatics GmbH, andere Namen sind teilweise Markenzeichen der jeweiligen Hersteller.

Einleitung

Die vorliegende Dokumentation beschreibt das Workflow–Management–System (WFMS) @enterprise. Die Ziel–Lesergruppe sind Systemadministratoren, die für die Online–Wartung des Systems sowie für die Verwaltung workflowspezifischer Informationen (z.B. Anlegen neuer Organisationseinheiten, Definition von Prozessen etc.) verantwortlich sind.

Das Handbuch für den Systemadministrator besteht aus folgenden Kapiteln:

- Systemarchitektur: Hier wird die Architektur des Systems das Zusammenwirken der einzelnen Komponenten beschrieben.
- **Das HTML–Interface:** Die Administration des Systems erfolgt weitgehend über ein HTML–Interface. Die allgemeine Struktur dieses Interfaces wird in diesem Kapitel beschrieben.
- Ids, Namen und Internationalisierung: Hier wird beschrieben, welche Attribute einer Objektklasse in @enterprise als Id verwendet werden und wie eine Id aussieht. Weiters wird die Internationalisierung von Stammdatenobjekten und Objektklassen beschrieben.
- **Definition der Organisationsstruktur:** Ein Teil der Stammdaten, die für die Ausführung eines Workflows benötigt werden, besteht aus der Aufbauorganisation Ihres Unternehmens: Organisationseinheiten, OE–Bäume, Personen, Rollen dieser Personen, Rechte und Berechtigungslisten von Personen bzw. Rollen. Die folgenden Objektklassen werden dazu in @enterprise verwaltet:
 - Server: Eine @enterprise Installation kann aus mehreren Servern bestehen.
 - **Rollen:** dienen zur Beschreibung von Funktionen und Aufgabengebieten von Personen.
 - Rechte: werden an Rollen bzw. an Benutzer vergeben. Dadurch kann man den Benutzerkreis f
 ür Systemfunktionen einschr
 änken.
 - Berechtigungslisten: Mehrere Rechte können in Berechtigungslisten zusammengefasst und dann gemeinsam an eine Rolle oder einen Benutzer vergeben werden.
 - Benutzer: sind die im System registrierten Personen.

- Organisationseinheiten dienen zur Beschreibung der Struktur Ihrer Organisation. Im weiteren Verlauf dieses Dokumentes wird Organisationseinheit mit OE oder Org.-Einheit abgekürzt.
- Organisationsklassen: dienen der Typisierung von Organisationseinheiten.
- OE-Bäume: Die Organisationseinheiten können Hierarchien bilden, d.h. eine OE kann einer anderen untergeordnet sein und umgekehrt. Die Hierarchie von Organisationseinheiten wird durch das Einhängen der entsprechenden OEs in einen OE-Baum definiert. Dabei kann eine OE mehrere untergeordnete Organisationseinheiten besitzen, eine untergeordnete OE (in einem Baum) aber nur eine übergeordneten OE besitzen. Eine OE kann in mehrere OE-Bäume eingehängt werden (auf diese Weise können abteilungsübergreifende Organisationseinheiten abgebildet werden).
- Berechtigungssystem für @enterprise: Dieses Kapitel beschreibt das Berechtigungssystem von @enterprise, mit dessen Hilfe Zugriffsrechte von Benutzern auf Objekte definiert werden können.
- Workflowelemente: Aufbauend auf der Definition der o.g. Stammdaten können Prozesse (Workflows) definiert werden. Die einzelnen Elemente werden in den folgenden Kapiteln beschrieben:
 - Applikationen: Logisch zusammengehörende Prozesse können zu sog. Applikationen zusammengefasst werden, z.B. *Einkauf, Forschung, Entwicklung, Verkauf, Personal, Kreditabteilung, etc.*
 - Tasks sind die elementaren Bestandteile (Schritte) in Prozessen.
 - Funktionen sind Programmstücke, die automatisierte Teilaufgaben eines Tasks darstellen.
 - Formulare enthalten in strukturierter Form die Daten der Prozesse.
 - Prozesse beschreiben den Ablauf eines Geschäftsprozess als Ganzes.
 - Interfaces erlauben das Starten von Prozessen über ein HTML-Formular.
- **Prozessdefinition:** In diesem Kapitel wird die Definition von Prozessen erläutert, dafür gibt es in @enterprise drei Möglichkeiten:
 - Workflow–Definition–Language (WDL): eine Script-Sprache zur Prozessdefinition,
 - XWDL: die XML-Version von WDL; eine Beschreibung von XWDL finden Sie im XWDL-Handbuch von @enterprise,
 - grafische Prozessdefinition mit dem Prozesseditor von @enterprise.
- Die Suche in @enterprise: Dieses Kapitel verweist auf die entsprechenden Handbücher, in denen die Suche von @enterprise beschrieben ist.
- Administrationsfunktionen: Weitere Administrationsfunktionen, wie Import/Export, Reorganisation, SSL etc. werden in diesem Kapitel beschrieben.
- Konfiguration: In diesem Kapitel wird die Konfiguration des @enterprise-Servers beschrieben.

• **Dashboard:** Hier wird beschrieben, wie das Dashboard von @enterprise verwendet werden kann.

Geschlechtsneutrale Differenzierung: Aus Gründen der einfacheren Lesbarkeit wird auf die geschlechtsneutrale Differenzierung, z.B. Benutzer/innen, verzichtet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung grundsätzlich für beide Geschlechter.

Inhaltsverzeichnis

1	Syst	emarchi	itektur	11
	1.1	Das W	orld Wide Web (WWW)	11
	1.2	Die Sy	stemkomponenten	12
2	Das	HTML-	-Interface	13
	2.1	Tabelle	ndarstellung	15
		2.1.1	Spaltenauswahl, Sortierung und Filter	15
		2.1.2	Standard-Funktionen	17
	2.2	Objekt	-Detailansicht	18
		2.2.1	Registerkarte: Allgemein	19
		2.2.2	Registerkarte: Historie	21
		2.2.3	Registerkarte: Zugriff	21
		2.2.4	Registerkarte: Referenziert von	22
		2.2.5	Weitere Funktionen	23
3	Ids,	Namen	und Internationalisierung	25
	3.1	Ids und		25
	3.2	Interna	tionalisierung der Stammdatenobjekte und Objektklassen	26
4	Defi	nition d	er Organisationsstruktur	27
	4.1	Rollen	~ · · · · · · · · · · · · · · · · · · ·	27
		4.1.1	Registerkarte: Allgemein	27
		4.1.2	Registerkarte: Benutzer	29
		4.1.3	Registerkarte: Berechtigungen	29
		4.1.4	Systemrollen	30
	4.2	Rechte	*	30
		4.2.1	Registerkarte: Allgemein	30
		4.2.2	Registerkarte: Benutzer	31
		4.2.3	Standardrechte	31
	4.3	Benutz	er	33
		4.3.1	Registerkarte: Allgemein	33
		4.3.2	Registerkarte: Rollenzuordnungen	35
		4.3.3	Registerkarte: Vertretungen	36
		4.3.4	Registerkarte: Rollenvertretungen	36
		4.3.5	Registerkarte: Berechtigungen	37

		4.3.6	Registerkarte: Alle Berechtigungen	37
		4.3.7	Registerkarte: Einstellungen	38
	4.4	Organi	sationseinheiten	38
		4.4.1	Registerkarte: Allgemein	38
		4.4.2	Registerkarte: Übergeordnete OE's	40
		4.4.3	Registerkarte: Rollen	40
	4.5	OE-Ba	um	40
		4.5.1	Registerkarte: Allgemein	41
		4.5.2	Registerkarte: OE-Baum	41
		4.5.3	Funktion OE-Bäume zusammenführen	43
	4.6	Organi	sationsklassen	14
		4.6.1	Registerkarte: Allgemein	45
	4.7	Schlag	worte	46
	4.8	Server		46
5	Bere	echtigun	agssystem 4	47
	5.1	Allgen	neines	17
		5.1.1	Berechtigungsliste	1 7
		5.1.2	Standardberechtigungen	48 18
		5.1.3	Objektklassen	18
	_	5.1.4	Formulare und Berechtigung	19
	5.2	Verwei	ndung von Berechtigungen	49
		5.2.1	Berechtigungsdialog	49
		5.2.2	Berechtigungen in der Benutzerverwaltung	52
		5.2.3	Berechtigungen in der Rollenverwaltung	52
		5.2.4	Berechtigungen beim Objekt – Zugriff	53
		5.2.5	Verwaltung von Berechtigungslisten	<i>3</i> 3
		5.2.6	Rechte um Berechtigungen zu bearbeiten	53
		5.2.7	Sonderfall: Berechtigungen für Rollenzuordnungen	<u>,</u>
	5.3	Standa	rdeinstellungen	۶4
	5.4	Wofür	werden welche Rechte benötigt?	۶4
	5.5	Beispie		۶4 - د
	5.6	Berech	tigung und Vertretung)6
6	Wor	kflowel	emente	58
U	61	Applik	ationen	59
	0.1	611	Registerkarte: Allgemein	50
		612	Registerkarte: Figenschaften	50 51
	62	Tasks		52
	0.2	621	Registerkarte: Allgemein	52
		622	Registerkarte: Eskalation	,2 55
		623	Registerkarte: Funktionen	,5 57
		624	Froänzen von Formularen	57 58
	63	Funktia	onen f	50 58
	0.5	631	Registerkarte: Allgemein	,0 59
		632	Standardfunktionen	71
	64	Formul	are	- 1 72
	U. T	. ormu	······································	-

	6.4.1	Neues Formular erstellen
	6.4.2	Formular bearbeiten
	6.4.3	Registerkarte: Allgemein
	6.4.4	Registerkarte: Javaklasse
	6.4.5	Registerkarte: Datenbanktabelle
	6.4.6	Registerkarte: Rechte
	6.4.7	Registerkarte: Standardberechtigungen
	6.4.8	Registerkarte: Vorschau
	6.4.9	Registerkarte: Ordnereinstellungen
6.5	Prozess	se
	6.5.1	Neuen Prozess mit Prozesseditor erstellen
	6.5.2	Prozess mit Prozesseditor bearbeiten
	6.5.3	WDL / XWDL laden
	6.5.4	Prozessübersicht
	6.5.5	Registerkarte: Allgemein
	6.5.6	Registerkarte: Source
	6.5.7	Registerkarte: Graph. Darstellung
	6.5.8	Registerkarte: Komponenten
	6.5.9	Registerkarte: Formular-Sichtbarkeiten
	6.5.10	Registerkarte: Eskalation
	6.5.11	Registerkarte: Funktionen
	6.5.12	Registerkarte: Ordnereinstellungen
6.6	Interfac	ce
	6.6.1	Registerkarte: Allgemein
6.7	Funktio	onsgruppe
6.8	GUI-K	onfiguration
	6.8.1	Registerkarte: GUI-Konfiguration
	6.8.2	Registerkarte: Zuordnungen
6.9	Ressou	rcen Editor
	6.9.1	Toolbarfunktionen
	6.9.2	CSV-Datei Konvertierung 109
6.10	Web-D	ienste
	6.10.1	Webservice Clients
	6.10.2	Webservice Server
Duca		-ition 112
Proz	WDI	
/.1	WDL	Levikalische Konventionen 114
	7.1.1	Prozestkenf 115
	7.1.2	PIOZESSKOPI
	7.1.3	Einforder American and Statement 117
	7.1.4 7.1.5	Emiache Allweisungen
	1.1.3 7.1.6	Konuonsuukuuren 120 Eventmechenismus 120
	/.1.0	Evenumechamismus
7.2	/.1./	he Prozosade finition 121
1.2		Aufbau das Prozessaditors
	1.2.1	Autoau des Prozesseditors
	1.2.2	Die Funktionen der Menuleiste

		7.2.3	Prozesseigenschaften	138
		7.2.4	Tasks	139
		7.2.5	Timeout Task	142
		7.2.6	Eigenschaften einer Aktivität definieren	143
		7.2.7	Zeitmanagement	146
		7.2.8	Die Funktionsleiste	147
		7.2.9	Bedingungen für Ifs, Choice, Schleifen	152
		7.2.10	Die Methode eines Systemschrittes bestimmen	152
		7.2.11	Die Eigenschaften eines Batchschrittes bestimmen 1	152
		7.2.12	Die Eigenschaften eines Subprozesses bestimmen 1	152
		7.2.13	Eigenschaften eines Parallel For definieren	153
		7.2.14	Eigenschaften eines Events definieren	154
		7.2.15	Eigenschaften eines Web-Dienste-Knoten definieren	155
8	Die	Suche ir	n @enterprise 1	157
	8.1	Prozes	ssuche	157
	8.2	Dokun	nentensuche	157
	8.3	Erweite	erte Suche	157
	8.4	Gespei	cherte Anfragen	157
0	۸dn	ninistrat	tionsfunktionen 1	158
1	9 1	Server		158
	7.1	911	Servermonitor	159
		912	Serversteuerung	161
		913	Freignisse	161
		9.1.5	Worklist-Cache	162
		015	Klassennfad	163
		9.1.5	Timer	165
		9.1.0	Objekthistorie	160
		9.1.7		170
		9.1.0		
		9.1.9	Fuent Degistrierungen	
		9.1.10	Zertifiketemenogement	170
		9.1.11		170
		9.1.12		172
		9.1.15		172
		9.1.14		173
	0.2	9.1.1J		174
	9.2		Anmelden de /eltivieren	177
		9.2.1	Perceptigungen überprüfen	170
		9.2.2	Abgeleufene Descuärter	170
	0.2	9.2.3 Imment	Augenaurene Passworter	170 170
	9.5	import	/Export	179 170
		9.3.1	Import/Export in AIVIL FORMAT	1/9
		9.5.2	Prozesse archivieren	103
		9.3.3	Applikation installieren	183
	0.4	9.3.4 D		183
	9.4	Keorga	inisation	186

	9.4.1	Rollenzuordnungen verändern
	9.4.2	Prozessinstanzen analysieren
	9.4.3	OE-Historie
9.5	5 Komm	unikation
	9.5.1	Mailboxen
	9.5.2	LDAP
	9.5.3	BatchJobs
	9.5.4	WfXML
	9.5.5	Web Dienste
10 Ko	onfigurati	on 193
11 Da	shboard	194
11	.1 Neu .	
11	.2 Öffner	195
11	.3 Speich	ern
11	.4 Speich	ern unter
11	.5 Lösche	en
12 Ad	lministra	tion via Kommandozeile 197
12	.1 Archit	ektur und Anwendung
12	.2 Komm	andos
	12.2.1	Client Kommandos
	12.2.2	Server Kommandos
12	.3 Beispi	ele
	12.3.1	Einen Konfigurationsparameter setzen
	12.3.2	Serverneustart
	12.3.3	Rolle einem Benutzer hinzufügen bzw. entziehen
	12.3.4	Zeitintervall eines Timers setzen
	12.3.5	Arbeitskorb Handling
	12.3.6	Session Handling
13 Pr	ozess-Co	ckpit 202
	13.0.1	Konfiguration
	13.0.2	Berechtigungen

1 Systemarchitektur

Das Workflowsystem @enterprise ist vollständig webbasiert. Deshalb werden zuerst kurz die wesentlichen Konzepte dieser Technologie genannt und anschließend die Architektur des Systems beschrieben.

1.1 Das World Wide Web (WWW)

Das WWW beruht auf drei Grundkonzepten:

- die einheitliche Adressierung von Informationen im Internet mit dem Uniform Resource Locator (URL),
- der Darstellung von Information mit der Sprache HTML und
- der Übertragung über das HTTP Protokoll.

HTML (Hyper Text Markup Language) bietet die Möglichkeit, unterschiedliche Medientypen, wie Texte, Grafiken, Ton und Video, in einem Dokument zu vereinen. Mittels so genannter Hyperlinks kann mit Mausklick von einem Dokument zu einem anderen verzweigt werden - komplex strukturierte Informationen können so übersichtlich präsentiert werden. Wichtig für die Interaktion mit dem WWW ist die Möglichkeit, ausfüllbare Formulare zu repräsentieren. Sie erlauben dem Benutzer, Daten an den Server zu schicken. Eine HTML-Erweiterung erlaubt auch das Laden von Dateien auf den Server.

HTTP (Hyper Text Transfer Protocol) ist ein einfaches Übertragungsprotokoll von Dokumenten über das Internet. Der Client (Browser) fordert ein Dokument bei einem Server an, indem eine TCP/IP Verbindung zu diesem Server aufgebaut wird und der String *GET* '*Dokument-URL*' übertragen wird. Der Server antwortet mit einigen Zeilen Statusinformation und dem Inhalt des Dokuments. Dann schließt er die Verbindung. Die Dokument-URL kann auch ein Programmaufruf am Server sein, zurückgeschickt wird dann die Ausgabe des Programms. Große Bedeutung erlangte auch das Prinzip von mobilem Code. Dabei werden in HTML Dokumenten kleine Programme, so genannte Applets, eingebettet, die dann am Client ausgeführt werden und somit die Funktionalität des Clients erhöhen. Als Programmiersprache wird dabei vor allem Java eingesetzt; die meisten Browser erlauben die Ausführung von vorkompiliertem Java-Code.

1.2 Die Systemkomponenten

Abbildung 1.1 zeigt den Aufbau des Workflow–Systems @enterprise mit seinen Hauptkomponenten:



Abbildung 1.1: @enterprise Systemarchitektur

Das System basiert vollständig auf dem World Wide Web. Jegliche Kommunikation sowohl des normalen Benutzers als auch des Systemadministrators mit dem System kann über einen Web–Browser durchgeführt werden.

Der HTTP–Server ist dabei die Schnittstelle des Workflowsystems nach außen. Anfragen von einem Client werden in Methodenaufrufe übersetzt und an das Workflow–Managementsystem weitergeleitet. Dieses besteht im wesentlichen aus zwei Komponenten: Der Engine und dem HTML Interface.

Die Workflow-Engine selbst ist eine Sammlung von Methoden, die die eigentliche Funktionalität des Workflow-Systems, wie z.B. Verändern von Formularen, Beenden einer Aktivität, usw. bereitstellt.

Das HTML–Interface interpretiert die Anfrage vom Browser, führt einen API–Call auf die Workflow–Engine durch und übersetzt das Ergebnis wieder in eine am Browser darstellbare HTML-Seite.

Das Datenbankmanagementsystem (DBMS) speichert alle Stammdaten und für die Prozessausführung relevanten Informationen. Dies sind die Prozessdefinitionen, die Stati der laufenden Prozesse, sowie alle Benutzerdaten, die in Formularen abgelegt werden.

Weitere Details zur Systemarchitektur entnehmen Sie dem Handbuch Applikationsprogrammierung mit @enterprise.

2 Das HTML–Interface

Sie gelangen in die @enterprise -Systemadministration in dem Sie im Menü @enterprise den Menüpunkt Administration auswählen. Diese Funktion steht Ihnen nur zur Verfügung, wenn Sie das Recht admin besitzen.

Wenn Sie sich als Standard-Systemadministrator (Benutzer: *sysadm*) anmelden, gelangen Sie direkt in die @**enterprise** -Systemadministration.

🥹 genterprise-Administration - Mozilla Firefox								
Datei Bearbeiten Ansicht Chronik	Lesezeichen Extras	Hilfe						
📸 @enterprise-Administration	+							~
+> N ttp://localhos	t:8380/wf/servlet.metho	d/com.dec.avw.html.HTML	Gui.showAdmin	☆ -) 🔞	C 🚼 - Google		P	
enter-	Administration				ep80 (oracle)	- angemeldet:	Markus Irra	sch
() prise	📑 🕑 🤰	/ × I		2 🔒 🔎 🗹			1 1	<u>8</u>
Organisation								
Applikationen	Formulare							
🗄 Default	‡ Id	Name	Version	Beschreibung	Тур	Basisformular	Maskentyp	66
🕀 demos	itsm change	Change	1		Prozessformular		XHTML	
⊡-ITSM	itsm changereq	Change Request	1		Prozessformular		XFORMS	
Prozesse	itsm incident	Incident	1		Prozessformular		XHTML	
Estructure	itsm_problem	Problem	1		Prozessformular		XHTML	
Formulare	itsm_product	Product	1		Prozessformular		XHTML	
lasks	itsm_release	Release	1		Prozessformular		XHTML	
Funktionen	itsm_value	Value	1		Prozessformular		HTML	
····Rollen	Anzahl der Einträge: 7	0 ausgewähit						
Rechte								
···· Objektklassen								
Funktionsgruppen								
GUI-Konfigurationen								
Bessourcen								
Webservice Crients								
Webservice Server								
unit_testApplication								
t≝ Kel-Applikation								
🖭 lexisnexis								
I ⊕ Personalprozesse								
Projektmanagement								
🖲 ResourceEditorTest2de								
Suche								
Admin-Tasks								
Konfiguration								

Abbildung 2.1: Die Systemadministration

Das HTML-Interface (siehe Abbildung 2.1) gliedert sich in die folgenden Bereiche:

- 1. **Information:** Im oberen Bereich der @**enterprise** -Systemadministration sehen Sie Informationen zum angemeldeten Benutzer und dem aktuell laufenden Server.
- 2. Werkzeugleiste: Direkt unter dem Informationsbereich befindet sich die Werkzeugleiste, die Ihnen zu den jeweils im Bearbeitungsbereich angezeigten Informationen verschiedene Funktionen zur Verfügung stellt. Die vier Funktionen *Hilfe*, *Dashboard*, *Arbeitskorb* und *Abmelden*, rechts außen, stehen Ihnen immer zur Verfügung.

171

A

- Hilfe: Wenn Sie diese Funktion aktivieren, öffnet sich in einem neuen Fenster die @enterprise-Hilfe.
- **Dashboard:** Nach dem Aktivieren dieser Funktion wird im Bearbeitungsbereich Ihr Dashboard angezeigt (siehe Kapitel 11).
- Arbeitskorb: Durch das Anklicken dieser Funktion gelangen Sie in die @enterprise - Arbeitskorbkomponente.
- Abmelden: Wenn Sie diese Funktion aktivieren, melden Sie sich von @enterprise ab.
- Hinweis: Wenn dieses Symbol erscheint, wurde eine Änderung am @enterprise -System vorgenommen. Sie erhalten durch Anklicken dieses Symbols nähere Informationen darüber, ob Sie entweder den Server neu starten oder die Cachestrukturen aktualisieren müssen.
- Information: Wenn dieses Symbol erscheint, sind Neuigkeiten vorhanden. Durch einen Klick auf dieses Symbol wird ein Popup geöffnet, in dem die Neuigkeiten zu sehen sind.
- 3. **Navigation:** Dieser Bereich enthält eine Auflistung der zur Verfügung stehenden Administrationsfunktionen. Diese Funktionen sind in verschiedene Bereiche - *Organisation, Applikationen, Suche, Admin-Tasks* und *Konfiguration* - gegliedert. Sie öffnen einen Bereich, indem Sie die jeweilige Schaltfläche aktivieren.
 - **Organisation:** In diesem Bereich stehen Ihnen Links für folgende Verwaltungsaufgaben zur Verfügung: *Benutzer*, *Organisationseinheiten*, *Organisationsklassen*, *OE-Baum*, *Server*, *Berechtigungsliste*, *Interface* und *Schlagworte*.
 - Applikationen: Hier werden in einzelnen Unterordnern die jeweiligen @enterprise -Applikationen aufgelistet. Pro Applikation werden folgende Bereiche zur Verwaltung angeboten: *Prozesse, Formulare, Tasks, Funktionen, Rollen, Rechte, Objektklassen, Funktionsgruppe* und *GUI-Konfiguration*. Die Übersicht über alle Applikationen wird als oberster Link in diesem Bereich dargestellt.
 - Suche: In @enterprise können Sie auf unterschiedliche Arten nach Prozessinstanzen bzw. DMS-Objekten suchen. Folgende Möglichkeiten werden in diesem Bereich angeboten: *Prozesssuche*, *Dokumentensuche*, *Erweiterte Suche*, *Gespeicherte Anfragen*.
 - Admin-Tasks: Dieser Bereich enthält alle Administrationsfunktionen. Eine genaue Auflistung und Beschreibung der einzelnen Links finden Sie in einem späteren Kapitel in diesem Handbuch.

- **Konfiguration:** Alle Funktionen zur Konfiguration Ihrer Installation finden Sie in diesem Bereich. Eine genaue Beschreibung dieser Funktionen finden Sie im Installationshandbuch.
- 4. Bearbeitung: Der Hauptbereich der @enterprise -Systemadministration steht Ihnen zur Bearbeitung der ausgewählten Funktion zur Verfügung. Hier sehen Sie unterschiedliche Masken bzw. Tabellen, deren Inhalt Sie mit den Funktionen der Werkzeugleiste bearbeiten können. Nach dem Öffnen der Systemadministration wird in diesem Bereich Ihr Dashboard angezeigt.

2.1 Tabellendarstellung

Stammdaten werden in @enterprise zunächst in einer Tabelle aufgelistet. Diese Tabelle enthält in den Zeilen die einzelne Objekte des ausgewählten Bereichs und in den Spalten verschiedene Eigenschaften des Objektes.

Detailinformationen zum Objekt und weitere Funktionen zum Objekt (Historie, Zugriff etc.) werden Ihnen in einem eigenen Fenster angezeigt (siehe Kapitel 2.2). Sie öffnen dieses Fenster indem Sie auf einem Eintrag einen Doppelklick ausführen oder eine Zeile zunächst mit einem einfachen Klick selektieren und danach die Menüfunktion *Bearbeiten* aktivieren.

Bei der Anzeige der Tabelle wird, falls die Anzahl der gefundenen Einträge eine Grenze übersteigt, nachgefragt, ob wirklich alle Einträge dieser Tabelle angezeigt werden sollen. Diese Obergrenze ist in der Systemkonfiguration einstellbar (siehe Installationshandbuch).

In der Tabellendarstellung werden die folgenden Zeilenformatierungen verwendet:

- Zuletzt bearbeitet: Jener Eintrag, den Sie zuletzt bearbeitet haben, wird farblich hinterlegt dargestellt.
- **Inaktive Einträge:** Objekte, die inaktiv sind, werden in grauer Schrift und *kursiv* dargestellt. Beim Objekt **Formulare** werden zusätzlich zu den inaktiven Formularen, auch Einträge als inaktiv markiert, bei denen die Formularklasse nicht geladen werden kann.
- Selektierte Einträge: Einträge, die Sie aktuell selektiert haben, werden farblich hinterlegt.

2.1.1 Spaltenauswahl, Sortierung und Filter

Sie können die Anzahl der Spalten beliebig ändern. Dazu steht Ihnen ganz rechts außen im Tabellenkopf die Spaltenauswahl zur Verfügung. Klicken Sie auf die Funktion und es öffnet sich direkt bei dieser Spalte ein Dialog, in dem alle auswählbaren Spalten aufgelistet werden.

Jene Spalten, die bereits angezeigt werden, werden mit einen Häkchen dargestellt. Um eine zusätzliche Spalte anzuzeigen, wählen Sie aus dieser Liste einen Namen (ohne Häkchen) aus. Die Tabelle wird nun aktualisiert und die zusätzliche Spalte angezeigt. Um eine bestimmte Spalte auszublenden, wählen Sie in der Spaltenauswahl diese Spalte (mit Häkchen)

2.1. TABELLENDARSTELLUNG

🥹 @enterprise-Administratior	- Mozilla Firefox				
Datei Bearbeiten Ansicht Chro	nik Lesezeichen Extras Hilf	9			
% @enterprise-Administration	+				
💪 👌 💦 🕅 http://loca	lbost:8380/wf/servlet_method/com	dec avvu btrol HTMI Gui showAdmin 🛛 🔶 👻 🎯 🌈 🚺	Coogle		
• • • • • • • • • • • • • • • • • • •	in rescreeced in your vice internet pay com		doogie		
enter	Administration	e p 81	0 (oracle) - angemeldet	: Markus Irra	sc
(prise	1 6 m l				
	* 2 × I	2 2 2 🗹		U 11 1	
Organisation					
Applikationen	Rollen				
P Default	▲ ‡Id	Name	Beschreibung	Тур	
Prozesse	admin	admin		lokal	
Formulare	all	Alle		alobal	
Teeke	B SatzGrafik	Satz/Grafik		lokal	
Tasks	betreuer	Betreuer		lokal	
Funktionen	dept	OE		lokal	
Rollen	docuRole	Rollenzuordnungen bearbeiten in OE Service		lokal	
····Rechte	globaldumm	globaldumm		global	
- Objektklassen	home	Home		lokal	
Funktionsgruppen	hugo	hugo		lokal	
GUI-Konfigurationen	leiter	Abteilungsleiter		lokal	
Boooguroop	Lektorat	Lektorat		lokal	
Ressourcen	Lektoratsassistenz	Lektoratsassistenz		lokal	
vvebservice Clients	manager	manager		hierarchisch	
·····Webservice Server	PEM	Produktion Elektronische Medien		lokal	
demos	process_agent	Prozessakteur		global	
HTSM	Programmplanung	Programmplanung		lokal	
Prozesse	public_stat	Öffentliche Statistiken		global	
Formulare	sekretaer	Sekretär		lokal	
Tacke	SM	Sourcemanagement		lokal	
Funktionen	sys	Sys		global	
Funktionen	testrolle	testrolle		lokal	
Rollen	Verlagsleitung	Verlagsleitung		lokal	
Rechte	XSLTK	XSLT-K		lokal	
Objektklassen	Z	Z		lokal	
Funktionsgruppen	ZDK	ZDK		lokal	
GUI-Konfigurationen	ZDK_QS	ZDK-QS		lokal	
	ZDKA	ZUK-A		lokal	
Suche	Anzahl der Einträge: 27 0 au	isgewanit			
Admin-Tasks	-				
Konfiguration					

Abbildung 2.2: Beispiel für Tabellendarstellung (Rollen)

aus.

Durch Aktivieren eines einzelnen Spaltenkopfs ändern Sie die Sortierreihenfolge der Tabelle. Nach welcher Spalte und in welche Richtung sortiert wird, wird durch einen Pfeil links neben dem Spaltennamen dargestellt.

Der Link *Filter* soll Ihnen bei vielen Einträgen in Ihrer Tabelle helfen, die Übersicht nicht zu verlieren. Der Filter stellt eine Art Auswahlkriterium dar. Damit können Sie gewisse Einträge der Tabelle ausblenden.

Durch einen Klick in der entsprechenden Spaltenüberschrift der Tabelle wird ein kontextabhängiges Filtermenü mit folgenden Einträgen angezeigt.

- Aufsteigend sortieren: Die Einträge der Tabelle werden nach der aktuellen Spalte aufsteigend sortiert. Dies wird in der entsprechenden Spalte durch einen Pfeil nach oben angezeigt.
- Absteigend sortieren: Die Einträge der Tabelle werden nach der aktuellen Spalte absteigend sortiert. Dies wird in der entsprechenden Spalte durch einen Pfeil nach unten angezeigt.
- Alle Einträge: Die Anwendung des Spaltenfilters der aktuellen Spalte wird aufgeho-

ben.

- Benutzerdefiniert: Durch das Auswählen dieses Menüpunktes wird eine HTML–Seite angezeigt, in der Sie einen Wert eintragen können. Bestätigen Sie in dieser Seite Ihre Eingaben mit der Schaltfläche *OK*, so wird die aktuelle Tabelle nach dem eingegebenem Wert gefiltert.
- Die ersten 20 unterschiedlichen Spalteninhalte: Wenn Sie einen dieser Werte auswählen, wird die aktuelle Tabelle nach diesem Wert gefiltert.

Sie haben auf die erste Art einen Filter auf eine oder mehrere Spalten definiert. Wenn Sie diese Filterkombination abspeichern möchten, klicken Sie auf den Link *Filter* in der Überschrift des Arbeitsbereiches. Es wird das Filter–Menü mit folgenden Einträgen angezeigt:

- Filter speichern: Dadurch speichern Sie die gerade aktuelle Kombination der einzelnen Spaltenfilter unter einem von Ihnen zuvor festgelegtem Namen mit einer entsprechenden Beschreibung.
- Filter löschen: Dadurch löschen Sie den gerade aktiven, abgespeicherten Filter.
- Alle Einträge: Die Anwendung des abgespeicherten Filters wird aufgehoben.
- Eine Liste aller gespeicherten Filter: Wenn Sie einen dieser Werte auswählen, wird die aktuelle Tabelle nach diesem Filter gefiltert. Die Liste kann auch Filter enthalten, die nicht von Ihnen sondern vom Systemadministrator definiert wurden. Diese Filter können Sie lediglich verwenden, aber nicht löschen.

Hinweis: Achtung! Das Löschen eines Filters kann nicht mehr rückgängig gemacht werden.

2.1.2 Standard-Funktionen

Folgende Funktionen finden Sie bei den meisten Tabellen der Systemadministration.

- Neu: Wenn Sie diese Funktion aktivieren, öffnet sich eine leere Detailansicht. Nach Eingabe aller relevanten Daten wird ein neues Objekt angelegt.
- **Bearbeiten:** Diese Funktion öffnet die Detailansicht zum ausgewählten Objekt. Sie können diese Funktion auf ein markiertes Objekt anwenden. Der Doppelklick auf einen Tabelleneintrag führt ebenfalls diese Funktion aus.
- Löschen: Mit Hilfe dieser Funktion können Sie die ausgewählten Objekte löschen. Sie können die Funktion auf ein oder mehrere Objekte gleichzeitig anwenden.
- Ansehen: Diese Funktion öffnet die Objekt-Detailansicht readonly, d.h. Sie können die Objektdetails nicht verändern. Ausnahme: Bei den Formularen öffnet sich die Formularvorschau und bei Prozessen eine Übersicht der Prozessdefinition nachdem Sie diese Funktion aktiviert haben.





അ



2.2 *Objekt-Detailansicht*

Nach dem Aktivieren der Funktion *Bearbeiten* oder *Neu* bzw. einem Doppelklick auf ein Objekt, öffnet sich ein neues Fenster mit der Detailansicht des ausgewählten Objekts. Diese Detailansicht enthält verschiedene Registerkarten, die unterschiedliche Informationen bzw. Funktionen zum Objekt beinhalten (siehe Abbildung 2.3).

Unabhängig von der gerade aktiven Registerkarte stehen Ihnen immer die Schaltflächen **Ok** und **Abbrechen** zur Verfügung. Durch das Aktivieren der Schaltfläche **Abbrechen** wird die Detailansicht des Objekts ohne Änderungen zu speichern geschlossen. Wenn Sie die Schaltfläche **Ok** aktivieren und Änderungen vorgenommen haben, werden diese zunächst gespeichert und danach die Detailansicht geschlossen. Änderungen können Sie nur in Registerkarten mit Masken vornehmen (z.B. *Allgemein*). Hier steht Ihnen zusätzlich die Schaltfläche **Übernehmen** zur Verfügung.

Neben Masken können auch Tabellen in einzelnen Registerkarten dargestellt sein. Sie können diese Tabellen, wie in Kapitel 2.1 beschrieben, bearbeiten.

Felder, die Sie unbedingt ausfüllen müssen, um ein erfolgreiches Speichern durchführen zu können (Mussfelder), werden in @enterprise fett dargestellt. Wenn Sie eines oder mehrere dieser Mussfelder inkorrekt ausgefüllt haben, werden diese nach dem Aktivieren der Schaltfläche *Ok*, *Übernehmen* oder einem *Registerkartenwechsel* rot markiert. Eine Speicherung wird erst erfolgreich ausgeführt, wenn alle Mussfelder korrekt befüllt sind.

🤌 Rollen: Home	(Default) - @en	terprise - M	lozilla Fire	efox			Rollen: Home (Default) - @enterprise - Mozilla Firefox								
📸 http://localhos	st:8380/wf/servlet.n	nethod/com.gr	oiss.storegui	. Tabbed Wir	ndow.showDial	og?node=admii	n.role&foreignKey=applica 🏹	<u>_</u>							
Allgemein E	Berechtigungen	Benutzer	Historie	Zugriff	Referenzier	t von									
ld:	home														
Name:	home						I18n: Home								
Applikation:	Default		~												
Тур:	lokal 🔽														
Beschreibung	:														
D ()															
Referenzrolle:							*								
Aktiv															
Änderung duro	chführen am:														
Löschen				Ok		Abbreche	n Übernehme	n							

Abbildung 2.3: Beispiel für Objekt-Detailansicht (Rollen)

2.2.1 Registerkarte: Allgemein

Die erste Registerkarte *Allgemein* enthält immer die eigentlichen Objekteigenschaften. Nach dem Sie Änderungen in der Maske durchgeführt haben, können Sie diese auf verschiedene Weise speichern:

- Ok: Aktivieren Sie diese Schaltfläche, wenn Sie die Änderungen abspeichern und anschließend die Detailansicht schließen wollen.
- Übernehmen: Wenn nach dem Speichern weiterhin die Detailansicht geöffnet bleiben soll, aktivieren Sie diese Schaltfläche.
- **Registerkartenwechsel**: Wenn Sie nach dem Speichern auf eine andere Registerkarte der Detailansicht wechseln wollen, aktivieren Sie einfach die gewünschte Karte. Sie werden (bei Änderungen) gefragt, ob Sie die Registerkarte wirklich wechseln und gleichzeitig Ihre Änderungen speichern oder in der aktuellen Ansicht verbleiben wollen. Beenden Sie diesen Dialog mit *Ok*, so werden Ihre Änderungen gespeichert und die Ansicht wechselt zur ausgewählten Karte. Beenden Sie diesen Dialog mit *Abbrechen*, werden Ihre Änderungen NICHT gespeichert und die Ansicht bleibt in der aktuellen Karte.

In der ersten Karte ist außerdem die Schaltfläche **Löschen** aktiv. Mit dieser Funktion können Sie das aktuelle Objekt löschen. Danach wird die Detailansicht geschlossen. Diese Funktion entspricht der Standard-Funktion *Löschen* der Werkzeugleiste.

Änderungen zu späterem Zeitpunkt

Bei manchen Stammdaten– bzw. Workflow–Objekten besteht die Möglichkeit, die durchgeführten Änderungen erst zu einem bestimmten Zeitpunkt wirksam werden zu lassen. Dazu gibt es in der jeweiligen Änderungsmaske das Textfeld **Änderungen durchführen am** (siehe Abbildung 2.3).

Tragen Sie in dieses Textfeld jenes Datum (und Uhrzeit) ein, an dem die vorgenommenen Änderungen wirksam werden sollen und speichern Sie Ihre Eingabe ab. Bis zu diesem Datum enthält das entsprechende Objekt jene Werte in den Attributfeldern, die vor den Änderungen darin enthalten waren. Wird hier kein Datum eingetragen, so werden die Änderungen sofort übernommen. Bei der Eingabe des Datums unterstützt Sie der Kalender, der sich nach dem Aktivieren der Kalenderfunktion öffnet.

Sie können für ein Objekt immer nur eine Änderung zu einem späteren Zeitpunkt eintragen. Ob eine ausstehende Änderung eingetragen ist, erkennen Sie in der Maske am befüllten Textfeld **Objektänderung am** und an der nebenbei platzierten Funktion **Spätere Objektänderung anzeigen**. Wenn Sie diese Funktion aktivieren, öffnet sich ein neues Fenster, welches die ausstehende Objektänderung darstellt (siehe Abbildung 2.4). Wenn Sie hier die Schaltfläche **Änderungen verwerfen** aktivieren, wird die ausstehende Objektänderung verworfen.

Objekt aktivieren / deaktivieren

Bei manchen Stammdaten– bzw. Workflow–Objekten besteht die Möglichkeit diese zu aktivieren oder zu deaktivieren. Dazu gibt es in der jeweiligen Änderungsmaske die Checkbox *Aktiv*.

Solange diese Checkbox beim entsprechenden Objekt nicht angeklickt ist, kann es von den Benutzern des Workflow–Systems nicht verwendet werden und scheint auch bei den Auswahllisten im Arbeitskorb (z.B. bei Bearbeiter ändern) nicht auf. Nur wenn diese Checkbox angeklickt wird, ist das Objekt für alle Workflowbenutzer verwendbar.

Deaktivierte Objekte werden in der Tabellendarstellung in grauer und kursiver Schrift dargestellt.

Internationalisierung

Bei applikationsabhängigen Objekten kann der Name in die zur Verfügung stehenden Sprachen übersetzt werden.

Neben dem Feld *Name* wird bei solchen Objekten der für die aktuelle Sprache internationalisierte Name als Link angezeigt. Wenn Sie auf diesen Link klicken, wird die Internationalisierung für alle zur Verfügung stehenden Sprachen angezeigt. Durch das Aktivieren der

æ

20

Mozilla Firefo>	(_ D
http://localhost	http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDeferredChange?object=com.dec.avw.core.User:4297 🟠						
Durchführungs eingefügt am: eingefügt von:	zeitpunkt:	11-07-2011 0 08-07-2011 0 Markus Irraso	8:00 8:44 ch]]			
						Änderung verwerfen	Abbrechen
ld:	maier						
Nachname:	Maier						
Vorname:	Franz						
Titel:							
Beschreibung:							
E-Mail:							
Telefon-Nr.:							
Server:	ep80 (orac	cle) 💌					
Sprache:				~			
Aktiv							
Sortierattribut:							
Passwort:							
Datum der letz	ten Passwi	ort Änderung:					
Passwort-Poli	ey:		Passwor	t läuft nie ab			
			Muss Pa	asswort beim i	nächsten Log	gin ändern	
			🔄 Kann Pa	isswort nicht ä	ndern		

Abbildung 2.4: Änderungen zu späteren Zeitpunkt (Detail)

Schaltfläche *Schließen* wird diese Anzeige wieder geschlossen. Wie Sie die Internationalisierung verändern können, ist in Kapitel 3.2 genauer beschrieben.

2.2.2 Registerkarte: Historie

Wenn Sie in der Detailansicht eines Objekts auf die Registerkarte *Historie* wechseln, wird eine Tabelle angezeigt, die alle am Objekt durchgeführten Änderungen auflistet (siehe Abbildung 2.5). In dieser Tabelle ist ersichtlich, wer, wann, was mit dem Objekt gemacht hat. Sie können jede einzelne dieser Änderungen im Detail ansehen (Funktion **Anzeigen**).

2.2.3 Registerkarte: Zugriff

Die Registerkarte **Zugriff** der Objekt-Detailansicht zeigt Ihnen, wer, welchen Zugriff direkt oder über eine Berechtigungsliste auf das aktuelle Objekt hat (siehe Abbildung 2.6). Sie können diesen Zugriff beliebig ändern (siehe Kapitel 5).

Historie							
Änderungsmodus	Akteur	Ånderungszeitpunkt	Eô				
🛨 insert	Irrasch Markus markus	08-07-2011 09:19					
ᢞ update	Irrasch Markus markus	08-07-2011 09:20					
ᢞ update	Irrasch Markus markus	08-07-2011 09:23					
ᢞ update	Eisenberg Roland eisenberg	08-07-2011 09:23					
Anzahl der Einträge: 4 0 ausgewählt							

Abbildung 2.5: Registerkarte: Historie

http://localhost:8380/	wf/servlet.me	ethod/com.groiss.s	toregui.TabbedWindow.sl	nowDialog?noo	de=admin.us	ser&func=edit&comingF	rom=%2Fwf%2Fservlet.me	ethod%2Fco	m.groiss.g	jui.table.T	at 🕤
jemein Rollenzuo	rdnungen	Vertretungen	Rollenvertretungen	Historie	Zugriff	Berechtigungen	Alle Berechtigungen	Einstellu	ingen 🔅	Zusatzin	fo
ugriff											
Akteur		Gilt nur, wenn	Rolle in OE		Re	cht	Anwenden i	n OE	Zugi	iff 🛤	ί.
				Berechtigu	ngen bear	beiten			٠		1
/dit_role				Berechtigu	ngen bear	beiten			•		
teilungsleiter				Berechtigu	ngen bear	beiten			٠		
e				Berechtigu	ngen bear	beiten			٠		
ed				Berechtigu	ngen bear	beiten			٠		
ugriff über Bere	echtigung	jsliste									
ugriff über Bere erechtigungsliste:	echtigung acllist	jsliste 💌									
ugriff über Bere lerechtigungsliste: ugeordnete Rechte:	echtigung acllist Zugrit	jsliste ✓ ff über Berec	htigungsliste							٩	
ugriff über Bere lerechtigungsliste: ugeordnete Rechte:	echtigung acllist Zugrit	jsliste ▼ ff über Berec ĈAkteur	:htigungsliste	Gilt n	ur, wenn	Rolle in OE	Reci	ht	E.	٩	
ugriff über Bere lerechtigungsliste: ugeordnete Rechte:	acliist Zugrit	jsliste ♥ ffüber Berec ĈAkteur	:htigungsliste	Gilt n	ur, wenn	Rolle in OE	Recl Objekte bearbeite	ht 1	Eż	٢	
ugriff über Bere lerechtigungsliste: ugeordnete Rechte:	e chtigung acllist Zugri Alle Irrasch M Anzehl der	jsliste ▼ ff über Berec ĈAkteur Aarkus markus Eintrăge: 210 aus	:htigungsliste	Gilt n	ur, wenn	Rolle in OE	Rect Objekte bearbeiter Objekte ansehen	ht 1	Eú:	٩	

Abbildung 2.6: Registerkarte: Zugriff

2.2.4 Registerkarte: Referenziert von

drilldown.png

Wenn Sie die Registerkarte **Referenziert von** ausgewählt haben, wird Ihnen eine Übersicht über alle Objekte gezeigt, welche auf das aktuelle Objekt referenzieren (siehe Abbildung 2.7). Die Objekte werden hierbei hierarchisch dargestellt. Die dazugehörigen Symbole werden folgend beschrieben:

• *Plus-Zeichen:* dieses Objekt hat eines oder mehrere untergeordnete Objekte, die jedoch noch nicht angezeigt werden. Wenn Sie auf dieses Plus–Zeichen klicken, wird dieses Objekt an dieser Stelle erweitert und die direkt untergeordneten Objekte werden angezeigt. Gleichzeitig wird aus dem Plus–Zeichen ein Minus–Zeichen.



Abbildung 2.7: Registerkarte: Referenziert von (Rollen)

2.2.5 Weitere Funktionen

In diesem Abschnitt werden jene Funktionen beschrieben, die sich in den Masken ständig wiederholen und im jeweiligen Kontext dieselbe Funktion besitzen.

• Auswählen: Durch das Aktivieren der Funktion Auswählen neben dem zugehörigen Feld öffnet sich ein HTML-Auswahldialog, mit dem Sie ein Objekt für dieses Feld auswählen können.

Beispiel: Auswahl eines Benutzers, einer Organisationseinheit, etc.

• Entfernen: Durch das Aktivieren der Funktion *Entfernen* neben dem zugehörigen Feld wird der Eintrag in diesem Feld entfernt. Diese Funktion steht Ihnen immer zusammen mit der Funktion *Auswählen* zur Verfügung.

Ŧ

(🖌

Seit @enterprise 8.0 werden neben bzw. anstatt den Funktionen *Auswählen* und *Ent-fernen* Dropdown Listen (Ajax) verwendet. Diese können durch Anklicken dieses Symbols aufgeklappt werden. Anschließend können Sie das gewünschte Objekt auswählen.

- **Kalender:** Wenn Sie diese Funktion aktivieren, öffnet sich ein Kalender, der Sie dabei unterstützt ein Datum einzufügen. Eine detaillierte Beschreibung zum Kalender finden Sie im Benutzerhandbuch.
- Klassenpfad-Checker: Mit dieser Funktionen können Sie die im zugehörigen Textfeld eingegebene URL überprüfen. Hierbei wird überprüft, ob die Klasse vorhanden ist. Weiters wird auch auf das Vorhandensein der Methode und ihrer korrekten Signatur überprüft. Nur in gewissen Fällen wird überprüft, ob die Klasse das geforderte Interface implementiert (z.B. Logger-Klasse muss das Interface com.groiss.log.ILogger implementieren). Das Symbol für den Klassenpfad-Checker wird grün dargestellt, wenn die angegebene URL im Klassenpfad gefunden werden kann. Ansonsten wird dieser rot angezeigt.

3 Ids, Namen und Internationalisierung

3.1 Ids und Namen

In **@enterprise** wird jedes Stammdatenobjekt durch einen eindeutigen Identifier gekennzeichnet. Abhängig von der Objektklasse werden folgende Attribute als Identifier verwendet:

- die Id
- der Name
- sowohl die Id als auch der Name
- eine Kombination von Id und Version
- eine Kombination von Name und Version

Für welche Objektklasse in @**enterprise** welcher/welche Identifier verwendet wird/werden, sehen Sie in der Tabelle 3.1.

Der Identifier ist innerhalb eines @enterprise-Servers eindeutig, und ist auch über alle Applikationen hinweg eindeutig. Daher ist es nicht möglich, in zwei verschiedenen Applikationen ein Objekt der selben Objektklasse mit dem gleichen Identifier anzulegen (z.B. Benutzer mit Id A in Applikation X und Benutzer mit Id A in Applikation Y).

Eine weitere Besonderheit ist, dass sich Benutzer und Rollen den Scope teilen, d.h. es kann innerhalb eines @enterprise-Servers keinen Benutzer und keine Rolle geben, deren Ids übereinstimmen, bzw. keinen Benutzer dessen Id mit dem Namen einer Rolle übereinstimmt und umgekehrt.

Für Ids gelten die folgenden Konventionen:

- Ids beginnen mit einem Buchstaben oder \$ oder / oder \, danach können weitere der genannten Zeichen oder eine Ziffer kommen.
- Die Gesamtlänge von Ids darf 80 Zeichen nicht überschreiten.

3.2. INTERNATIONALISIERUNG DER STAMMDATENOBJEKTE UND OBJEKTKLASSEN

 Ids können auch Sonderzeichen enthalten (z.B. Email-Adressen), jedoch sind Leerzeichen, Ausrufezeichen (außer WDL) und Beistriche nicht erlaubt. In einer WDL-Definition muss die Akteurs-Id mit einem Ausrufezeichen beginnen, falls die Akteurs-Id keine gewöhnliche Id ist.
 Beispiel:

!right.user@xy.com do_something(f);

Objektklasse	Identifier
Benutzer	Id
Organisationseinheit	Id
Task–Funktion	Id
Berechtigungsliste	Name
Objektklasse	Name
Funktionsgruppe	Id
Rolle	Id, Name
Recht	Id, Name
Organisationsklasse	Id, Name
OE–Baum	Id, Name
Applikation	Id, Name
Server	Id, Name
Task	Id+Version
Prozess	Id+Version UND Name+Version
Formular	Id+Version UND Name+Version

Tabelle 3.1: Objektklassen und ihre Identifier

3.2 Internationalisierung der Stammdatenobjekte und Objektklassen

In @enterprise können folgende Stammdaten internationalisiert werden:

- Applikationen
- Tasks
- Funktionen
- Rollen
- Rechte

Die Stammdaten von eigenen Applikationen können internationalisiert werden, indem man entsprechende *java.lang.ResourceBundles* implementiert, und diese in das entsprechende Applikationsverzeichnis stellt. Nähere Informationen dazu finden Sie im Programmierhandbuch. Dort finden Sie auch die Information, wie man die Stammdaten der Default–Applikation internationalisieren kann.

4 Definition der Organisationsstruktur

4.1 Rollen

Rollen dienen dazu, sinnvolle Berechtigungsprofile zusammenzufassen. Beispiele für Rollen wären *Sekretär*, *Manager* oder *Brandschutzbeauftragter*. In der Rollenliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Rollen gliedert sich in die folgenden Bereiche:

- Allgemein
- Berechtigungen
- Benutzer
- Historie
- Zugriff
- Referenziert von

4.1.1 Registerkarte: Allgemein

Informationen der Registerkarte *Allgemein* des Stammdatenobjekts *Rolle* (Mussfelder sind fett dargestellt):

- Id: Eindeutige Kurzbezeichnung der Rolle. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1
- Name: Eindeutiger Name der Rolle. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- Typ: In @enterprise werden drei Arten von Rollen unterschieden:
 - lokal: Eine lokale Rolle ist eine Rolle, die genau einer Organisationseinheit zugeordnet wird, d.h. ihr Gültigkeitsbereich beschränkt sich genau auf diese Organisationseinheit.

4.1. ROLLEN

🥹 Rollen: Home (Default) - @enterprise - Mozilla Firefox									
📸 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.role&foreignKey=applic: 🏫 🧕									
Allgemein E	Berechtigungen	Benutzer	Historie	Zugriff	Referenzie	ert von			
ld:	home								
Name:	home							118n: Home	
Applikation:	Default		~						
Тур:	lokal 💌	1							
Beschreibung:									
Referenzrolle:									
AKTIV									
Änderung durc	Änderung durchführen am:		<u></u>						
Löschen				Ok		Abbrech	ien (Übernehme	en

Abbildung 4.1: Objekt-Detailansicht (Allgemein): Rollen

- *global:* Eine globale Rolle ist keiner Organisationseinheit zugeordnet, und daher in jeder Organisationseinheit verfügbar.
- *hierarchisch:* Eine hierarchische Rolle wird, wie eine lokale Rolle, einer bestimmten Organisationseinheit zugeordnet. Ihr Gültigkeitsbereich erstreckt sich über sämtliche untergeordnete Organisationseinheiten.

Beispiel: Die Rolle X ist eine hierarchische Rolle und wird der Organisationseinheit A zugeordnet. Die Organisationseinheit A hat zwei untergeordnete Organisationseinheiten *Sub1* und *Sub2*. Die Rolle X ist auch in den Organisationseinheiten *Sub1* und *Sub2* bekannt.

- Beschreibung: Freier Text.
- Referenzrolle: Durch den Einsatz von Referenzrollen ist es möglich, mehreren Benutzern die selbe Rolle zuzuordnen und dennoch für jeden dieser Benutzer die Berechtigungen, die mit dieser Rolle verbunden sind, individuell festzulegen. Mehrstufige Referenzrollen sind nicht möglich, d.h. für eine Rolle, die als Referenzrolle verwendet wird, kann keine Referenzrolle eingetragen werden. Wie eine Referenzrolle angewendet werden kann, wird anhand des folgenden Beispiels beschrieben:

1. Beispiel: In einem Unternehmen gibt es mehrere Sekretäre. Manche von ihnen haben mehrere Rechte, andere wiederum haben weniger Rechte. In der Prozessdefinition wird als Akteur, für eine Aufgabe die von einem Sekretär erledigt werden soll, die Rolle *sek* eingetragen.

Ein Sekretär V mit vielen Rechten bekommt die Rolle *sek_viele_rechte* und ein Sekretär W mit weniger Rechten bekommt die Rolle *sek_wenige_rechte* zugeordnet. Bei der Rolle *sek_viele_rechte* UND bei der Rolle *sek_wenige_rechte* wird als Referenzrolle die Rolle *sek* eingetragen.

Wird nun der Prozess abgearbeitet, landet der Task im Rollenarbeitskorb beider Sekretäre (da beiden die Rolle *sek* als Referenzrolle zugeordnet wurde).

Welche Aktionen der Sekretär V auf den Task und die damit verbundenen Formulare anwenden kann, wird jedoch durch die Rolle *sek_viele_rechte* festgelegt, und welche Aktionen der Sekretär W auf den Task und die damit verbundenen Formulare anwenden kann, wird durch die Rolle *sek_wenige_rechte* festgelegt.

2. Beispiel: Im Unternehmen gibt es Sekretäre und einen Abteilungsleiter. In der Prozessdefinition P wird als Akteur die Rolle *abt_sek* eingetragen. Diese Rolle wird bei der Rolle *abt* und *sek* als Referenzrolle ausgewählt. Die Benutzer haben weiterhin nur entweder die Rolle *abt* bzw. *sek*, jedoch können Sekretäre und Abteilungsleiter den Prozess P starten und haben somit die gleichen Rechte im 1. Schritt.

- Applikation: Innerhalb dieser Applikation ist die Rolle bekannt.
- Aktiv: siehe Kapitel 2.2.1.
- Änderung durchführen am: siehe Kapitel 2.2.1.

In dieser Ansicht stehen Ihnen die Funktionen *I18n*, *Aktivieren* und *Änderungen zu späteren Zeitpunkt* zur Verfügung.



Hinweis: Wenn Sie eine Rolle **löschen**, werden auch die dieser Rolle zugeordneten Berechtigungen gelöscht.

4.1.2 Registerkarte: Benutzer

In der Registerkarte **Benutzer** sehen Sie eine Auflistung aller Benutzer, die dieser Rolle zugeordnet sind. Sie können sich die Details dieser Zuordnung ansehen.

4.1.3 Registerkarte: Berechtigungen

In der Registerkarte **Berechtigungen** können Sie beliebig viele Rechte dieser Rolle zuordnen. Benutzer, die dieser Rolle zugeordnet sind, haben wiederum die hier festgelegten Berechtigungen. Eine detaillierte Beschreibung zum Berechtigungsdialog finden Sie in Kapitel 5.

4.1.4 Systemrollen

In @enterprise sind vier Systemrollen immer vorhanden:

- Alle (all): Allen Benutzern wird diese Rolle zugeordnet. Damit können allen Benutzern zugängliche Funktionen mit dieser Rolle verbunden werden (z.B. hat ein Prozess, den jeder starten darf, als Akteur des ersten Tasks die Rolle *all*.
- Sys (sys): Die Rolle *sys* wird an Administratoren vergeben. Sie ermöglicht das Durchführen sämtlicher Systemadministrationsfunktionen, wie Stammdaten ändern, Prozesse abbrechen, u.ä.
- home (home): Die lokale Rolle *home* verbindet einen Benutzer mit seiner *home* Organisationseinheit. Ein Benutzer kann maximal in einer Organisationseinheit eine *home*–Rolle haben.
- **OE** (**dept**): Die lokale Rolle *dept* wird als *Eingangskorb* der Organisation verwendet. Wenn man einen Workflow an eine Organisationseinheit schicken will, ohne den Empfänger zu kennen, kann man die Rolle *dept* angeben.

4.2 Rechte

Rechte werden verwendet, um bestimmte Funktionen nur einem eingeschränkten Benutzerkreis zur Verfügung zu stellen. Die Zuordnung zu Benutzern erfolgt entweder direkt oder über Rollen. Eine ausführliche Beschreibung des Berechtigungssystems finden Sie in Kapitel 5. In der Rechteliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Rechten gliedert sich in die folgenden Bereiche:

- Allgemein
- Benutzer
- Historie
- Zugriff
- Referenziert von

4.2.1 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Id: Eindeutige Kurzbezeichnung des Rechtes. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Eindeutiger Name des Rechtes. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche Speichern in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).

4.2. RECHTE

🥹 Rechte: Prozessinstanzen bearbeiten (Default) - @enterprise - Mozilla Firefox 📃 🗖 🔀									
🔝 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.right&foreignKey=applic 🏫 🛞									
Allgemein B	enutzer	Historie	Zugriff	Referenziert von				,	
ld:	proc_inst	t							
Name:	edit_proc	inst				18n: Prozessin	stanzen bearb	eiten	
Applikation:	Default			~					
Löschen	Löschen			Ok		Abbrecher	n Übe	ernehmen	

Abbildung 4.2: Objekt-Detailansicht (Allgemein): Rechte

- Applikation: Innerhalb dieser Applikation ist das Recht bekannt.
- Beschreibung: Freier Text.

In dieser Ansicht steht Ihnen die Funktion 118n zur Verfügung.

4.2.2 Registerkarte: Benutzer

In der Registerkarte *Benutzer* sehen Sie eine Auflistung aller Benutzer, die das gerade ausgewählte Recht besitzen. Sie können sich die Details dieser Berechtigungszuordnung ansehen. Ist dieses Recht auf ein bestimmtes Objekt beschränkt, so wird dieses in der Spalte *Betroffenes Objekt* angezeigt.

4.2.3 Standardrechte

In @enterprise sind standardmäßig folgende Rechte vorhanden:

- Objekte erzeugen (create): Erzeugen eines Objekts
- Objekte bearbeiten (edit): Bearbeiten eines Objekts

- Objekte löschen (delete): Löschen eines Objekts
- Berechtigungen bearbeiten (edit-acl): Bearbeiten von Berechtigungen
- Objekte ansehen (view): Anzeigen eines Objekts
- Objekt ausführen (execute): Ausführen einer Funktion
- Konfiguration (conf): Konfiguration bearbeiten
- Administration (admin): Zugang zur Administration, Arbeiten im restricted mode
- **Prozessinstanzen ansehen (view_procinst):** Ansehen von Prozesshistorie, Liste der Dokumente und Notizen sowie alle Prozessformulare und -versionen.
- **Prozessinstanzen bearbeiten (proc_inst):** Prozessinstanzen abbrechen, Bearbeiter ändern, Ansehen von Prozesshistorie, Liste der Dokumente und Notizen sowie alle Prozessformulare und -versionen. Dieses Recht wird im OE–Kontext des Prozesses aufgelöst, d.h. wenn jemand dieses Recht in einer Organisationseinheit hat, darf er die Prozesse dieser Organisationseinheiten bearbeiten. Es gibt jedoch auch die Möglichkeit, dass nur Prozessinstanzen einer bestimmten Prozessdefinition abgebrochen werden können. Dazu muss zuvor der Objektklasse *Prozesse* das Recht *Prozessinstanzen bearbeiten* zugeordnet werden. Anschließend kann man das Recht *Prozessinstanzen bearbeiten* für eine Prozessdefinition vergeben.
- **OE bearbeiten (dept_edit):** Organisationseinheiten bearbeiten
- Akteur ändern (set_agent): Setzen von Akteuren in Prozessinstanzen, ansehen von Prozesshistorie, Liste der Dokumente und Notizen sowie alle Prozessformulare und -versionen.
- **Statistik (stat):** Erstellung von Statistiken (mit Ausnahme von benutzerbezogenen Statistiken)
- Suchbar (searchable): Um in Formularen suchen zu können bzw. Liste der gespeicherte Anfragen anzuzeigen
- Named User (named_user): Um Benutzer als *Named User* zu qualifizieren und damit das Anmelden unabhängig von der Anzahl der aktuell angemeldeten Benutzer immer zu erlauben.
- Schritt abbrechen (abort_step): Um den Task einer Prozessinstanz abzubrechen.
- Kalendertermine bearbeiten (editCal): Berechtigung zum Bearbeiten von Kalendereinträgen.
- Kalendertermine erzeugen (insertCal): Um Kalendertermine zu erzeugen.
- Kalendertermine ansehen (viewCal): Um auch Kalendertermine zu sehen, die nicht die eigenen sind.
- Freigabe (share): Um anderen Benutzern die Verwendung des Objekts zu gestatten (z.B. Arbeitskorb-Filter)

4.3 Benutzer

Alle Personen, die mit @enterprise arbeiten möchten, müssen als Benutzer eingetragen werden. In der Benutzerliste stehen Ihnen die unter 2.1 beschriebenen Funktionen zur Verfügung. In der erweiterten Suche können Sie durch verschiedene Suchattribute die Anzahl der angezeigten Benutzer in der Benutzerliste beeinflussen. Wenn Sie zum Beispiel als Suchattribut eine Organisationseinheit angeben, werden nur jene Benutzer aufgelistet, die eine Rolle in dieser OE besitzen.

Die Objekt-Detailansicht von Benutzern gliedert sich in die folgenden Bereiche:

- Allgemein
- Rollenzuordnungen
- Vertretungen
- Rollenvertretungen
- Historie
- Zugriff
- Berechtigungen
- Alle Berechtigungen
- Einstellungen

4.3.1 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Id: Die eindeutige Kurzbezeichnung des Benutzers. Die Id des Benutzers muss beim Einloggen in das System im Feld *Benutzer–Id* eingegeben werden. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Nachname: Der Nachname des Benutzers.
- Vorname: Der Vorname des Benutzers.
- Titel: Der akademische oder sonstige Titel.
- Beschreibung: Freier Text.
- E-Mail: Die E-Mail Adresse des Benutzers.
- Telefon-Nr: Die Telefonnummer des Benutzers.
- Server: Der @enterprise-Server, von dem der Arbeitskorb für diesen Benutzer abrufbar ist.
- Sprache: Hier ist die Sprache auswählbar, in der das Benutzerinterface erscheint. Es werden jene Sprachen zur Auswahl angeboten, in die @enterprise übersetzt wurde.

4.3. BENUTZER

🕹 Benutzer: lı	rrasch Markus marku	ıs - @enterpris	e - Mozilla Firefox						-		R
[http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.user&func=edit&comingFrom=%2Fwf%2Fservlet.method%2Fcom.groiss.gui.table.Tat 🏠 🧕)	
Allgemein	Rollenzuordnungen	Vertretungen	Rollenvertretungen	Historie	Zugriff	Berechtigungen	Alle Berechtigungen	Einstellungen	Zusatzinfo		_
ld:	markus										
Nachname:	Irrasch										
Vorname:	Markus										
Titel:											
Beschreibun	g:										
E-Mail:	markus@groiss.co	m									
Telefon-Nr.:	0463/12345										
Server: Sprache: Aktiv Sortierattribu	ep80 (oracle) 💌 Deutsch/Österreicl 🔽	n									
Passwort: Datum der la Passwort-Po	atzten Passwort Ände plicy:	vung: 20-12-2011 ✓ Passw Muss I Kann F	o 0 13:02 rort läuft nie ab Passwort beim nächs Passwort nicht änderr	ten Login á	indern						
Löschen	ırchführen am:						Ok Abb	rechen	Übernehme	en	

Abbildung 4.3: Objekt-Detailansicht (Allgemein): Benutzer

- Aktiv: Kapitel 2.2.1.
- Sortierattribut: Hier kann ein freier Text eingetragen werden. Dieser Text kann bei der Implementierung von eigenen Applikationen für eine attributunabhängige Sortierung (z.B. zuerst alle Ärzte, dann alle dipl. Krankenschwestern, dann alle Krankenpfleger, ... in der Benutzerliste auflisten) verwendet werden.
- Passwort: Das Passwort des Benutzers. Das Passwort muss beim Einloggen in das System im Feld *Passwort* eingegeben werden.
- Datum der letzten Passwortänderung: Datum, wann das Passwort das letzte mal geändert wurde.
- Passwort-Policy:
 - 1. Passwort läuft nie ab: Das Passwort dieses Benutzers muss nie geändert werden.
 - 2. Muss Passwort beim nächsten Login ändern: Der Benutzer wird beim nächsten Login aufgefordert sein Passwort zu ändern.
 - 3. Kann Passwort nicht ändern: Der Benutzer sollte sein Passwort nicht ändern können.
- Änderung durchführen am: siehe Kapitel 2.2.1.

In dieser Ansicht stehen Ihnen die Funktionen Aktivieren und Änderungen zu späteren Zeitpunkt zur Verfügung.



Hinweis: Wird ein Benutzer gelöscht, so werden auch die ihm zugeordneten Rollen, Rechte und Vertreter gelöscht.

4.3.2 Registerkarte: Rollenzuordnungen

In der Registerkarte *Rollenzuordnungen* sehen Sie alle Rollen, die dem Benutzer zugeordnet sind. Sie können in dieser Ansicht die Rollen des Benutzers bearbeiten. Es stehen Ihnen hierzu die Funktionen in der Werkzeugliste zur Verfügung. Wenn Sie eine neue Rollenzuordnung einfügen oder eine bestehende ändern, öffnet sich ein neuer Objektdialog.

Über diesen Dialog legen Sie außerdem die Rollenvertreter des Benutzers fest. Aktivieren Sie hierfür die Registerkarte *Vertretungen* im Objektdialog der Rollenzuordnung. Hier sehen Sie alle Vertretungen für die ausgewählte Rollenzuordnung. Sie können in dieser Ansicht die Vertretungen bearbeiten. Es stehen Ihnen hierzu die Funktionen in der Werkzeugliste zur Verfügung. Wenn Sie eine neue Vertretung einfügen oder eine bestehende ändern, öffnet sich ein weiterer Objektdialog.

Rollenzuordnungen bearbeiten

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- **Benutzer:** Der Benutzer, dem die Rolle zugeordnet werden soll standardmäßig jener Benutzer, in dessen Detailmaske man sich befindet.
- Rolle: Die Rolle, die dem aktuellen Benutzer zugeordnet werden soll.
- Organisationseinheit: Der Benutzer hat die Rolle in dieser Organisationseinheit. Wird dem Benutzer eine hierarchische Rolle zugeordnet, so hat er diese Rolle auch in allen, der hier ausgewählten Organisationseinheit, untergeordneten Organisationseinheiten (siehe 4.1).

Wenn man dem Benutzer eine globale Rolle zuordnet, so kann dieses Feld leer gelassen werden. In diesem Fall hat der Benutzer die Rolle in allen Organisationseinheiten.

- Aktiv: siehe Kapitel 2.2.1.
- Änderungen durchführen am: siehe Kapitel 2.2.1

Stellvertreter für einen Benutzer innerhalb einer bestimmten Rolle definieren

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt);

- Vertreter: Der Benutzer, der die aktuelle Rolle vertreten soll.
- Von: Ab diesem Datum beginnt die Vertretung. Wenn Sie hier kein Datum eintragen, beginnt die Vertretung sofort. Durch das Anklicken des Kalender–Icons neben dem Feld wird ein Kalender angezeigt, der Sie bei der Eingabe des Datums unterstützt.

- Bis: Mit diesem Datum endet die Vertretung. Wenn Sie hier kein Datum eintragen, endet diese Vertretung erst, wenn sie explizit aufgehoben wird. Durch das Anklicken des Kalender–Icons neben dem Feld wird ein Kalender angezeigt, der Sie bei der Eingabe des Datums unterstützt.
- Aktiv: siehe Kapitel 2.2.1

Hinweis: Durch den Timer *CurrentSubstitutes* wird bei Angabe eines Von-/Bis-Datums die Vertretung aktiviert bzw. wieder deaktiviert (siehe Abschnitt 9.1.6).

4.3.3 Registerkarte: Vertretungen

In dieser Ansicht sehen Sie die *Persönlichen Vertreter* des Benutzers, d.h. alle Benutzer, die den ausgewählten Benutzer vertreten und die *Vertretungen* des Benutzers, d.h. alle Benutzer, die vom ausgewählten Benutzer vertreten werden.

Sie können in dieser Ansicht die *Persönlichen Vertreter* des Benutzers bearbeiten. Es stehen Ihnen hierzu die Funktionen in der Werkzeugliste zur Verfügung. Wenn Sie einen neuen Vertreter einfügen oder einen bestehenden ändern, öffnet sich ein neuer Objektdialog.

Persönliche Vertreter für einen Benutzer definieren

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Vertreter: Der Benutzer, der den aktuellen Benutzer vertreten soll.
- Vertretung inkl. Rollen: Ist diese Checkbox angekreuzt, so wird der aktuelle Benutzer auch in all seinen Rollen vertreten.
- Von und Bis: Wie bei Stellvertreter für einen Benutzer in Kapitel 4.3.2. Durch den Timer *CurrentSubstitutes* wird bei Angabe eines Von-/Bis-Datums die Vertretung aktiviert bzw. wieder deaktiviert (siehe Abschnitt 9.1.6).
- Änderungen durchführen am: siehe Kapitel 2.2.1

4.3.4 Registerkarte: Rollenvertretungen

In dieser Ansicht erscheint eine HTML-Seite, in der Ihre Rollenvertretungen angezeigt werden.

Diese HTML-Seite setzt sich aus 2 Bereichen zusammen:

- Im ersten Bereich *Benutzer, die meine Rollen vertreten* sind alle Benutzer aufgelistet, von denen Sie in einer Rolle vertreten werden. Wird eine Aufgabe an eine Rolle weitergeleitet, die Sie innehaben und in der Sie von einem Benutzer vertreten werden (und zwar inkl. Rollen), so scheinen diese Aufgaben im Rollenarbeitskorb Ihres Vertreters auf.
- Im zweiten Bereich *Benutzer, deren Rollen ich vertrete* sind alle Benutzer aufgelistet, die von Ihnen in einer Rolle vertreten werden. Wird eine Aufgabe an eine Rolle weitergeleitet, die einem Benutzer zugeordnet ist, den Sie vertreten (und zwar inkl. Rollen), so scheinen diese Aufgaben in Ihrem Rollenarbeitskorb auf.
Die Tabelle Benutzer, die meine Rollen vertreten enthält die folgenden Informationen:

- Aktiv: Zeigt an, ob die Rolle aktiv (= grüner Punkt) oder inaktiv (= roter Punkt) ist.
- Rolle: Die Rolle, in der Sie von dem Benutzer vertreten werden.
- Organisationseinheit: In dieser Organisationseinheit besitzen Sie die Rolle.
- Benutzer: Der Benutzer, der Sie in einer bestimmten Rolle vertritt.
- Von: In dieser Spalte ist eingetragen, ab welchem Zeitpunkt die Rollenvertretung wirksam ist.
- **Bis:** In dieser Spalte ist eingetragen, bis zu welchem Zeitpunkt die Rollenvertretung wirksam ist.

Die Tabelle Benutzer, deren Rolle ich vertrete enthält die folgenden Informationen:

- Aktiv: Zeigt an, ob die Rolle aktiv (= grüner Punkt) oder inaktiv (= roter Punkt) ist.
- **Rolle:** In dieser Spalte ist die Bezeichnung für die Ihnen (über einen Vertreter) zugeordnete Rolle zu finden.
- Organisationseinheit: In dieser Spalte ist ersichtlich, in welcher Organisationseinheit Sie die entsprechende Rolle vertreten.
- Benutzer: Benutzer, den Sie in einer bestimmten Rolle vertreten.
- Von: In dieser Spalte ist eingetragen, ab welchem Zeitpunkt die Rollenvertretung wirksam ist.
- **Bis:** In dieser Spalte ist eingetragen, bis zu welchem Zeitpunkt die Rollenvertretung wirksam ist.

4.3.5 Registerkarte: Berechtigungen

In dieser Registerkarte können Sie die Berechtigungen des aktuellen Benutzers bearbeiten. Es stehen Ihnen hierzu die Funktionen der Werkzeugleiste zur Verfügung. Informationen zum Einfügen einer neuen bzw. Bearbeiten einer bestehenden Berechtigung sind im Kapitel 5 beschrieben.

4.3.6 Registerkarte: Alle Berechtigungen

Diese Ansicht zeigt Ihnen alle Berechtigungen an, die der Benutzer inne hat. Berechtigungen können Benutzern über Rollen oder direkt zugeordnet werden. Eine allgemeine Einführung in das Berechtigungssystem finden Sie in Kapitel 5. Sie können diese Tabelle mit den Funktionen in der Werkzeugleiste bearbeiten.

Weiters können Sie die Berechtigungen des Benutzers zu einem bestimmten Zeitpunkt ansehen. Geben Sie dazu ein beliebiges Datum (und Uhrzeit) ein und starten die Suche.

4.3.7 Registerkarte: Einstellungen

Hier ist es dem Systemadministrator möglich die lokalen Einstellungen eines Benutzers zu bearbeiten. Um welche Einstellungen es sich dabei handelt entnehmen Sie dem Benutzerhandbuch.

4.4 Organisationseinheiten

Um die Aufbauorganisation Ihres Unternehmens verwalten zu können, bietet @enterprise zwei Konstrukte an: Organisationseinheiten und OE-Bäume. Organisationseinheiten oder kurz *OEs* entsprechen den Abteilungen in Unternehmen. OE–Bäume bilden das Organigramm des Unternehmens ab und werden in Kapitel 4.5 beschrieben.

In der Liste der Organisationseinheiten stehen Ihnen die unter 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Organisationseinheiten gliedert sich in die folgenden Bereiche:

- Allgemein
- Übergeordnete OE's
- Rollen
- Historie
- Zugriff
- Referenziert von

4.4.1 Registerkarte: Allgemein

- Id: Eindeutige Kurzbezeichnung der OE. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Eindeutiger Name der OE. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Beschreibung: Freier Text.
- E-Mail: Die E-Mail-Adresse der OE.
- Tel-Nr: Die Telefonnummer der OE.
- Adresse: Die Anschrift der OE.

4.4. ORGANISATIONSEINHEITEN

Ulgomoin	üharmaardinata OEa – Ballan – Hista	io Zugriff Boforonziart	Non	
Aigemein	ubergeordnete OES - Rollen - Histor	ie zugini Reierenzient	VOIT	
ld:	service			
Name:	Service			
Beschreibung	42			
E-Mail:				
TelNr.:				
Adresse:				
Externe OE:				
Unselbständi	g: 🔲			
@enterprise i	nstalliert: 🗹			
Organisation	klasse:			
AKUV: Naabfalaar O				
Nachiorger-O Sortiorgttribut		•		
SomeratinDu	·			
Änderung dur	chführen am:			

Abbildung 4.4: Objekt-Detailansicht: Organisationseinheiten

- Externe OE: Ist diese Checkbox angekreuzt, so handelt es sich bei der aktuellen OE um eine externe OE, d.h. um eine Organisationseinheit außerhalb Ihres Unternehmens. Dadurch kann während der Abarbeitung eines Prozesses kein Task an diese OE oder eine(n) Rolle (Benutzer) in dieser OE weitergeleitet werden.
- Unselbständig: Dieses Attribut dient der OE–Untergliederung. Es wird verwendet, um OEs zu kennzeichnen, die zu einer anderen OE gehören, aber keine untergeordneten OEs sind (Bsp.: Stabsstellen in einer Organisation, etc.). In der Hierarchie (OE–Baum) werden diese unselbständigen OEs jedoch trotzdem der OE, der sie zugeordnet sind, untergeordnet.

Interpretiert wird dieses Attribut nur innerhalb des Berechtigungssystems (siehe Kapitel 5) von @**enterprise**. Es wird dort dafür verwendet, den OE–Scope von Berechtigungen einzuschränken.

• @enterprise installiert: Ist diese Checkbox angekreuzt, so ist in der aktuellen OE

@enterprise installiert. Sonst ist in der OE **@enterprise** nicht installiert, d.h. es kann während der Abarbeitung eines Prozesses kein Task an diese OE oder eine(n) Rolle(Benutzer) in dieser OE weitergeleitet werden.

- Organisationsklasse: Die Organisationsklasse der OE. Weitere Informationen zu Organisationsklassen finden Sie im Kapitel 4.6.
- Aktiv: siehe Kapitel 2.2.1.
- Nachfolger–OE: Im Rahmen von Umstrukturierungen in Ihrem Unternehmen kann es vorkommen, das Organisationseinheiten durch andere Organisationseinheiten ersetzt werden. Hier kann festgehalten werden, welche OE die aktuelle OE ersetzt hat.
- Sortierattribut: Hier kann ein freier Text eingetragen werden. Dieser Text kann bei der Implementierung von eigenen Applikationen für eine attributsunabhängige Sortierung verwendet werden.
- Änderungen durchführen am: siehe Kapitel 2.2.1

In dieser Ansicht stehen Ihnen die Funktionen Aktivieren und Änderungen zu späterem Zeitpunkt zur Verfügung.

4.4.2 Registerkarte: Übergeordnete OE's

Eine Organisationseinheit kann in mehrere OE–Bäume eingehängt werden. Daher kann eine OE, die normalerweise nur zu einer übergeordneten OE gehören kann, mehrere übergeordnete OEs besitzen (pro OE–Baum genau eine). In dieser Registerkarte sehen Sie, welche übergeordneten OE's die aktuelle Organisationseinheit hat.

4.4.3 Registerkarte: Rollen

In dieser Registerkarte werden alle Rollenzuordnungen für die aktuelle OE aufgelistet.

In der Tabelle *Direkte Rollen* wird angezeigt, welche Rolle, in welcher Applikation, welchem Benutzer in der aktuellen OE direkt zugeordnet ist.

In der Tabelle *Von übergeordneten OEs geerbte Rollen* wird angezeigt, welche hierarchische Rolle, in welcher Applikation, welchem Benutzer durch welche übergeordnete OE in die aktuelle OE vererbt wurde. Genauere Informationen zu hierarchischen Rollen und deren Gültigkeitsbereich finden Sie in Kapitel 4.1.

4.5 OE-Baum

OE–Bäume werden in @enterprise verwendet um das Organigramm eines Unternehmens abzubilden. Durch das Einhängen von Organisationseinheiten in einen OE–Baum wird die Hierarchie der einzelnen OEs abgebildet. Dabei kann eine OE in einem OE–Baum mehrere untergeordnete OEs, aber nur eine übergeordnete OE, besitzen. Eine OE kann auch in mehrere OE–Bäume eingehängt werden.

Es ist möglich mehrere OE–Bäume zu definieren und sie dann Applikationen zuzuordnen. Ein OE-Baum kann in mehreren Applikationen verwendet werden, eine Applikation verwendet aber immer genau einen OE-Baum.

Default-OE-Baum: Nach der Installation enthält das System genau einen OE–Baum mit dem Namen *default*. Dieser OE–Baum ist der Applikation *default* zugeordnet.

Die Objekt-Detailansicht von OE-Bäumen gliedert sich in die folgenden Bereiche:

- Allgemein
- OE-Baum
- Historie
- Zugriff
- Referenziert von

4.5.1 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Id: Eindeutige Kurzbezeichnung des OE-Baums. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Eindeutiger Name des OE-Baums. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.

🕹 OE-Bäume	: OE-Structure - @enterprise - Mozilla Firefox
[http://loca	ihost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.depttree&func=edit&coi 🏫 🛞
Allgemein	OE-Bäume Historie Zugriff Referenziert von
ld: str Name: OE	uct
Lösche	n Ok Abbrechen Übernehmen

Abbildung 4.5: Objekt-Detailansicht: OE-Baum

4.5.2 Registerkarte: OE-Baum

In der Registerkarte *OE-Baum* sehen Sie den Aufbau des OE-Baums. Dabei werden alle OEs, die dem OE-Baum angehören mit dem Namen dargestellt.

4.5. OE-BAUM

Γ

🛿 OE-Bäume: OE-Structure - @enterprise - Mozilla Firefox
📸 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.depttree&func=&cominç 🏫 🎯
Allgemein OE-Bäume Historie Zugriff Referenziert von
۰۰۰ ۵۰۰
Groiss Informatics
 Product Product Bestehende OE hinzufügen Purchas Neue OE Staff an Detail von Baum entfernen OE umhängen
Löschen Übernehmen

Abbildung 4.6: **OE-Baum**

Anmerkung: Bei der folgenden Beschreibung der Funktionen und der Navigation im Baum wird jene Organisationseinheit, durch deren Aktivierung das PopUp–Fenster geöffnet wurde, als *aktuelle OE* bezeichnet.

drilloff.png	Navigation im Baum Im OE-Baum werden drei Symbole verwendet:
	• blauer Kreis: zeigt, dass die aktuelle OE keine untergeordneten OEs besitzt.
drilldown.png	• <i>Plus-Zeichen:</i> diese OE hat eine oder mehrere untergeordnete OEs, die jedoch noch nicht angezeigt werden. Wenn Sie auf dieses Plus-Zeichen klicken, wird der OE-Baum an dieser Stelle expandiert, und die direkt untergeordneten OEs werden angezeigt. Gleichzeitig wird aus dem Plus-Zeichen ein Minus-Zeichen.
drillup.png	• <i>Minus-Zeichen:</i> dieses Zeichen zeigt Ihnen, dass der Teilbaum bereits expandiert dar- gestellt wird, d.h. Sie sehen die untergeordneten OEs einer bestimmten OE. Wenn Sie auf das Minus-Zeichen klicken, werden alle untergeordneten OEs verborgen. Gleich- zeitig wird dadurch auch das Minus-Zeichen vor dieser OE in ein Plus-Zeichen um-

gewandelt.

Funktionen im Baum

× 🛛 – 🔾

ø

770

Funktionen im OE-Baum werden Ihnen einerseits über die Werkzeugleiste und andererseits als Popup-Menü einer ausgewählten OE angeboten. Um das Popup-Menü zu öffnen, klicken Sie mit der linken Maustaste auf jene OE, für die Sie Funktionen ausführen wollen.

- **Teilbaum expandieren:** Durch das Anklicken des Links *Teilbaum expandieren*, wird der gesamte Teilbaum (inkl. seiner Subteilbäume), der sich unter der aktuellen OE befindet, expandiert, d.h. alle untergeordneten OEs werden sichtbar.
- **Bestehende OE hinzufügen:** Durch das Anklicken des Links *Bestehende OE hinzufügen* wird der HTML-Dialog zur Auswahl einer OE angezeigt. Die ausgewählte OE wird anschließend unter der aktuellen OE im OE–Baum eingehängt.
- Neue OE: Durch das Anklicken des Links *Neue OE* kann in einem Schritt zunächst eine neue OE angelegt und anschließend unter die aktuelle OE im OE–Baum eingehängt werden.
- **Detail:** Durch das Anklicken des Links *Detail* wird die Objekt-Detailansicht der aktuellen OE geöffnet.
- Von Baum entfernen: Durch das Anklicken des Links *Von Baum entfernen* wird die aktuelle OE aus dem OE– Baum entfernt.
- **OE umhängen:** Durch das Anklicken des Links *OE umhängen* wird der HTML-Dialog *übergeordnete Org.Einheit von 'aktueller OE'* angezeigt. Wenn Sie hier eine OE auswählen und die Schaltfläche *Übernehmen* aktivieren, wird die aktuelle OE im OE–Baum unter die ausgewählte OE eingehängt. Wenn Sie den Leereintrag aus der Liste auswählen, hat die aktuelle OE keine übergeordnete OE mehr.
- In der Werkzeugleiste finden Sie als erstes die Funktion **Zum Baum auf oberster Ebene hinzufügen**. Mit dieser Funktion können Sie OEs in einem Baum auf oberster Ebene einfügen. Wenn Sie diese Funktion aktivieren, öffnet sich ein HTML-Dialog, in dem Sie die OE, die Sie einfügen wollen, auswählen können.

Sie können über die Suchfunktion in der Werkzeugleiste nach einer beliebigen OE suchen. Diese Funktion verhält sich wie die **Kurzsuche** in der Tabellendarstellung (siehe Kapitel 2.1). Haben Sie mit der Funktion *Suche* nach einer oder mehreren OEs gesucht, werden die gefundenen OEs im OE–Baum **fett** dargestellt. Zusätzlich gibt es *Vorwärts–* und *Zurück–* Links (dargestellt durch >> und <<), die es Ihnen ermöglichen, durch alle gefundenen OEs zu navigieren.

4.5.3 Funktion OE-Bäume zusammenführen

Mit der Toolbarfunktion *OE-Bäume zusammenführen* ist es möglich einen OE-Baum zu einem zweiten OE-Baum hinzuzufügen. Dies kann unter Umständen nötig sein, wenn es

zwei @enterprise Installationen gibt, deren OE-Bäume abgeglichen werden sollen. In der ersten Installation (A) wird der OE-Baum gewartet und an die zweite Installation (B) über XML Export übertragen. In dieser Installation B gibt es allerdings zusätzliche OEs mit Hierarchien, die erhalten bleiben sollen.

In Installation B wird nun ein eigener OE-Baum für die nur in dieser Installation vorhandenen OEs und Beziehungen erstellt. Nach Synchronisation des OE-Baums mit der Installation A (durch XML Import) werden mit dieser Funktion die nur in Installation B vorhandenen Relationen wieder in den OE-Baum hinzugefügt.

Beispiel:

OE-Baum default (auf A): Abt. A Abt. A1 Abt. A2

A1 und A2 sind Sub-OEs von A.

In der Installation B soll der Baum folgendermaßen aussehen: Abt. A

Abt. A1 Abt. A2 Abt. X Abt. Y Abt. Z

Das heißt, X unter A2 und Z unter Y in einem eigenem Teilbaum.

Um die Funktion "OE-Bäume zusammenführen" nutzen zu können, wird die private Struktur auf B in einem eigenen OE-Baum definiert:

Abt. A2 Abt. X Abt. Y Abt. Z

Ein Zusammenführen dieser Struktur in den default OE-Baum ergibt die gewünschte Struktur.

4.6 Organisationsklassen

Organisationsklassen dienen der Klassifizierung von Organisationseinheiten. Diese Information wird in der Logik von @enterprise nicht verwendet, ist allerdings hilfreich für die Strukturierung großer Organisationen. In der Liste der Organisationsklassen stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Organisationsklassen gliedert sich in die folgenden Bereiche:

- Allgemein
- Historie
- Zugriff
- Referenziert von

4.6.1 Registerkarte: Allgemein

🥹 Organisation	sklassen: GI Kärnten - @enterprise - Mozilla Firefox
[http://localho	st:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.orgclass&func=edit&con 🏠 🛞
Allgemein	Historie Zugriff Referenziert von
ld:	kaernten
Name:	GI Kärnten
Beschreibung	OE's mit Hauptsitz in Kärnten
-	
Löschon	
Luschen	

Abbildung 4.7: Objekt-Detailansicht: Organisationsklasse

- Name: Eindeutiger Name der Organisationsklasse. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Id: Eindeutige Kurzbezeichnung der Organisationsklasse. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Beschreibung: Freier Text.

4.7 Schlagworte

Durch das Anklicken dieses Links wird im Arbeitsbereich eine HTML–Seite angezeigt, mit der Sie eine Liste von Schlagworten verwalten können. Die hier erfassten Schlagworte können im Dokumentenmanagement (siehe Benutzerhandbuch) einzelnen Dokumenten zugeordnet werden. Diese Schlagworte können in der Dokumentensuche verwendet werden.

4.8 Server

Dieses Stammdatenobjekt wird aus Gründen der Abwärtskompatibilität zu früheren Versionen von @enterprise, wo es für den Verteilungsmechanismus relevant war, weiterhin angeboten.

Ab Version 6.1, und der damit verbundenen Einführung der Cluster–Architektur, wird die Verwaltung von mehreren Servern unnötig. Es ist jedoch weiterhin erforderlich, dass hier ein einziger Server eingetragen wird, nämlich der, der Ihre Installation ausmacht.

Sollte dieser Server aus Versehen gelöscht werden, muss er wieder eingefügt werden, und zwar mit den Attributwerten, die im Konfigurationsfile von @enterprise eingetragen sind.



Hinweis: Einstellungen, die die Ports betreffen, haben keinerlei Auswirkungen auf das System. Diese Einstellungen müssen in der *Konfiguration* von @enterprise getätigt werden.

5 Berechtigungssystem

5.1 Allgemeines

Das @enterprise Berechtigungssystem erlaubt eine flexible Definition der Zugriffsrechte von Benutzern auf Objekte.

In @enterprise werden zur Umsetzung des Berechtigungssystems die folgenden Begriffe und Objekte verwendet:

- **Rechte**: siehe Kapitel 4.2
- Berechtigungsliste: Berechtigungslisten können Sie in @enterprise verwenden, um mehrere Rechte zusammenzufassen. Sie können diese Berechtigungslisten mehreren Objekten, die das gleiche Berechtigungsprofil haben sollen, zuordnen.
- **Objektklasse:** Objektklassen definieren applikationsabhängige Klassen, die vom Berechtigungssystem verwaltet werden können. Dieses Verhalten können Sie auch bei Formularen definieren.
- **Berechtigung**: Eine Berechtigung legt fest, wer mit welchem Recht auf welches Objekt zugreifen kann. Ein Objekt kann keine, eine oder mehrere Berechtigungen haben. (Kapitel 5.2.1)
- **Standardberechtigungen:** Standardberechtigungen werden bei Objektklassen und Formularen definiert. Wird ein neues Objekt erzeugt, werden die in der zugehörigen Objektklasse bzw. Formular definierten Standardberechtigungen dem neuen Objekt zugeordnet.

5.1.1 Berechtigungsliste

Berechtigungslisten sind Aggregationen von Berechtigungen. Die Objekte, die dem Berechtigungssystem unterliegen, haben ein Attribut (acl), das auf eine Berechtigungsliste verweist. Die für ein Objekt relevanten Berechtigungen sind

- Berechtigungen, die direkt auf das Objekt verweisen
- Berechtigungen, die auf die Berechtigungsliste verweisen, die im Objekt verwendet wird.

5.1.2 Standardberechtigungen

Standardberechtigungen spezifizieren, welche Berechtigungsliste beim Anlegen eines Objektes an das neue Objekt angehängt wird. Standardberechtigungen werden mit einer Objektklasse, einer Berechtigungsliste und optional einer Organisationseinheit definiert. Die Angabe der Organisationseinheit hat den Zweck, Standardberechtigungen OE-abhängig zu definieren. Je nachdem, welcher OE ein Objekt zugeordnet ist, wird die entsprechende Standardberechtigung zugeordnet.

5.1.3 Objektklassen

Objektklassen definieren applikationsabhängige Klassen, die vom Berechtigungssystem für die einzelnen Objekte in @**enterprise** verwaltet werden können.

Die Objekt-Detailansicht von Objektklassen gliedert sich in die folgenden Bereiche:

- Allgemein
- Rechte
- Standardberechtigungen
- Historie
- Zugriff

Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Name: Der Name der Objektklasse. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Klasse: Die Java-Klasse, durch die die Objektklasse implementiert ist.
- Applikation: Gibt an, zu welcher Applikation die Objektklasse gehört.

Registerkarte: Rechte

Zu jeder Objektklasse wird definiert, welche Rechte für die Objekte dieser Objektklasse verwendet werden können. Die Zuordnung von Rechten zu Objektklassen ermöglicht eine Eingrenzung der Rechte, die für die jeweilige Objektklasse beim Festlegen von Berechtigungen verwendet werden können.

Sie können außerdem zu jedem Recht einen Modus definieren. Folgende Modi werden unterschieden:

- Objekte ansehen
- Objekte erzeugen
- Objekte bearbeiten

• Objekt ausführen

Wenn Sie zu einem Recht einen dieser Modi angeben, überschreiben Sie ein Standardrecht. Das heißt, dass dann zur Berechtigungsprüfung für die entsprechende Methode (Einfügen, Bearbeiten, Ansehen) das überschriebene Recht verwendet wird.

Registerkarte: Standardberechtigungen

In dieser Registerkarte erfolgt die Zuordnung von Standardberechtigungen zu einer Objektklasse. Es kann eine OE-unabhängige und je eine OE-abhängige Standardberechtigung vergeben werden.

5.1.4 Formulare und Berechtigung

Formular-Objekte haben wie Objektklassen die Registerkarten *Rechte* und *Standardberechtigungen*. Hier können Sie für einzelne Formulare das gleiche Verhalten, wie für Objektklassen definieren.

5.2 Verwendung von Berechtigungen

In der Stammdatenverwaltung können Berechtigungen von vier Seiten aus bearbeitet werden:

- in der Benutzerverwaltung
- in der Rollenverwaltung
- beim Objekt (Zugriff)
- in der Verwaltung von Berechtigungslisten

Egal, von welcher Seite aus, Sie Berechtigungen definieren, es wird immer der gleiche HTML-Dialog zur Bearbeitung von Berechtigungen verwendet (siehe Abbildung 5.1). Je nachdem von welcher Seite Sie kommen, sind bestimmte Felder befüllt bzw. deaktiviert. Dies soll Sie beim Umgang mit Berechtigungen in @enterprise unterstützen.

5.2.1 Berechtigungsdialog

- **Recht:** Wählen Sie hier aus, welches Recht Sie dem Benutzer zuordnen wollen. Diese Liste enthält alle Standardrechte von @**enterprise** und alle Rechte, die Sie zusätzlich in Ihren Applikationen angelegt haben. Wenn Sie ein Recht aus der Liste auswählen, werden die aktiven Felder *Anwenden auf*, *Gültigkeitsbereich*, *Anwenden in OE* oder *Akteur* zurückgesetzt.
- **Zugriff:** Legen Sie hier fest, ob es sich um eine positive (erlaubte) oder negative (nicht erlaubte) Berechtigung handelt.

5.2. VERWENDUNG VON BERECHTIGUNGEN

🥹 Berechtigungen: Neu - @ente	erprise - Mozilla Firefox	- 0 ×
http://localhost:8380/wf/servlet.r	nethod/com.groiss.storegui.TabbedWindow.showDialog?node=admin.auth&fc	reignł 🏫 🛞
Allgemein Historie		
Recht: Objekte ansehen	~	
Zugriff: Erlaubt		
Akteur		
 Benutzer: Rolle: Gilt nur, wenn Rolle in 	Maier Franz maier	v
Anwenden auf Objektklasse: Formularklasse:	Organisationseinheiten	•
Objekt / Berechtigungsliste:	Service	-
Gültigkeitsbereich:	Objekt 💌	
Anwenden in OE:	×	
Löschen	Ok Abbrechen Über	nehmen

Abbildung 5.1: Objekt-Detailansicht: Berechtigungen

- Akteur: Hier legen Sie fest, für wen diese Berechtigung gilt: Benutzer, Rolle oder Rolle in Organisationseinheit.
 - Wenn Sie von der Benutzerverwaltung aus Berechtigungen bearbeiten, ist der Akteur immer mit dem ausgewählten Benutzer befüllt. Sie können diesen nicht ändern.
 - Wenn Sie aus der Rollenverwaltung Berechtigungen bearbeiten, ist der Akteur immer mit der ausgewählten Rolle befüllt. Auch diese können Sie nicht bearbeiten. Sie können allerdings bei *lokalen* bzw. *hierarchischen* Rollen das Feld *Gilt nur, wenn Rolle in OE* bearbeiten.
 - Wenn Sie vom Objekt aus Berechtigungen (Zugriff) bearbeiten, müssen Sie den Akteur festlegen.
 - Wenn Sie die Rechte einer Berechtigungsliste festlegen, müssen Sie den Akteur ebenfalls festlegen.
- Anwenden auf: Dieser Bereich legt fest, auf welches Objekt, auf welche Objektklasse bzw. welches Formular Sie eine Berechtigung definieren möchten. Ein **Objekt** kön-

nen Sie nur auswählen, wenn Sie zuvor eine **Objektklasse** definiert haben. Welche Objektklasse Sie zur Auswahl haben, hängt davon ab, welches Recht Sie ausgewählt haben. Wenn Sie die Berechtigung auf ein **Formular** anwenden wollen, stehen Ihnen hier wiederum nur in Abhängigkeit zum ausgewählten Recht bestimmte Formulare zur Auswahl.

- Gültigkeitsbereich: Wählen Sie hier den Gültigkeitsbereich der ausgewählten Objektklasse, des Objekts bzw. des Formulars aus. Je nach Auswahl im Bereich Anwenden auf stehen Ihnen unterschiedliche Auswahlmöglichkeiten zur Verfügung.
 - alle Objekte: Wenn Sie keine Auswahl im Feld Anwenden auf getroffen haben (keine Objektklasse, kein Objekt und kein Formular) ausgewählt haben, gilt die Berechtigung für alle Objekte.
 - **Objektklasse:** Wenn Sie ein Formular oder eine Objektklasse, aber kein Objekt ausgewählt haben, gilt die Berechtigungen für alle Objekte dieser Objektklasse.
 - Rollen-OE: Wenn der Akteur eine Rolle ist (und die Objektklasse nicht Organisationseinheiten), gilt die Berechtigung für Objekte der Organisationseinheit, in der der jeweilige Benutzer die bei Akteur angegebene Rolle hat bzw. weitere, abhängig vom OE-Wirkungsbereich.
 - Objekt: Wenn Sie eine Objektklasse und ein Objekt ausgewählt haben, die Objektklasse aber nicht *Berechtigungsliste* ist, gilt die Berechtigungen f
 ür genau dieses Objekt.
 - Objekte der Berechtigungsliste: Wenn Sie die Objektklasse Berechtigungsliste und ein Objekt ausgewählt haben, gilt die Berechtigung für alle Objekte der ausgewählten Berechtigungsliste.
 - Objekte der OE: Wenn Sie die Objektklasse Organisationseinheiten und ein Objekt ausgewählt haben, gilt das Recht für Objekte dieser OE und eventuell weitere, abhängig vom OE-Wirkungsbereich.
- Anwenden in OE (OE-Wirkungsbereich): Der OE-Wirkungsbereich ist notwendig, wenn die Berechtigung OE-bezogen vergeben werden soll. Die Organisationseinheit, in der dem Benutzer die Berechtigung zugewiesen wird, kann mit Hilfe des OE-Wirkungsbereichs abgeändert oder erweitert werden. Sie können hier eine Auswahl vornehmen, wenn der Gültigkeitsbereich *Rollen-OE* ist und Sie kein Objekt ausgewählt haben oder wenn der Gültigkeitsbereich *Objekte der OE* ist und eine beliebige Objektklasse, mit Ausnahme von *Organisationseinheit*, ausgewählt ist. Es stehen Ihnen folgende Möglichkeiten zur Auswahl:
 - lokal: Die Berechtigung gilt für die angegebene Organisationseinheit.
 - hierarchisch: Die Berechtigung gilt für die angegebene und alle Sub-OEs.
 - unselbständig-hierarchisch: Die Berechtigung gilt f
 ür die angegebene und alle unselbständigen Sub-OEs.
 - selbständig: Die Berechtigung gilt f
 ür die n
 ächst
 übergeordnete selbst
 ändige OE.
 - selbständig und selbständig-hierarchisch: Die Berechtigung gilt f
 ür n
 ächst
 übergeordnete selbständige OE und alle unselbständigen Sub-OEs dieser OE.
 - übergeordnete OE: Die Berechtigung gilt für die übergeordnete OE.

OE-Wirkungsbereich

Die unterschiedlichen OE-Wirkungsbereiche werden anhand des folgenden Beispiels verdeutlicht:

Beispiel: Abbildung 5.2 zeigt einen OE-Baum mit selbständigen und unselbständigen Organisationseinheiten. Die dunklen Kreise (OE4, OE6 und OE7) stellen die unselbständigen OEs dar. Eine Berechtigungsvergabe für die Organisationseinheit OE2 in den unterschiedlichen Wirkungsbereichen umfasst die folgenden OEs:

- lokal: OE2
- hierarchisch: OE2, OE3, OE4, OE5, OE6, OE7, OE8, OE9
- unselbständig-hierarchisch: OE2, OE4, OE6, OE7
- selbständige OE: OE2
- selbständige und unselbständige Sub-OEs: OE2, OE4, OE6, OE7
- übergeordnete OE: OE1



Abbildung 5.2: OE-Baum mit selbständigen und unselbständigen OEs

5.2.2 Berechtigungen in der Benutzerverwaltung

Bei jedem Benutzer-Objekt gibt es die Registerkarte *Berechtigungen*, die eine Tabelle aller direkt vergebenen Berechtigungen darstellt. Diese Tabelle kann mit den Funktionen der Werkzeugliste bearbeitet werden. Als Akteur ist der aktuelle Benutzer eingetragen. Sie können diesen nicht ändern.

5.2.3 Berechtigungen in der Rollenverwaltung

Bei jedem Rollen-Objekt gibt es die Registerkarte *Berechtigungen*, die eine Tabelle aller an die Rolle geknüpften Berechtigungen darstellt. Diese Tabelle kann mit den Funktionen der

Werkzeugliste bearbeitet werden. Als *Akteur* ist die aktuelle Rolle eingetragen. Sie können die Rolle nicht verändern. Allerdings können Sie die das Feld **Gilt nur, wenn Rolle in OE** bearbeiten, wenn es sich bei der aktuellen Rolle um eine *lokale* oder *hierarchische* Rolle handelt.

5.2.4 Berechtigungen beim Objekt – Zugriff

Bei allen Objekten, die der Kontrolle des Berechtigungssystems unterliegen, gibt es die Registerkarte Zugriff. Hier tragen Sie die Berechtigungen des Objekts ein, d.h. wer mit welchem Recht auf das Objekt zugreifen kann. Sie können den Zugriff auf ein Objekt über direkte Berechtigungen (oberer Teil der Registerkarte) oder über eine Berechtigungsliste (unterer Teil der Registerkarte) definieren.

5.2.5 Verwaltung von Berechtigungslisten

Um Rechte gruppieren und wiederverwenden zu können, können Sie Berechtigungslisten anlegen. Im Bereich *Organisation* finden Sie den Link *Berechtigungslisten*. Dieser öffnet die Tabellendarstellung, die alle bereits definierten Berechtigungslisten anzeigt. Über die Funktionen der Werkzeugleiste können Sie die Einträge bearbeiten. Eine Berechtigungsliste wird nur mit ihrem Namen (Kapitel 3.1) beschrieben.

In der Registerkarte *Rechte* können Sie beliebig viele Rechte zur jeweiligen Berechtigungsliste hinzufügen bzw. diese bearbeiten. Dieser Dialog entspricht dem Berechtigungsdialog in der Benutzerverwaltung (Kapitel 5.2.1). Allerdings können Sie hier die Felder im Bereich *Anwenden auf* nicht bearbeiten. Auf welches Objekt die jeweilige Berechtigungsliste angewendet wird, hängt davon ab, an welches Objekt Sie die Berechtigungsliste binden. Um Ihnen diesen Umstand zu signalisieren, wird der Gültigkeitsbereich **Objekte dieser Berechtigungsliste** verwendet.

Um überhaupt Berechtigungen einer Berechtigungsliste zuordnen zu können, brauchen Sie allerdings das Recht *Berechtigungen bearbeiten (edit-acl)* auf die Berechtigungsliste.

5.2.6 Rechte um Berechtigungen zu bearbeiten

Benutzer dürfen Berechtigungen bearbeiten, wenn Sie das Recht Berechtigungen bearbeiten (edit-acl) für alle Objekte besitzen, sofern der Eintrag auf kein bestimmtes Objekt verweist. Verweist der Eintrag auf ein bestimmtes Objekt, muss der Akteur das Recht Berechtigungen bearbeiten (edit-acl) für das betreffende Objekt oder die Objektklasse besitzen. Außerdem ist es notwendig, das Recht Objekt ausführen (execute) für das Recht, welches in der Berechtigung auftritt, zu besitzen. Damit ist es möglich, die Vergabe von Berechtigungen auf bestimmte Rechte einzuschränken.

5.2.7 Sonderfall: Berechtigungen für Rollenzuordnungen

Die Vergabe von Rollenzuordnungen stellt einen Sonderfall dar, da durch eine Zuordnung von Rollen ein Benutzer sich selbst Rechte geben kann. Für die Manipulation von Rollenzuordnungen ist es daher notwendig zwei Rechte zu besitzen:

- die Berechtigung den Benutzer zu bearbeiten Objekt bearbeiten (edit)
- die Berechtigung Rollenzuordnungen zu vergeben *Objekt ausführen (execute)* auf die Rolle

5.3 Standardeinstellungen

Die in @enterprise zur Verfügung stehenden Rechte sind in Kapitel 4.2 beschrieben.

Der Rolle **sys** ist das Recht *Objekte bearbeiten (edit)*, *Objekt ausführen (execute)*, *Berechtigungen bearbeiten (edit-acl)*, *Objekte erzeugen (create)* und *Administration (admin)* für alle Objekte zugeordnet.

Der Benutzer **sysadm** hat die Rolle *sys*, ihm ist außerdem das Recht *Konfiguration (conf)* direkt zugeordnet.

Alle Stammdatenmanipulationen können daher von einem Benutzer mit der Rolle sys durchgeführt werden. Änderungen in der Konfiguration, sowie das Ausführen der Funktionen Server abschalten und Logfile ansehen darf nur der Benutzer sysadm oder ein Benutzer, dem das Recht Konfiguration (conf) zugeordnet ist.

5.4 Wofür werden welche Rechte benötigt?

Die Tabellen 5.1 und 5.2 geben Ihnen einen Überblick darüber, welche Rechte Sie für welche Aktionen benötigen:

5.5 Beispiel

Anhand des folgenden Beispiels soll die Verwendung des Berechtigungssystems in @enterprise gezeigt werden.

Aufgabenstellung: Der Benutzer *Franz Maier* soll die Berechtigung erhalten, Benutzer der Abteilung *Service* zu administrieren, d.h. die Eigenschaften eines Benutzers inkl. dessen Rollenzuordnungen zu bearbeiten. Die Benutzer werden zentral angelegt.

Lösung: Um die Benutzer der OE *Service* administrieren zu können, wird ihm das Recht *Objekte bearbeiten (edit)* für die Objektklasse *Organisationseinheit* und dem Objekt *Service* gegeben. Um in die Administration von **@enterprise** zu gelangen, wird ihm das Recht *Administration (admin)* zugeordnet.

Um die Rollenzuordnungen administrieren zu können, wird eine Rolle definiert, die alle Rollenzuordnungen (mit Ausnahme der Rolle *sys*) für Benutzer bearbeiten darf. Eine lokale Rolle mit dem Namen *Rollenzuordnungen bearbeiten* wird angelegt. Dieser Rolle werden die Berechtigungen *Objekt ausführen (execute)* auf die Objektklasse *Rolle* erlaubt und die Berechtigung *Objekt ausführen (execute)* auf die Rolle *sys* NICHT erlaubt (siehe Abb. 5.3).

Benutzer A will	Benötigtes Recht (Name, Id)	Anwenden		
ein Objekt anlegen	Objekte erzeugen (create)	auf alle Objekte oder die Objektklasse		
ein Objekt ändern oder löschen	Objekte bearbeiten (edit)	auf das Objekt, die Objektklasse, alle Ob- jekte oder die OE (wenn das Objekt zu ei- ner OE gehört). <i>Gilt für alle Stammdaten-</i> <i>objekte außer OEs</i> .		
eine OE ändern oder lö- schen	OE bearbeiten (dept_edit)	auf alle Objekte, die Objektklasse Organi- sationseinheiten oder eine bestimmte OE		
konfigurieren	Konfiguration (conf)	auf alle Objekte		
in die Administration	Administration (admin)	auf alle Objekte		
die Log-Datei ansehen	Administration (admin)	auf alle Objekte		
eine Berechtigung ein- tragen, ändern oder lö- schen	Berechtigungen bearbei- ten (edit_acl) UND Objekt ausführen (execute)	auf das Objekt, die Objektklasse oder alle Objekte (<i>Berechtigungen bearbeiten</i>); auf das Recht (<i>Objekt ausführen</i>)		
eine Rollenzuordnung eintragen, ändern oder löschen	Objekt bearbeiten (edit) UND Objekt ausführen (execute)	auf Benutzer (Objekt bearbeiten); auf die Rolle (<i>Objekt ausführen</i>)		
eine Funktion ausführen	Objekt ausführen (execute)	auf alle Objekte, die Objektklasse <i>Applika-</i> <i>tion</i> oder eine bestimmte Applikation		
einen Prozess abbrechen	Prozessinstanzen bearbeiten (proc_inst)	auf alle Objekte oder alle OEs oder die OE, in der der Prozess gestartet wurde (<i>Objekt</i> <i>ansehen</i>)		
den Prozessakteur in der Prozesshistorie ändern	Prozessinstanzen bearbeiten (proc_inst) ODER Akteur ändern (set_agent)	auf alle Objekte oder alle OEs oder die OE, in der der Prozess gestartet wurde (<i>Objekt</i> <i>ansehen</i>)		
die Prozesshistorie, Liste der Dokumente und No- tizen, alle Prozessformu- lare und -versionen anse- hen	Prozessinstanzen ansehen (view_procinst)	auf einen bestimmten Prozess oder alle		
Prozesse archivieren	Konfiguration (conf)	auf alle Objekte		
Prozessinstanzen suchen	Objekte ansehen (view) UND Prozessinstanzen bearbeiten (proc_inst)	auf alle Objekte oder alle OEs oder die OE, in der der Prozess gestartet wurde (<i>Objekt</i> <i>ansehen</i>)		
Statistiken erstellen	Statistik (stat)	auf alle Objekte		
abgespeicherte Statisti- ken ausführen	Objekt ausführen (execute)	auf alle Objekte, die Objektklasse Gespei- cherte Anfragen oder eine bestimmte An- frage		
Stammdatenobjekt(e) lö- schen	Objekte löschen (delete)	auf alle Objekte, die Objektklasse		
standalone Formular lö- schen	Objekte löschen (delete)	auf die Formularklasse		
Formularänderungen in Prozessinstanz pi sehen	Prozessinstanzen bearbeiten (proc_inst) ODER Akteur ändern (set_agent)	auf die OE/Prozessdefinition, in der pi läuft		
Formularänderungen in Prozessinstanz pi sehen	Objekte ansehen (view) ODER A ist Akteur der Stepinstance	auf die OE, in der pi läuft		

Tabelle 5.1: Wofür braucht man welches Recht?

5.6. BERECHTIGUNG UND VERTRETUNG

Benutzer A will	Benötigtes Recht (Name, Id)	Anwenden
ein Objekt erzeugen	Objekte erzeugen (create) UND Objekt bearbeiten (edit)	create auf die Objektklasse des Objekts und edit auf den Ordner, in dem das Objekt er- stellt wird
ein Objekt ändern, Me- tadaten ändern oder ein Dokument ersetzen	Objekte bearbeiten (edit)	auf das Objekt
ein Objekt löschen, einen Folder mit Inhalt löschen	Objekte bearbeiten (edit)	auf das Objekt, auf den Ordner und wenn das Objekt ein Ordner ist auf alle Inhalte
ein Objekt ansehen	Objekte anzeigen (view)	auf das Objekt
ein Objekt verschieben	Objekte bearbeiten (edit)	auf den Quell-Folder und auf den Ziel- Folder
ein Objekt kopieren	Objekte bearbeiten (edit) UND Objekte anzeigen (view)	auf den Ziel-Folder (<i>Objekte bearbeiten</i>); auf das Objekt bzw. wenn das Objekt ein Ordner auf alle Inhalte (<i>Objekte anzeigen</i>)
ein Objekt umbenennen	Objekte bearbeiten (edit)	auf das Objekt
Berechtigungen auf das Objekt ändern (Zugriff)	Berechtigungen bearbeiten (edit_acl)	auf das Objekt
eine Version erzeugen	Objekte bearbeiten (edit)	auf das Objekt
eine Version ansehen	Objekte anzeigen (view)	auf das Objekt
eine Version löschen	Objekte bearbeiten (edit)	auf das Objekt
die Eigenschaften anse- hen	kein Recht notwendig	
ein Formular löschen	Objekte löschen (delete), Ob- jekte bearbeiten (edit)	<i>Objekte löschen</i> auf die Formularklasse, <i>Objekte bearbeiten</i> auf den Ordner, in dem sich das Formular befindet
ein Subformular löschen	Objekte löschen (delete), Objekte bearbeiten (edit)	<i>Objekte löschen</i> auf die Formularklasse, <i>Objekte bearbeiten</i> auf das Hauptformular

Tabelle 5.2: Wofür braucht man welches Recht im DMS?

Recht	Betroffenes Objekt	Gültigkeitsbereich	Anwenden in OE	Positiv
Persönliche Berechtigungen				
Objekte bearbeiten	Organisationseinheiten : Service	Objekte der OE	lokal	•
Administration		alle Objekte		•
Berechtigungen über Rolle Roll	enzuordnungen bearbeiten in OE Service in OE Se	rvice		
Objekt ausführen	Rollen	Objektklasse		•
Objekt ausführen	Rollen : Sys	Objekt		•

Abbildung 5.3: Beispiel: Berechtigungen

5.6 Berechtigung und Vertretung

Das Verhalten von Berechtigungen bei Vertretungen ist noch eine Betrachtung wert. Die Implementierung folgt den folgenden Regeln:

- 1. Bei Übernahme einer Vertretung verliert der Benutzer keine eigenen Rechte.
- 2. Bei Übernahme einer Vertretung von einer anderen Person hat der Benutzer nachher nie mehr Rechte als beide Personen in Summe haben.

5.6. BERECHTIGUNG UND VERTRETUNG

Die Auswertungsstrategie der Berechtigungen geht folgendermaßen vor sich: **Frage:** Auf welche Objekte hat ein Benutzer ein bestimmtes Recht?

- Schritt 1: Ermittle die Menge der Objekte ohne Berücksichtigung der Vertretungen.
- Schritt 2: Für alle Vertretenen: Berechne die Menge der Objekte mittels aller positiven Rechte der durch die Vertretung übernommenen Rollen. Subtrahiere davon alle Objekte, für die negative Rechte beim Vertretenen existieren, egal ob dieses Recht zu einer Rolle gehört, die vertreten wird oder nicht. Addiere die resultierende Menge von Objekten zur Ergebnismenge.

Bei Übernahme einer persönlichen Vertretung übernimmt der Benutzer alle persönlichen Rechte des Vertretenen, bei Übernahme einer Rollenvertretung alle mit der Rollenzuordnung verbundenen Rechte. Bei Konflikten, z.B. ein Recht wird einem Vertretenen explizit entzogen, dem Vertreter aber gewährt, gilt Regel 1.

6 Workflowelemente

Anmerkung: Im folgenden werden die Begriffe *Workflow* und *Prozess* synonym verwendet. Ein Prozess im Workflow–Management–System entspricht einem Geschäftsprozess in Ihrem Unternehmen (z.B.: wie läuft die Aufnahme einer Bestellung ab, wie werden Kreditanträge in einer Bank abgewickelt, wie wird ein Urlaubsantrag bearbeitet).

Im folgenden werden die Objektklassen vorgestellt, die für die Definition von Prozessen notwendig sind. Im wesentlichen beinhaltet die Definition neuer Prozesse Antworten auf folgende Frage:

WER macht WAS, WANN und WOMIT?

- WER: Wer ist für die Bearbeitung eines Workflows zuständig? Es muss für jede Aktivität eines Workflows definiert sein, wer für dessen Abarbeitung verantwortlich ist. Die Zuordnung von Akteuren zu Aktivitäten basiert in der Regel auf dem Rollenkonzept, d.h. einer Aktivität wird eine Rolle als Akteur zugeordnet, und jeder Benutzer, der diese Rolle besitzt, kommt als potenzieller Bearbeiter der Aktivität in Frage. Der Benutzer, der die entsprechende Aktivität als erster annimmt, wird dann zum tatsächlichen Akteur (Bearbeiter) für diese Aktivität. Ein Benutzer kann auch direkt als Bearbeiter festgelegt werden.
- WAS: Was (welche Arbeit) wird mit Hilfe des Workflows erledigt? Eine Aufgabe wird in Einzelaufgaben (Aktivitäten) zerlegt, die von verschiedenen Bearbeitern ausgeführt werden können.
- WANN: Nachdem festgelegt wurde, welche Aktivitäten es gibt und wer dafür zuständig ist, muss noch geklärt werden, in welcher Reihenfolge die einzelnen Aktivitäten vernünftigerweise bearbeitet werden sollen. In der Regel werden Aktivitäten sequentiell, also nacheinander, abgearbeitet. Es kann in bestimmten Situationen aber durchaus auch z.B. eine parallele Abarbeitung sinnvoll und vor allem zeitsparend sein.
- WOMIT: Für die Bearbeitung der Aktivitäten sind bestimmte Informationen erforderlich. Es muss also festgelegt werden, welche Aktivität welche Information benötigt und wie Aktivitäten ihrerseits Informationen für andere Aktivitäten zugänglich machen. Die Verwendung von Formularen mit strukturierten und nicht strukturierten Feldern ermöglicht eine sehr einfache Form des Informationsaustausches zwischen Aktivitäten.

Eine Prozessdefinitionen besteht aus folgenden Elementen:

- Applikationen: Logisch zusammengehörende Prozesse werden zu Applikationen zusammengefasst. Beispiele: Einkauf, Qualitätssicherung, administrative Aufgaben, Callcenter.
- Tasks: Die elementaren Bestandteile in Prozessen werden als Task (oder auch als Aktivität oder Schritt) bezeichnet. Beispiele: Bestellung aufnehmen, informieren, genehmigen, kopieren, Anruf entgegennehmen, testen. (Beantwortung der Frage *was?*)
- Funktionen: Programmstücke, die automatisierte Teilaufgaben eines Tasks darstellen, werden als Funktionen bezeichnet. Beispiele: Notiz anhängen, kopieren. (Beantwortung der Frage *was?*)
- **Prozesse:** Ein Prozess wird verwendet, um den Ablauf eines Geschäftsprozesses als Ganzes zu beschreiben. Beispiele: Kreditantrag, Urlaubsantrag, Bestellung. (Beantwortung der Frage *wann?*)
- Formulare: Um die Daten (Informationen) der Prozesse zu verwalten, und diese über die einzelnen Tasks hinweg auszutauschen, werden Formulare verwendet. (Beantwortung der Frage *womit?*)
- Interfaces: Um Prozesse über ein Formular starten zu können, werden Interfaces verwendet.
- Akteure: Die in einem Prozess beteiligten Akteure werden über Rollen (und Organisationseinheiten) oder durch Benutzer direkt beschrieben. (Beantwortung der Frage *wer?*)

Die einzelnen Elemente einer Prozessdefinition werden im folgenden Abschnitt beschrieben. Rollen, Organisationseinheiten und Benutzer wurden bereits weiter oben in diesem Handbuch detailliert beschrieben (siehe Kapitel 4.1, Kapitel 4.3 und Kapitel 4.4).

6.1 Applikationen

Applikationen erlauben es, zusammengehörende Prozesse und deren Elemente (Formulare, Rollen etc.) zu gruppieren. In der Applikationenliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Applikationen gliedert sich in die folgenden Bereiche:

- Allgemein
- Historie
- Zugriff
- Eigenschaften

6.1. APPLIKATIONEN

🥹 Applikationsliste: ITSM -	©enterprise - Mozilla Firefox
http://localhost:8380/wf/ser	vlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.application&func=edit&c 🏠 餐
Allgemein Historie Zu	griff Eigenschaften
ld:	itsm
Name:	ITSM I18n: ITSM
OE-Baum:	Default
Desenteibung.	
Applikationsklasse: Client-Applikationsklasse: Applikationsverzeichnis:	com.groiss.itsm.ITSMApplication
Version:	8.0
Startup Position:	
Löschen	Ok Abbrechen Übernehmen

Abbildung 6.1: Objekt-Detailansicht: Applikationen

6.1.1 Registerkarte: Allgemein

- Id: Eindeutige Kurzbezeichnung der Applikation. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Eindeutiger Name der Applikation. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- **OE–Baum:** Nach dieser OE–Hierarchie werden hierarchische Rollen und Rechte ausgewertet. Pro Applikation kann genau ein OE-Baum ausgewählt werden. Genauere Informationen zu OE–Bäumen finden Sie im Kapitel 4.5.
- Beschreibung: Freier Text.
- Applikationsklasse: Hier kann eine Java–Klasse angegeben werden, die das Interface com.groiss.wf.ApplicationAdapter implementiert. Dieses Interface kann das Ver-

halten der Applikation beeinflussen. Für eine genaue Beschreibung siehe die API– Dokumentation und das Programmierhandbuch.

- Client–Applikationsklasse: Analog zur Applikationsklasse wird diese Klasse f
 ür den Java–Client verwendet.
- Applikationsverzeichnis: Dieses Verzeichnis gibt an, wo die Applikation installiert ist. Klicken Sie auf das Icon *Konfiguration ansehen*, um den Inhalt der entsprechende Konfigurationsdatei (der Applikation) anzuzeigen.
- Version: Hier können Sie eine Version für die Applikation definieren. Dies kann z.B. bei einem Upgrade hilfreich sein, um Applikationen aus älteren @enterprise-Versionen zu unterscheiden.
- Startup Position: Die Applikationen werden aufgrund der hier eingegebenen Position in aufsteigender Reihenfolge geladen. Dies kann z.B. nötig sein, wenn eine Applikation A2 Abhängigkeiten zu einer Applikation A1 besitzt und daher sichergestellt werden muss, dass Applikation A1 vor A2 geladen wird.
- Button *Upgrade*: Dieser Button erscheint nur, wenn eine aktuellere Version der Applikation am Dateisystem gefunden wurde als die installierte. Durch Aktivieren dieses Buttons werden alle definierten Aktionen im Hintergrund ausgeführt. Für nähere Informationen darüber, sehen Sie bitte im API von @enterprise nach (*ApplicationA-dapter.getVersion()* und *ApplicationAdapter.upgrade()*).

6.1.2 Registerkarte: Eigenschaften

In dieser Registerkarte können Sie weitere Eigenschaften für die Applikation festlegen, die unter anderem in einer XML-Datei (*properties.xml*) gespeichert werden. Folgend wird kurz auf die einzelnen Elemente dieser Registerkarte eingegangen:

- Ressource Strings: Hier können Sie den Pfad zu einem Ressource-Bundle (*.xls bzw.
 *.properties-Files) angeben, dass von dieser Applikation zur Internationalisierung verwendet wird, z.B. *com.groiss.itsm.resource.Strings*. Falls noch kein Ressource-Bundle existiert, wird eines angelegt, wenn die Ressourcen Editor Toolbarfunktion *Neu* verwendet und anschließend gespeichert wird (siehe Kapitel 6.9). Weitere Informationen über Ressource-Bundles können Sie im *Programmierhandbuch* unter dem Kapitel *Internationalization of Applications* finden.
- Applikations-Parameter: Hier können Sie Parameter festlegen, die für diese Applikation verwendet werden. Diese Parameter werden in einer XML-Datei gespeichert und sind daher auch in der *Konfiguration* von @enterprise verfügbar (siehe Abschnitt 10).
- Benutzer-Parameter: In diesem Bereich können Sie Parameter für Benutzer festlegen, die diese Applikation verwenden. Diese Parameter werden ebenfalls in einer XML-Datei gespeichert und sind über die *Einstellungen* eines jeden Benutzers aufrufbar (siehe *Benutzerhandbuch* - Abschnitt *Die Gruppe Extras*).

Als *Applikations-* und *Benutzer-Parameter* können hierbei über diesen Dialog nicht alle HTML-Elemente (z.B. Textfelder, Radio-Buttons, etc.) abgebildet werden. Um dies zu realisieren, muss in der jeweiligen Applikation die Datei *properties.xml* angelegt bzw. adaptiert werden (siehe *Programmierhandbuch*).

Hinweis: Die Funktionen dieser Registerkarte stehen nur dann zur Verfügung, wenn für die Applikation ein *Applikationsverzeichnis* festgelegt wurde.

6.2 Tasks

Tasks sind die einzelnen Schritte eines Prozesses. Tasks können in unterschiedlichen Prozessen verwendet werden. In der Taskliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Tasks gliedert sich in die folgenden Bereiche:

- Allgemein
- Eskalation
- Funktionen
- Historie
- Zugriff
- Referenziert von

6.2.1 Registerkarte: Allgemein

- Id: Die eindeutige Kurzbezeichnung des Tasks. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Der Name des Tasks. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- Version: Die eindeutige Versionsnummer des Tasks. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Applikation: Gibt an, zu welcher Applikation der Task gehört.
- Beschreibung: Freier Text. Dieser Text wird im Arbeitskorb auf der HTML–Seite *Taskbeschreibung* (durch Anklicken des Links in der Spalte *Task* eines Arbeitskorbeintrags) angezeigt. Sie können hier z.B. eine Kurzhilfe oder eine Anleitung für den Benutzer anführen. @enterprise bietet hierbei die Möglichkeit, den Text in diesem Feld entsprechend zu formatieren wie es z.B. aus Microsoft Word bekannt ist.
- Preprocessing: Hier kann eine Java–Methode angegeben werden, die ausgeführt wird, bevor der Task in den Arbeitskorb eines Benutzers gelangt. Weiters gibt es die Möglichkeit ein GROOVY-Skript einzutragen, indem Sie groovy: in das Feld schreiben und anschließend das Skript anfügen. Details dazu finden Sie im Programmierhandbuch.

🎱 Task: inform ((1) - @enter	prise - Mozi	lla Firefox				_ 🗆 🔀
http://localhos	t:8380/wf/serv	let.method/com	i.groiss.storegui.Tabb	edWindow.showD	ialog?node=admin.task&foreig	nKey=application≈	opl 🚖 🛞
Allgemein E	skalation	Funktionen	Historie Zugriff	Referenzier	t von		
ld:	inform						
Name:	inform					I18n: inform	- 1
Version:	1						
Applikation:	Default		~				
Beschreibung:	B <i>I</i> <u>U</u>	1 🗄 🖉					
Methoden							
Preprocessing	g:						8
Postcondition	:						\odot
Postcondition	-Meldung:						
Compensatior	1:						✓
Take:							\odot
Untake:							\odot
Aktiv: 💌	Al	ktuelle Versio überschreiber	n .: 💌				
Maximale Dau	er:	0	l Tage	~	Kosten: 0		
Erstakteur zur	Laufzeit set	zbar:	alle OEs 🛛 🔽	_			
weitere Akteur	e zur Laufze	it setzbar: 🛛	alle OEs 🛛 🔽				
Löschen				Ok	Abbrechen	Überneł	nmen

Abbildung 6.2: Objekt-Detailansicht: Tasks

- Postcondition: Hier können Sie eine (XPath-)Bedingung, Java–Methode oder GROOVY-Skript eingeben, die/das vom System überprüft wird, bevor ein Benutzer einen Task weiterleitet. Sollte die Bedingung nicht erfüllt sein, ist ein Weiterleiten nicht möglich. Die Syntax dieser Bedingungen wird in Kapitel 7.1.5 erläutert. Um eine XPath-Bedingung hinzuzufügen, müssen Sie analog zu Groovy den Präfix *xpath:* und anschließend die Bedingung angeben.
- Postcondition–Meldung: In dieses Feld können Sie einen Text eingeben, der im Falle einer negativen Auswertung der Postcondition ausgegeben wird.
- Compensation: Hier wird eine Java–Methode oder GROOVY-Skript angegeben, die/das beim Zurückgehen zu einem früheren Schritt ausgeführt wird. Diese Java–Methode

kann verwendet werden, um die Auswirkungen der Bearbeitung des aktuellen Tasks rückgängig zu machen. So könnten z.B. die Auswirkungen der in den Feldern *Methodenaufruf, Take Hook* und *Untake Hook* definierten Java–Methoden rückgängig gemacht werden, sofern die entsprechenden Methoden bei der Bearbeitung des Tasks aufgerufen wurden.

- Take: Hier wird eine Java–Methode oder GROOVY-Skript angegeben, die/das beim Annehmen (vom Rollenarbeitskorb in den persönlichen Arbeitskorb) aufgerufen wird.
- Untake: Hier wird eine Java–Methode oder GROOVY-Skript angegeben, die/das beim Zurücklegen (vom Arbeitskorb in den Rollenarbeitskorb) aufgerufen wird.
- Aktiv: siehe Kapitel 2.2.1
- Aktuelle Version überschreiben: Beim Ändern eines Tasks wird normalerweise eine neue Version generiert. Durch das Ankreuzen dieser Checkbox kann dieses Verhalten unterdrückt werden, d.h. beim Ändern eines Tasks wird keine neue Version erzeugt sondern die aktuelle überschrieben.
- Maximale Dauer: Mit diesem Attribut wird die maximale Dauer des Tasks festgelegt (in Tagen, Stunden oder Minuten).
- Kosten: Diese Information wird von @enterprise nicht verwendet. Sie steht nur für etwaige Auswertungen zur Verfügung.
- Erstakteur zur Laufzeit setzbar: Durch dieses Attribut wird festgelegt, ob der Erstakteur für diesen Task erst zur Laufzeit eingetragen werden kann. Ist dies der Fall, so muss in der Prozessdefinition bei diesem Task kein Akteur angegeben werden. Weiters muss bei der Prozessdefinition darauf geachtet werden, dass dieser Task nicht als erster Task (Starttask) im Prozess verwendet wird, da der Prozess sonst nicht gestartet werden kann. Der Default–Wert für dieses Attribut ist *alle OEs*. Grundsätzlich stehen drei Werte zur Verfügung:
 - nicht: Der Erstakteur ist nicht zur Laufzeit setzbar, d.h. der Akteur muss schon bei der Prozessdefinition angegeben werden. Wurde der Erstakteur nicht bei der Prozessdefinition angegeben, kann der Prozess zur Laufzeit nicht weitergeleitet werden, wenn dieser Task der Folgetask des aktuellen Tasks des Prozesses ist.
 - *innerhalb OE:* Der Erstakteur ist zur Laufzeit setzbar, wenn beim Weiterleiten kein OE–Wechsel erfolgt.
 - *alle OEs:* Der Erstakteur ist in jedem Fall zur Laufzeit setzbar, unabhängig davon, ob beim Weiterleiten ein OE–Wechsel erfolgt oder nicht.
- Weitere Akteure zur Laufzeit setzbar: Durch dieses Attribut wird festgelegt, ob zur Laufzeit für diesen Task weitere Akteure angegeben werden können, d.h. ob für diesen Task die Funktion *Bearbeiter ändern* (siehe Benutzerhandbuch) zur Verfügung steht oder nicht. Der Default–Wert für dieses Attribut ist *alle OEs*. Grundsätzlich stehen drei Werte zur Verfügung:
 - nicht: F
 ür diesen Task steht die Funktion Bearbeiter
 ändern nicht zur Verf
 ügung.

- *innerhalb OE:* Für diesen Task kann ein Akteur gesetzt werden, wenn beim Weiterleiten kein OE–Wechsel erfolgt.
- *alle OEs:* Für diesen Task kann ein Akteur gesetzt werden, unabhängig davon, ob beim Weiterleiten ein OE– Wechsel erfolgt oder nicht.

٧

Hinweis: Wird ein Task gelöscht, werden auch die ihm zugeordneten Eskalationsobjekte, und die vorhandenen Zuordnungen von Task–Funktionen gelöscht.

6.2.2 Registerkarte: Eskalation

Eskalationsobjekte können Sie verwenden, um auf Zeitüberschreitungen bzw. das vorzeitige Erreichen eines Zeitpunktes bei der Ausführung von Tasks reagieren zu können. Es ist möglich verschiedene Aktionen zu definieren, die bei einer Zeitüberschreitung ausgeführt werden sollen. Der @**enterprise**–Timer *Escalations* ist für die Prüfung der Zeitüberschreitungen verantwortlich. Läuft dieser Timer nicht, erfolgt auch keine Prüfung der Zeitüberschreitungen! Weiters ist zu beachten, dass jedes Eskalationsobjekt genau 1x ausgeführt wird!

Die Registerkarte *Eskalation* zeigt zunächst alle zu diesem Task definierten Eskalationsobjekte. Sie können diese mit den Funktionen der Werkzeugleiste bearbeiten.

- Eskalationstyp: Hier können Sie zwischen 3 Eskalationstypen wählen:
 - 1. Aktivität Sollzeit: Dieser Eskalationstyp ist als Default eingestellt und wird beim Eintreten einer Eskalation im *Arbeitskorb*, *Rollen-Arbeitskorb*, *Wiedervorlage* oder *Rollen-Wiedervorlage* ausgelöst. Für jeden Task kann im Reiter *Allgemein* die *Maximale Dauer* festgelegt werden, auf die dieser Eskalationstyp reagiert.
 - 2. Aktivität nicht angenommen: Diese Art der Eskalation wird dann ausgelöst, wenn der Task eine gewisse Zeit (*Zeitabstand*) im Rollen–Arbeitskorb verweilt. Dabei ist zu beachten, dass dieser Typ nur im Rollen–Arbeitskorb funktioniert. Es kann hierbei auch ein negativer Zeitabstand angegeben werden, jedoch würde dies bei diesem Eskalationstyp keinen Sinn machen!
 - 3. Aktivität ungesehen: Dieser Eskalationstyp reagiert auf den aktuellen Task, wenn dieser im Arbeitskorb bzw. Rollenarbeitskorb als *Ungelesen* gesetzt ist. Dies ist z.B. der Fall, wenn der Task im Arbeitskorb landet und der Benutzer ihn noch nicht angesehen hat.
- Schritt: Ist in Taskeskalationen nicht auswählbar, nur in Prozesseskalationen.
- Zeitabstand: Dieser Zeitabstand gibt an, wann eine Zeitüberschreibung vom *Escalations*–Timer festgestellt werden soll. Dieser Wert kann auch negativ sein, um vorzeitig auf eine gewisse Deadline reagieren zu können. Ist dieser Zeitabstand überschritten wird die definierte Aktion ausgeführt. Dabei können Sie zwischen *Stunden*, *Tage* und *Arbeitstage* auswählen. Als Nicht-Arbeitstage werden standardmäßig immer Samstag

Allgemein				
Eskalationstyp:	Prozess S	Sollzeit	~	
Schritt:				-
Zeitabstand:	1	Tage	~	
Aktion				
💿 E-Mail senden				
🗹 Aktuelle A	kteure			
Empfänger:				
🔘 Andere Aktion				
Java Methode:				\odot
🔘 Task starten				
Task:				-
Rolle:				-
🔘 Prozess starte	n			
				-

Abbildung 6.3: Objekt-Detailansicht: Eskalation

und Sonntag angenommen. Sie können weitere Nicht-Arbeitstage unter *Konfiguration* \rightarrow *Kalender* festlegen (siehe Kapitel 10).

Beispiel:

Wenn 2 Arbeitstage (48h) eingegeben wurde, heute Donnerstag 16 Uhr ist und der Prozess am darauffolgenden Montag um 16 Uhr das Fertigstellungsdatum hat, wird die Eskalation ausgelöst (Annahme: nur Samstag und Sonntag sind Nicht-Arbeitstage).

- Aktion: Es werden vier Arten von Aktionen unterschieden:
 - *E-Mail senden:* Tritt bei der Bearbeitung eines Tasks eine Zeitüberschreitung auf, wird eine entsprechende E-Mail an die im Feld *Empfänger* eingetragene E-Mail Adresse geschickt. Ist die Option *Aktuelle Akteure* ausgewählt, wird an die Akteure des aktuellen Tasks eine E-Mail versandt, sofern beim Benutzer eine E-Mailadresse eingetragen wurde. Ist der aktuelle Akteur des Tasks eine Rolle, so werden an alle Benutzer, die die Rolle (in dieser Organisationseinheit) besitzen, per E-Mail benachrichtigt.

Damit das Verschicken der E-Mail funktioniert, muss in der Serverkonfiguration unter *Kommunikation* im Feld *SMTP Host* ein gültiger Mailserver eingetragen sein (siehe Installationshandbuch). Auf dieser HTML-Seite kann im Feld *Mail Sender* eine gültige E-Mail Adresse angegeben werden, die dann als *Sender* der durch die Zeitüberschreitung verschickten E-Mails aufscheint. Wird in diesem Feld kein Wert eingetragen, wird als Sender der Default-Wert enterprise@Hostname des @enterprise-Servers eingetragen. Beispiel

enterprise@lima.groiss.com

 Andere Aktion: Hier kann eine Java–Methode angegeben werden, die im Falle einer Zeitüberschreitung ausgeführt werden soll.
 Beispiel

com.groiss.DemoClass.demoMethod

- Task starten: Tritt bei der Bearbeitung eines Tasks eine Zeitüberschreitung auf, wird der im Feld Task ausgewählte Task gestartet. Wurde im Feld Rolle eine Rolle ausgewählt, so wird der Task für diese Rolle gestartet, sonst ist der Akteur des gestarteten Tasks gleich dem Akteur des Tasks, bei dem eine Zeitüberschreitung festgestellt wurde. Der gestartete Task gehört zum selben Prozess, wie der Task, bei dem die Zeitüberschreitung ausgelöst wurde.
- Prozess starten: Tritt bei der Bearbeitung eines Tasks eine Zeitüberschreitung auf, wird der im Feld Prozess ausgewählte Prozess als Subprozess des aktuellen Prozesses gestartet.
- Beschreibung: Freier Text.

6.2.3 Registerkarte: Funktionen

In dieser Registerkarte können Sie einzelne Task-Funktionen diesem Task zuordnen. Allgemeine Informationen zu Funktionen finden Sie im Kapitel 6.3.

In der Liste **Verfügbare Funktionen** sehen Sie alle Task-Funktionen, die Sie zum Task hinzufügen können. In der Liste **Ausgewählte Funktionen** sehen Sie alle Funktionen, die dem Task bereits zugeordnet sind.

Um eine Funktion dem Task zuzuordnen, wählen Sie zunächst eine Funktion aus der Liste *Verfügbare Funktionen* aus und aktivieren anschließend die Schaltfläche >. Die Funktion scheint nun in der Liste *Ausgewählte Funktionen* auf. Gespeichert wird diese Änderung aber erst nach Aktivieren der Schaltfläche **Ok**, **Übernehmen** oder beim Wechseln auf eine andere Registerkarte.

Um vom aktuellen Task wieder eine Funktion zu entfernen, wählen Sie diese Funktion aus der Liste *Ausgewählte Funktionen* aus und aktivieren anschließend die Schaltfläche <. Die Funktion scheint nun in der Liste *Verfügbare Funktionen* auf.

6.3. FUNKTIONEN

🎱 Task: adHoc (1) - @enterprise - Mozi	illa Firefox		
[http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.task&foreignKey=application&appl 🏫 🥹			
Allgemein Eskalation Funktionen	Historie Zugriff Referenziert von		
Verfügbare Funktionen:	Ausgewählte Funktionen:		
Als Prozessformulartemplate n	Priorität setzen		
Löschen	Ok Abbrechen Übernehmen		

Abbildung 6.4: Registerkarte: Funktionen

6.2.4 Ergänzen von Formularen

Formulare können normalerweise nur von den aktuellen Akteuren im Rahmen der für den jeweiligen Prozessschritt definierten Feldsichtbarkeiten bearbeitet werden. Dies würde selbst bei einfachsten Korrekturen (Umsetzen eines Wertes in einem Read-Only Feld) ein Zurückgehen zum jeweiligen Akteur und Schritt erfordern.

Mit Hilfe des vordefinierten *Ergänzen Tasks* (engl. *Supplement-Task*) kann diese im Einzelfall umständliche Handhabung vereinfacht werden. Der Ergänzen Task wird der jeweiligen Prozessdefinition zugeordnet (Prozesseditor: *Prozess* \rightarrow *Tasks*). Nun können diesem Task Formulare zugeordnet werden und Feldsichtbarkeiten definiert werden.

Benutzer mit dem Recht *Prozessinstanzen bearbeiten* oder *Akteur ändern* erhalten dann für solche Prozesse aus der Ansicht der Prozesshistorie die Möglichkeit, Änderungen an den Formularinhalten vorzunehmen.

Dazu erscheint in der Historie ein Icon *Ergänzen*, über das man eine Aktivität Ergänzen für die aktuelle Prozessinstanz in den eigenen Arbeitskorb legen kann. Dort ändert man die Formulare und leitet die Aktivität weiter. Die Änderungen sind über die Historie umfassend nachvollziehbar.

Wenn man selbst nicht aktueller Akteur der Prozessinstanz ist (aber eines der o.g. Rechte besitzt), dann kann man die Erstellung des Ergänzen Tasks und dessen Weiterleitung automatisch durch einfaches Ändern und Speichern der Formulare direkt aus der Historie auslösen.

6.3 Funktionen

Task–Funktionen (oder auch kurz Funktionen genannt) sind einer Applikation zugeordnete Java–Methoden. Abhängig vom Typ der Funktion kann/muss/darf diese allen Tasks, einem Task oder keinem Task zugeordnet werden. Durch das Anklicken des Task–Funktions–Icons (siehe Benutzerhandbuch) neben einem Task werden in einem Kontextmenü alle Funktionen aufgelistet, die dem Task entweder direkt zugeordnet wurden oder die vom Typ *zu allen*

Tasks sind. In der Funktionsliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Task-Funktionen gliedert sich in die folgenden Bereiche:

- Allgemein
- Historie
- Zugriff
- Referenziert von

6.3.1 Registerkarte: Allgemein

🥹 Funktionen: (Un)gelesen setzen - @enterprise - Mozilla Firefox			
nttp://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.taskfunc&foreignKey=aj 🏠 🥯			
Allgemein Historie	2 Zugriff Referenziert von		
Id:	toggle_seen		
Name:	toggle_seen I18n: (Un)gelesen setzen		
Applikation:	Default		
Anwenden auf:	einen Eintrag 💌		
	🗹 zu allen Tasks		
Anzeigen:	🗹 Arbeitskorb 🗹 Zu nicht angenommenen Tasks		
	Historie		
	in Funktionsliste		
Beschreibung:	$\mathbf{B} \ I \ \underline{\mathbf{U}} \ \vdots \ \vdots \ \not $		
	com.dec.avw.html.HTMLGui.toggleSeen)	
Methode:			
Zielfenster:			
Sortierattribut:			
Funktionsgruppe:	▼		
Icon Name:	Vorschau:		
·			
Löschen	Ok Abbrechen Übernehme	en 🖉	

Abbildung 6.5: Objekt-Detailansicht: Task-Funktionen

- Id: Die eindeutige Kurzbezeichnung der Funktion. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Der Name der Funktion. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- Applikation: Gibt an, zu welcher Applikation die Funktion gehört.
- Anwenden auf: Hier kann der Typ ausgewählt werden, ob die Task-Funktion eine globale oder lokale Funktion ist:
 - keinen Eintrag: Die Funktion kann als globale Funktion ausgeführt werden, d.h. es muss kein Eintrag z.B. im Arbeitskorb selektiert sein.
 - einen Eintrag: Die Funktion wird als lokale Funktion ausgeführt, d.h. es muss nur 1 Eintrag ausgewählt sein.
 - mehrere Einträge: Analog zu *einen Eintrag*, jedoch können mehrere Einträge ausgewählt werden.
- Anzeigen:
 - allen Tasks zuordnen: Die Funktion ist automatisch bei allen Tasks dieser Applikation sichtbar.
 - Arbeitskorb: Ist diese Checkbox aktiviert, scheint die Funktion (ihrem Typ entsprechend) im (Rollen-)Arbeitskorb über das Funktionsrädchen auf.
 - Zu nicht angenommenen Tasks: Ist diese Checkbox angekreuzt, scheint die Funktion auch bei jenen Tasks im Prozessformular auf, die sich im Rollenarbeitskorb befinden.
 - Historie: Ist diese Checkbox aktiviert, scheint die Funktion (ihrem Typ entsprechend) in der Prozesshistorie auf.
 - in Funktionsliste: Die Funktion kann keinem Task zugeordnet werden (z.B. Administrationsfunktion oder Suchfunktion). Diese Funktion scheint dann im Arbeitsbereich auf, wenn im Navigationsbaum der Link *Funktionen* angeklickt wird.
- Beschreibung: Freier Text. @enterprise bietet hierbei die Möglichkeit, den Text in diesem Feld entsprechend zu formatieren wie es z.B. aus Microsoft Word bekannt ist.
- Methode: Die Bezeichnung der Java–Methode, durch die die Funktion realisiert ist. Sie können bei der Methode auch Parameter angeben. Fügen Sie an den Methodennamen ein ? und die Parameterliste an.
 Beispiel: com.groiss.DemoClass.demoMethod?param1=val1¶m2=val2

Weiters gibt es auch die Möglichkeit wie bei *Tasks* ein GROOVY-Skript einzutragen, indem Sie *groovy:* in das Feld schreiben und anschließend das Skript anfügen.

- Zielfenster: Es kann sein, dass die Funktion eine Ausgabe erzeugt. Der Inhalt dieses Feldes bestimmt wo diese Ausgabe angezeigt wird. Wird in dieses Feld nichts eingetragen, wird die Ausgabe im Arbeitskorb angezeigt. Das Feld kann folgende Einträge enthalten:
 - _top: Die Ausgabe wird im gesamten aktuellen Browser-Fenster angezeigt.
 - Leerer Eintrag Die Ausgabe wird im Hauptfenster angezeigt.
 - Name eines Fensters: Die Ausgabe wird in diesem Fenster angezeigt. Es können Attribute wie width, height, scrollbars etc. mit Beistrich getrennt an den Namen gehängt werden. Syntax wie bei JavaScript-Methode window.open;
- Sortierattribut: Geben Sie hier jenes Attribut ein, nach dem sortiert werden soll.
- Funktionsgruppe: Eine selbst definierten Funktionsgruppe kann hier ausgewählt werden.
- Icon Name: Sie können im Java–Client zu einer Funktion ein beliebiges Icon definieren. Diese Angabe kann ein absoluter oder ein relativer Pfad zum Klassenpfad sein.

Damit ein Benutzer eine Task-Funktion ausführen darf, muss er das Recht *Objekt ausführen* (*execute*) für die jeweilige Funktion haben.

6.3.2 Standardfunktionen

In @enterprise werden vordefinierte Funktionen, die zur Erweiterung der Arbeitskorbfunktionalität dienen, zur Verfügung gestellt:

- (Un)gelesen setzen (toggle_seen): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*, *Zu nicht angenommenen Tasks*. Diese Funktion setzt den aktuellen Eintrag auf gelesen bzw. ungelesen.
- In die Zwischenablage (into_clipboard) Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Diese Funktion legt den aktuellen Eintrag in der Zwischenablage ab.
- Aus Zwischenablage einfügen (from_clipboard): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Mit dieser Funktion können Sie den Eintrag aus der Zwischenablage einfügen.
- Kopie an... (make_copy): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Diese Funktion erstellt eine Kopie des aktuellen Eintrags.
- **Prozessnotiz** (attach_note): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Mit dieser Funktion können Sie eine Notiz zum aktuellen Eintrag (Prozess) hinzufügen.
- **Prozessnotiz** (note_global): Anwenden auf: *keinen Eintrag*, Anzeigen: *Arbeitskorb*, *in Funktionsliste*. Mit dieser Funktion können Sie eine Notiz nach Auswahl eines Prozesses an diesen hängen.

- Fertigstellungsdatum setzen (set_duedate): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Sie können mit dieser Funktion das Fertigstellungsdatum des Tasks und/oder des gesamten Prozesses setzen.
- **Prozessrelationen definieren** (addRelation): Anwenden auf: *einen Eintrag*, Anzeigen: *Arbeitskorb*. Nähere Informationen zu dieser Task–Funktion finden Sie im Programmierhandbuch.
- **Priorität setzen** (set_priority): Anwenden auf: *einen Eintrag*, Anzeigen: *allen Tasks zuordnen*, *Arbeitskorb*. Mit Hilfe dieser Funktion kann die Priorität der Prozessinstanz verändert werden.

Nähere Informationen zu den Funktionen finden Sie im Kapitel Anhang des Benutzerhandbuches.

6.4 Formulare

Formulare stellen ein wesentliches Element in **@enterprise** dar. Alle prozessspezifischen Informationen werden in einem solchen Formular verwaltet. Es können für jeden Prozess (und dessen Aktivitäten) ein bis mehrere Formulare existieren, die die lokalen Daten des konkreten Geschäftsfalles enthalten.

In der Formularliste stehen Ihnen neben den in Kapitel 2.1 beschriebenen Funktionen, die folgenden Funktionen zur Verfügung:

- Neues Formular erstellen
- Formular bearbeiten
- HTML ersetzen
- Formular anzeigen
- View erzeugen

Formulare, deren Formularklasse nicht geladen werden kann, werden wie inaktive Tabelleneinträge (siehe Kapitel 2.1) angezeigt. Der Ablauf zur Erstellung bzw. Bearbeitung eines Formulars ist in Abbildung 6.6 zu sehen und wird nachfolgend erklärt.

Die Objekt-Detailansicht von Formularen gliedert sich in die folgenden Bereiche:

- Allgemein
- Javaklasse
- Datenbanktabelle
- Rechte
- Standardberechtigungen
- Historie
- Zugriff
- Vorschau
- Ordnereinstellungen
- Referenziert von



Abbildung 6.6: Ablauf zur Erstellung bzw. Bearbeitung eines Formulars

6.4.1 Neues Formular erstellen

Nach dem Sie diese Funktion aktiviert haben, öffnet sich der Assistent zum Erstellen und Bearbeiten von Formularen.

Schritt 1

1

Im ersten Schritt des Assistenten definieren Sie grundlegende Eigenschaften des Formulars. Welche Bedeutung die einzelnen Felder besitzen ist in Kapitel 6.4.3 detailliert beschrieben. Unter anderem können Sie hier ein *Basisformular* angeben. Wenn sich Ihr neues Formular auf eine bestehende Tabelle (eines anderen Formulars) bezieht, d.h. Sie wollen ein Formular erzeugen, das ein View auf eine bestimmte Tabelle (und damit auf ein anderes Formular) ist, wählen Sie in diesem Feld den Namen des Basisformulars aus. Durch Aktivieren der Schaltfläche **Weiter** wechseln Sie zum nächsten Schritt.

6.4. FORMULARE

🥹 @enterprise: Formul	arassistent - Mozilla Firefox	_ 🗆 🛛
http://localhost:8380/w	f/servlet.method/com.groiss.avw.html.HTMLFormBuilder.showMask1?node=admin.formtype&foreignKey=applications and the served statemethod and the served state	n&applica 🏠 🛞
Neues Formular	Formulardefinition	1 2 3 4 5
Id: Name: Version: Applikation: Versionierung: Typ: Im DMS verwendbar: Beschreibung:	candidature Bewerbungsformular 5 Default Prozessformular Dokumentenformular Ordnerformular	
Icon: Event-Handler:		
DMS Tabellen-Handler:		
Basisformular:		
	Weiter Zurück /	Abbrechen

Abbildung 6.7: Formularassistent: Schritt 1

Schritt 2

🥹 @enterprise: Formulara	assistent - Mozilla Firefox 📃 🗖 🔀
http://localhost:8380/wf/s	ervlet.method/com.groiss.avw.html.HTMLFormBuilder.loadHTML?&node=admin.formtype&foreignKey=application&applica 🏠 🎯
Formular laden	Formulardefinition 1 2 3 4 5
Formular-Id: candidature	Name: Candidature Form Version: 5
Maskentyp:	
Pfad am Server:	candidature_5.xhtml Bearbeiten
File hochladen:	
	Weiter Zurück Abbrechen

Abbildung 6.8: Formularassistent: Schritt 2

In diesem Schritt können Sie eine HTML-Datei auswählen bzw. XHTML-Datei angeben (Pfad) bzw. hochladen, die die Basis für das neue Formular darstellt. Falls Sie jedoch keine HTML- bzw. XHTML-Datei angeben möchten, wählen Sie den *Maskentyp* XFORMS aus und klicken Sie auf *Weiter* um zum nächsten Schritt zu gelangen, wo sich der Form-Wizard öffnet (siehe Abschnitt 6.4.1). Sie können jedoch auch für den Maskentyp XFORMS ei-

ne Datei hochladen bzw. einen Pfad angeben. Sobald ein Pfad angegeben wurde, kann die Datei direkt mit dem Button *Editieren* am Server bearbeitet werden, sofern in der Konfiguration (siehe Installationshandbuch) aktiviert, d.h. es kann direkt der HTML-Code verändert werden.

Als Quelldateien für Formulare dienen HTML–Formulare oder XHTML-Formulare. XHTML ist eine Reformulierung von HTML in XML, Dateien müssen also syntaktisch korrektes XML sein.

Geben Sie ein HTML-Formular an, wird dieses geparst und das Ergebnis in der Datenbank abgelegt. Die Quelldatei wird vom System dann nicht mehr benötigt.

Geben Sie eine XHTML-Datei an, muss sich diese im Klassenpfad des Servers befinden, also z.B. im *classes* Verzeichnis. Die Datei wird ebenfalls geparst, um die weiteren Ladeschritte durchzuführen. Die Quelldatei wird allerdings auch zur Laufzeit benötigt, da sie nicht in der Datenbank abgelegt wird. Vorteil ist, dass Änderungen im HTML die ohne Auswirkungen auf die Felddefinition bleiben, in der Quelldatei jederzeit durchgeführt werden können und sich sofort auf die Formularanzeige auswirken.

Schritt 3

Wenn Sie in Schritt 2 eine Datei angegeben haben, wird das Formular auf den Server geladen und geparst. Das Ergebnis wird in diesem Schritt dargestellt. Ein Formularfeld wird festgelegt durch die Angabe von *Name*, *Beschriftung*, *Typ* und *Länge*. Falls Sie jedoch keine Datei angegeben, wird in einem zusätzlichen Reiter *Layout* der Form-Wizard angeboten (siehe Abschnitt 6.4.1).

Informationen des Reiters Formularfelder in Schritt 3:

- Name: Der Name für das Feld in der Datenbank (wird aus der geladenen HTML-Datei übernommen).
- Beschriftung: Wird das Formular als Subformular in einem anderen Formular verwendet, wird es im Hauptformular als Tabelle dargestellt. Der Wert in diesem Feld wird dann als Spaltenüberschrift für dieses Feld in dieser Tabelle verwendet. Falls Sie dieses Feld im übergeordneten Formular in der Tabelle nicht anzeigen wollen, lassen sie die Spaltenüberschrift leer. Mit der Funktion **Entfernen** neben der Spaltenüberschrift, können Sie alle Überschriften zurücksetzen.
- Typ: Datenbanktyp für dieses Feld. Der Defaultwert für dieses Feld wird aus dem geladenen HTML–Formular ausgelesen. Er kann dort mit dem Attribut *dbtype* im *input*–Tag des jeweiligen Formularfeldes angegeben werden. In welchen Java–Typ ein *dbtype*–Eintrag bei der Erzeugung, der für das Formular benötigten Java–Klasse umgewandelt wird, entnehmen Sie bitte der Tabelle 6.1.

Bei Angabe der Typen sind jeweils die Datenbankspezifika zu beachten: z.B. erlaubt Oracle 8 bei varchar 4000 Zeichen, der SQL-Server 7 nur 256 Zeichen. Der Typ char dient zur Speicherung von Zeichenketten fixer Länge. Wird in ein char(3) Feld der String *sw* gespeichert, so steht in der Datenbank der Wert 'sw ' – ein Vergleich in Java mit 'sw' schlägt also fehl.



6.4. FORMULARE

ormular bear	beiten					1234	
Formular-Id: candi	idature	Name: Bewerbungs	sformular Version: 5				
Formularfelder	Layout	Tabellenansicht					
lame		Beschriftung	Тур		Länge	+	
candidate		candidate	string	-	30	×	
assessor		assessor	com.groiss.org.User	•			
assessor2		assessor2	com.groiss.org.User	-			
ou		ou	com.groiss.org.OrgUnit	-			
cdate		cdate	date	-			
ddate		ddate	dateTime	-			
desiredsalary		desiredsalary	double	-			
givensalary		givensalary	double	-			
notes		notes	string	-	4000		
employed		employed	boolean	-			
position		position	decimal	-	1		
education		education	string	-	5		

Abbildung 6.9: Formularassistent: Schritt 3

• Länge: Länge des Feldes in der Datenbank.

Wenn Sie alle Ihre Formularfelder richtig definiert haben, aktivieren Sie die Schaltfläche **Weiter**, um zum nächsten Schritt zu gelangen.

Hinweis: Bei XForms ist es möglich den Datenbanktyp *binary* für Felder zu definieren. Diese Felder sollten aber nur via API befüllt werden.

dbtype	DB	Java–Typ
VARCHAR	VARCHAR	String
CHAR	CHAR	String
LONG	LONG (Oracle < 8.1)	String
	CLOB (Oracle > 8.1)	
	Text (SQL–Server)	
DECIMAL	DECIMAL	long ^a
		double ^b
DATE	DATE (Oracle)	date ^c
	DATETIME (SQL–Server)	
DATETIME	DATETIME	date ^d
NONE ^e		
Klassenname ^f	$DECIMAL(20) - oid^g$	Klasse
	VARCHAR(200) — Klassen-	
	name ^h	

^aWenn für das Attribut Länge eine ganze Zahl (Bsp.: 3) eingegeben wurde.

^bWenn für das Attribut Länge eine Kommazahl (Bsp.: 3,4) eingegeben wurde.

^cNur das Datum wird angezeigt.

^dDas Datum und die Uhrzeit wird angezeigt.

^eDas Feld taucht in der Java-Klasse und in der Datenbank nicht auf. Wird zum Abspeichern von Zwischenergebnissen in Formularen benötigt.

^fDiese Klasse muss das Interface *com.groiss.store.Persistent* implementieren, oder die Subklasse einer Klasse sein, die dieses Interface implementiert.

^gIm Formular wird die *oid* gehalten. Angezeigt wird das Ergebnis der *toString()*–Methode des entsprechenden Objekts.

^hoptional, falls eine Klasse angegeben wird, die ein Interface, eine Abstrakte Klasse oder das Interface *com.groiss.store.HasSubclasses* implementiert.

Tabelle 6.1: dbtype eines HTML–Formularfeldes und dessen Entsprechung in der Datenbank bzw. zugehörigen Java–Klasse

Form-Wizard: Reiter Layout

Der Form-Wizard stellt eine bequeme Art der Formularerstellung und Bearbeitung dar (siehe Abbildung 6.10). Der Form-Wizard steht dabei nur für den *Maskentyp* XFORMS zur Verfügung.



Hinweis: Formulare, die in @**enterprise** 7.0 mit Hilfe des Form-Wizards erstellt wurden, sind im XHTML-Format abgespeichert. Um diese Formulare in @**enterprise** 8.0 weiterhin bearbeiten zu können, werden Sie beim Ausführen der Funktion *Formular bearbeiten* gefragt, ob Sie das Formular von XHTML nach XFORMS konvertieren wollen. Wenn Sie die Konvertierung durchführen, können Sie das Formular mit dem neuen Form-Wizard bearbeiten, andernfalls ist ein Bearbeiten mit dem Form-Wizard nicht mehr möglich.

In der Menüleiste unterhalb des Headers (Formular–Id, Name, Version) werden die Standardfunktionen zum Bearbeiten der Formularfelder bereitgestellt. Weiters können Sie einzelne Felder in ihrer Reihenfolge verändern, indem Sie ein Feld auswählen und mit den Pfeil-Buttons nach oben oder unten verschieben. Zudem gibt es noch den Menüpunkt Eigenschaften, wo Sie unter anderem dem Formular Überschriften zuordnen können. Die Toolbarfunktion *Neues Layout erstellen* generiert aus den zuvor erzeugten bzw. bestehenden Formularfeldern ein Standard-Layout, d.h. die Zuordnung Formularfeld zu Eingabefeld wird automatisch angelegt.

Für die Erstellung eines Formulars bietet der Form-Wizard folgende Elemente:

- Text: Einfacher Text ohne Eingabemöglichkeit.
- Linie: Horizontale Linie.
- Formularfelder: Beinhaltet alle Formularfelder, die nachfolgend näher beschrieben werden.
- **Tabelle:** Mit Hilfe dieser Funktion können Sie unter zwingender Angabe eines *Klassennamens* und einer *Id* Subformulare einbinden. Falls Sie bereits ein Formular erstellt haben, das als Subformular in Frage kommen würde, können Sie dieses mit dem Symbol rechts neben *Klassenname* aussuchen und ins Formular einbinden. Optional können Sie die Tabelle über das Eingabefeld *Beschriftung* beschriften.

Die *Formularfelder* bestehen aus mehreren Elementen, wobei folgende gleiche Eigenschaften in allen Formularfeldern vorkommen:

- **Beschriftung:** Frei wählbarer Bezeichner. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9). Der I18n-Link wird nur angezeigt, wenn die Checkbox *Lokalisieren* aktiviert wurde.
- Lokalisieren: Durch Aktivieren dieser Checkbox wird der Wert im Beschriftungsfeld internationalisiert.

6.4. FORMULARE

🥹 Formular bearbeiten - Mozilla Firefox			
http://localhost:8380/wf/servlet.method/com.groiss.avw.html.HTMLFormBuild	er.doLoadHtml		☆ 😵
Formular bearbeiten Formular	definition		1 2 <mark>3</mark> 4 5
Formular-Id: candidature Name: Bewerbungsformular Version	:5		
Formularfelder Layout Tabellenansicht			
Bewerbungsformular			
Bewerber			
Begutachter			
2. Begutachter			
Abteilung 🗸			
Bewerbungsdatur 💼			
Bewerbungsgesp			
Notizen			=
Gehaltsvorstellun			
Gehalt			
Einstellen?			
Position O Softwareentwickler O Administrator O Reinig	ungskraft		
	Weiter	Zurijck	Abbrechen
			, abbredien
			.::

Abbildung 6.10: Form-Wizard: Schritt 3

- CSS-Klasse: Angabe einer CSS-Klasse für Stylesheets.
- **Datenbankfeld:** Eindeutiger Bezeichner, der das Feld identifiziert wurde bereits unter 6.4.1 erstellt.

Jedes Formularfeld besitzt zu den allgemeinen Eigenschaften noch weitere spezifische Eigenschaften, auf die hier nun näher eingegangen wird:

- Ausgabe: Reines Ausgabefeld, wo keine Daten eingeben werden können.
- **Textfeld** und **Passwortfeld**: Eingabefelder, wo Daten eingegeben werden können. Die Angabe der *Spalten* gibt die Größe des Textfeldes an.
- Text Area: Die Angabe von *Spalten* und *Zeilen* gibt die Breite und Höhe der Text Area an.

🅘 Feldeigenschaften - M	ozilla Firefox		_ 🗆 🗙
http://localhost:8380/wf/	'servlet.method/com.gr	oiss.avw.html.HTMLFormBuilder.editField?row=11	☆ 🛞
Feldeigenschaften			
Text 🔨	Beschriftung:	Ausbildung	
Formularielder Ausgabe Textfeld	CSS-Klasse:	label100	
Passwortfeld Text Area	Datenbankfeld: Darstellung:	education 💌 🔿 Radio Buttons 💿 Select List 🔿 Dropdown List	
Auswahlliste Mehrfachauswa	Zeilen:	5	FormBuilder.editField?row=11
labelle	wene.	Pflichtschule	
		Bachelor Master	
		Doktor	
M	Werteliste		
		Ändern Abb	rechen
	Werteliste	Master Doktor	rechen .:

Abbildung 6.11: Form-Wizard: Feldeigenschaften

- Auswahlliste: Hier kann bei der *Darstellung* im Formular zwischen *Radio-Buttons*, *Dropdown-List* und *Select-List* ausgewählt werden. In der Liste *Werte* müssen Sie dabei mindestens 1 Eintrag angeben, der als Auswahlpunkt im Formularfeld angezeigt werden soll oder Sie haben die Möglichkeit die Id einer *Werteliste* anzugeben. Eine Werteliste kann im DMS erzeugt werden (Formular *Valuelist*), die eine Id besitzen muss und die dazugehörigen Werte. Ist die Darstellung *Select-List* ausgewählt und im Feld *Zeilen* kein ganzzahliger Wert eingetragen, wird standardmäßig eine Dropdown-List angezeigt.
- Mehrfachauswahl: Analog zu *Auswahlliste*, jedoch kann hier in der Darstellung zwischen *Checkboxes* und *Select-List* gewählt werden.

Informationen des Reiters Tabellenansicht in Schritt 3:

In diesem Reiter können Sie festlegen, welche Spalten angezeigt werden, wenn das Formular als Subformular (Subtabelle) in einem anderem Formular eingebettet ist. In der Liste *Verfügbare Spalten* sind alle Formularfelder aufgelistet, die verfügbar sind. Durch Auswählen eines Eintrags und Aktivieren des *Hinzufügen*-Buttons, wird das Feld in die Liste *Tabellenspalten* übernommen. Alle Felder, die in dieser Liste vorhanden sind, werden im (übergeordneten) Formular als Subtabellen-Spalten angezeigt. Mit Hilfe der Sortierbuttons neben der Liste *Tabellenspalten* kann die Reihenfolge verändert werden.

6.4. FORMULARE

Wenn Sie alle Ihre Einstellungen getätigt haben, aktivieren Sie die Schaltfläche **Weiter**, um zum nächsten Schritt zu gelangen.

Schritt 4

🕹 @enterprise: F	ormularassistent	- Mozilla Firefox		_ 🗆 🔀
http://localhost	:8380/wf/servlet.meth	od/com.groiss.avw.html.HTMLFormBuild	er.createOrUpdateForm	☆ 🛞
Tabellendefi	nition	Formularde	finition	1 2 3 <mark>4</mark> 5
Formular-Id: car	ndidature Name	: Bewerbungsformular Version	:5	
Sortiert nach:	oid		Q	
Namensattribut:	notes			
Tabelle:	create table form_ oid %OIDTYPE task %OIDTYPI transactionId % acl %OIDTYPE awwcreatedBy % awwchangedBy awwchangedAt xformsinvalid DI notes VARCHA employed decim	candidature_5(% not null primary key, E%, OIDTYPE%, %, 60IDTYPE%, 00IDTYPE%, %OIDTYPE%, %DATE%, ECIMAL(1), R(4000), nal(1),		
Klasse:	xformsinvalid DECIMAL(1), notes VARCHAR(4000), employed decimal(1), Feldname Typ			
	awchangedBy %OIDTYPE%, awwchangedAt %DATE%, xformsinvalid DECIMAL(1), notes VARCHAR(4000), employed decimal(1), Feldname Typ notes java.lang employed boolean	java.lang.String		
	lasse: Feldname moles moles Feldname moles mo	boolean		
	position	long	Intril.HTMLFormBuilder.createOrUpdateForm Formulardefinition 1 2 3 4 5 nular Version: 5 y key, String String String String String Fertig stellen Zurück Abbrechen	
	education	java.lang.String		
	candidate	java.lang.String		
	assessor	com.groiss.org.User		L 2 3 4 5
	assessor2	com.groiss.org.User		
	ou	com.groiss.org.OrgUnit		
	Ladata		Fertig stellen Zurück	Abbrechen

Abbildung 6.12: Formularassistent: Schritt 4

Dieser Schritt zeigt Ihnen die folgenden Informationen:

- Sortiert nach: Nach diesem Attribut werden die Einträge in der Tabelle, die in einem Hauptformular ein Subformular repräsentiert, sortiert.
- Namensattribut: Der Inhalt dieses Feldes wird an den entsprechenden Stellen (z.B. Auswahldialog) als Formularname angezeigt. Weiters kann eine Regular Expression angegeben werden, z.B. {formfield} (display_text), {formfield2}. Die geschwungene Klammer ist notwendig, um beim Anzeigen von mehreren Werten die Formularfelder

zu identifizieren, d.h. vorheriges Beispiel könnte folgende Ausgabe erzeugen: Hans (Vorname), Moser. *Hans* und *Moser* sind Werte aus dem Formularfeld, *Vorname* ist ein selbst eingegebener Text (= display_text).

- Tabelle: Hier wird das Datenbankstatement angezeigt, das abgesetzt wird, um die Tabelle für das neue Formular anzulegen.
- Klasse: Hier werden die Felder und deren Typen jener Java-Klasse angezeigt, durch die das Formular im System repräsentiert wird.

Durch das Aktivieren der Schaltfläche *Fertig stellen* wird die Tabelle und die entsprechende Java–Klasse erzeugt und das Formular im System angelegt. Die Java–Klasse wird **immer** im Verzeichnis *forms* von @**enterprise** abgelegt (z.B. com.dec.avw.appl.myform_1). Das Formular selbst wird bei der Erstellung auch im *forms*-Verzeichnis gespeichert. Bei der Erstellung in einer Applikation mit einem festgelegten Applikationsverzeichnis (siehe Kapitel 6.1), wird das Formular im *classes*-Verzeichnis des Applikationsverzeichnisses gespeichert und nicht im *forms*-Verzeichnis.

Schritt 5

1

Im letzten Schritt des Assistenten wird Ihnen angezeigt, ob das Formular erfolgreich geladen werden konnte. Das Laden eines neuen Formulars ist damit abgeschlossen.

6.4.2 Formular bearbeiten

Nachdem Sie diese Funktion aktiviert haben, öffnet sich der Assistent zum Erstellen und Bearbeiten von Formularen. Dieser Assistent entspricht dem Assistenten zum Erstellen eines neuen Formulars, startet aber hier erst im Schritt 3 bei der Definition der Formularfelder.

HTML ersetzen

Nach dem Sie diese Funktion aktiviert haben, öffnet sich der Assistent zum Erstellen und Bearbeiten von Formularen. Dieser Assistent entspricht dem Assistenten zum Erstellen eines neuen Formulars, startet aber hier erst im Schritt 2 bei der Auswahl der HTML-Datei.

Mit dieser Funktion können Sie die Gestaltung eines Formulars, das bereits in Verwendung ist, ersetzen. Es wird dazu einfach eine HTML Datei geladen, die das bestehende Template ersetzt.

Wenn das neue HTML–Formular keine neuen bzw. weniger Felder als das alte besitzt, wird die HTML–Seite *Schritt 3* übersprungen. Durch das Aktivieren der Schaltfläche *Weiter* wird das Ersetzen des HTML–Formulars abgeschlossen. Die Java–Klasse und die Datenbanktabelle des Formulars werden bei dieser Operation nicht verändert.

Wenn das neue HTML–Formular auch neue Felder besitzt, wird das vom System erkannt und die HTML–Seite *Schritt 3* angezeigt. Durch das Aktivieren der Schaltfläche *Fertig stellen* wird die Java–Klasse und die Datenbanktabelle angepasst und das Ersetzen des HTML– Formulars ist damit abgeschlossen. Falls Ihr Formular auf eine XHTML-Datei verweist, ist HTML ersetzen nur notwendig, wenn neue Felder hinzugefügt werden.

View erzeugen

In @enterprise haben Sie im Rahmen der *Erweiterten Suche* die Möglichkeit unabhängig von der Formularversion eines Formulars nach Inhalten in dessen einzelnen Formularfeldern zu suchen. Damit diese Funktion genutzt werden kann, muss ein Datenbank–View über alle Formularversionen dieses Formulars erzeugt werden. Dieser Datenbank–View enthält all jene Formularfelder, die in allen Formularversionen vorkommen.

Nach dem Aktivieren der Funktion *View erzeugen* öffnet sich eine neue HTML-Seite mit den folgenden Informationen:

- 1. View erzeugen: *formularId*: Hier ist das SQL–Statement zu sehen, das in der Datenbank abgesetzt wird, um den Datenbank–View zu erzeugen.
- 2. View erzeugen: Ist zum ausgewählten Formular noch kein Datenbank-View vorhanden, ist diese Schaltfläche sichtbar. Nach dem Sie diese aktiviert haben, wird der Datenbank–View erzeugt.
- 3. **Bestehenden View ersetzen:...:** Ist zum ausgewählten Formular bereits ein Datenbank-View vorhanden, ist diese Schaltfläche sichtbar. Nach dem Sie diese aktiviert haben, wird der bestehende View durch den Neuen ersetzt.

Anzeigen

۲

Wenn Sie die Funktion Anzeigen aktivieren, öffnet sich die Objekt-Detailansicht des ausgewählten Formulars. Allerdings befinden Sie sich sofort auf der Registerkarte Vorschau.

6.4.3 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Id: Die eindeutige Kurzbezeichnung des Formulars (kann nicht mehr geändert werden). Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Der Name des Formulars. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- Version: Die Versionsnummer des Formulars (kann nicht mehr geändert werden). Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Applikation: Gibt an, zu welcher Applikation das Formular gehört.

🥹 @enterprise - Mozilla Firefox	- 🗆 🗙
http://localhost:8380/wf/servlet.method/com.dec.avw.html.HTMLFormType.showView?&node=admin.formtyp	Def 🚖 🛞
Datenbank-View über alle Formularversionen mit der Id 'candidature' erze	ugen
View erzeugen: form_candidature	
Create view form_candidature as select oid,candidate,assessor,ou,assessor2,notes,position,desiredsalary,ddate,givensalary,educati on,cdate,employed from form_candidature_2 union select oid,candidate,assessor,ou,assessor2,notes,position,desiredsalary,ddate,givensalary,educati on,cdate,employed from form_candidature_1 union select oid,candidate,assessor,ou,assessor2,notes,position,desiredsalary,ddate,givensalary,educati on,cdate,employed from form_candidature_3	
View erzeugen Abbrechen	

Abbildung 6.13: Funktion: View erzeugen

- **Typ:** Der Typ bestimmt in welchem Kontext das Formular verwendet werden soll (kann nicht mehr geändert werden). Folgende Typen werden unterschieden:
 - *Prozessformular:* Das Formular wird an einen Prozess angehängt, und kann bei den entsprechenden Tasks verwendet werden.
 - *Dokumentenformular:* Dieses Formular wird verwendet, um im DMS die Metadaten von Dokumenten zu beschreiben.
 - *Ordnerformular:* Dieses Formular wird verwendet, um im DMS die Metadaten von Ordnern zu beschreiben.

6.4. FORMULARE

	vacati	on (Vacation 1) - @e	nterprise	- Mozilla Firefox						
localhost:8280/ja	avaclient	/servlet.method/com.gro	iss.storegui.	TabbedWindow.showDialog?noc	e=admin.forr	ntype&fore	eignKey=applic	ation&application=com.dec	.avw.core.Application:4	2951: 🏠
lgemein Javak	lasse	Datenbanktabelle	Rechte	Standardberechtigungen	Historie	Zugriff	Vorschau	Ordnereinstellungen	Referenziert von	
ld: h	ir_vacat	tion								
Name: V	/acation	n							I18n: Vacat	ion
Version: 1										
Applikation: F	Person	alprozesse	~							
Typ:		sformular 🔽								
Maskentyp: 🔉	KFORM	1S 🔽								
Beschreibung: F	orm for	r vacations.								
Aktiv:	~									
Aktiv:	☑ dhar [1								
Aktiv: E m DMS verwend Versionierung:	dbar:] icht automatisch								
Aktiv: E m DMS verwenc Versionierung: Sortierattribute:	dbar: [n] icht automatisch	~							
Aktiv:	dbar: [n oi e: er] icht automatisch d mplovee								2
Aktiv: m DMS verwend /ersionierung: Sortierattribute: Namensattribute Suchattribute:	dbar: [n oi e: er] icht automatisch d mployee mployee	×							Q Q Q
Aktiv: m DMS verwenc /ersionierung: Sortierattribute: Namensattribute Suchattribute: con:	dbar: n oi er er] icht automatisch d mployee mployee	v							
Aktiv: m DMS verwenc Versionierung: Sortierattribute: Namensattribute: Suchattribute: con: Event-Handler:	dbar: n oi er er] icht automatisch d mployee mployee	M							
Aktiv: m DMS verwenc Versionierung: Sortierattribute: Namensattribute Suchattribute: con: Event-Handler: Basisformular:	dbar: n oi e: er	icht automatisch d mployee mployee								
Aktiv: Im DMS verwenc Versionierung: Sortierattribute: Namensattribute: Icon: Event-Handler: Basisformular: KHTML-Datei:	dbar: n oi er er	icht automatisch d mployee mployee forms/vacation.html								
Aktiv: E Im DMS verwenc Versionierung: Sortieratribute: Namensattribute: Icon: Event-Handler: Basisformular: HTML-Datei: Breite:	dbar: [n oi e: er fr	icht automatisch d mployee mployee forms/vacation.html								
Aktiv: E Im DMS verweng: Versionierung: Sortieratribute: Namensattribute: Icon: Event-Handler: Basisformular: XHTML-Datei: Breite: Höhe:	dbar: [n oi er er hr 0 0	icht automatisch d mployee mployee								
Aktiv: E Im DMS verwenc Versionierung: Sortierattribute: Namensattribute: Icon: Event-Handler: Basisformular: XHTML-Datei: Breite: Höhe:	dbar: n oi er er hr 0									
Aktiv: E m DMS verwenc Versionierung: Sortierattribute: Vamensattribute: con: Suchattribute: con: Suchattribute: Suchattribute: Basisformular: (HTML-Datei: Breite: Höhe:	dbar: n oi er er hr 0	cht automatisch d mployee mployee							(HTML do	₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽
Aktiv: E m DMS verwend /ersionierung: Sortierattribute: Suchattribute: con: Event-Handler: Jasisformular: dHTML-Datei: Breite: Höhe:	dbar: n oi er er hr 0	cht automatisch d mployee mployee							(HTML do	₽ ₽ ₽ ₩mloade

Abbildung 6.14: Objekt-Detailansicht: Formulare

- **Maskentyp:** Der Maskentyp bestimmt, welches Format für das Formular verwendet wird. Folgende Typen werden unterschieden:
 - HTML
 - XHTML
 - XFORMS
- Beschreibung: Freier Text.
- Aktiv: siehe Kapitel 2.2.1
- Im DMS verwendbar: Ist diese Checkbox angekreuzt, kann das Formular im DMS verwendet werden.
- Versionierung: Hier können Sie angeben, wann eine Version von einem Formular erstellt werden soll. Diese Einstellung gilt nur für Formulare, die im DMS verwendet werden können, und wird nur wirksam, wenn das Formular nicht als *Prozessformular* verwendet wird. Im letzteren Fall erfolgt bei jedem Weiterleiten automatisch eine Versionierung des Prozessformulars. Es stehen Ihnen vier Möglichkeiten für die Konfiguration der Versionierung zur Verfügung:
 - gemäß Konfiguration: Ist diese Option ausgewählt, erfolgt die Versionierung

so, wie es in der Serverkonfiguration unter dem Punkt *DMS* (siehe Installationshandbuch) eingestellt ist.

- nicht automatisch: Ist diese Option ausgewählt, muss der Benutzer das Formular explizit versionieren (siehe Funktion Version erzeugen im Benutzerhandbuch).
- bei Bearbeiterwechsel: Ist diese Option ausgewählt, erfolgt immer dann eine Versionierung des Formulars, wenn sich der Bearbeiter des Formulars geändert hat.
- bei jeder Änderung: Ist diese Option ausgewählt, erfolgt bei jeder Änderung des Formulars eine Versionierung.
- Sortierattribute: Wählen Sie hier jenes Attribut aus, nach dem sortiert werden soll.
- Namensattribute: Der Inhalt der hier ausgewählten Attribute wird als Namensattribut verwendet. Weiters kann eine Regular Expression angegeben werden, z.B. {form-field} (display_text), {formfield2}. Die geschwungene Klammer ist notwendig, um beim Anzeigen von mehreren Werten die Formularfelder zu identifizieren, d.h. vorheriges Beispiel könnte folgende Ausgabe erzeugen: Hans (Vorname), Moser. *Hans* und *Moser* sind Werte aus dem Formularfeld, *Vorname* ist ein selbst eingegebener Text (= display_text).
- Suchattribute: Hier können Sie angeben, nach welchen Attributen gesucht werden soll. Dies wird unter anderem wirksam, wenn man in einem DMS Ordner die Kurzsuche im Toolbar dazu konfiguriert (siehe Abschnitt 6.4.9) und über das Suchfeld nach einem bestimmten Begriff sucht. Wird kein Suchattribut festgelegt, wird standardmäßig der Name des DMS Objekts für die Suche herangezogen.
- Icon: Das hier eingetragene Icon wird im Zusammenhang mit dem Formular angezeigt.
- Event-Handler: Hier kann eine Klasse eingetragen werden, die das Java-Interface com.groiss.dms.FormEventHandler oder com.groiss.dms.XHTMLFormEventHandler implementiert. Damit hat der Applikationsprogrammierer die Möglichkeit auf bestimmte Ereignisse (*onDelete, onInsert, onShow* und *onUpdate*) im Zusammenhang mit der Formularbearbeitung reagieren zu können.
- Basisformular: Dieses Feld zeigt den Namen des Basisformulars an, von dem das aktuelle Formular abgeleitet wurde.
- XHTML-Datei: Referenz zu einer XHTML Datei im Klassenpfad
- Breite und Höhe: Spezifiziert die Größe der zu öffnenden HTML Seite

In dieser Ansicht stehen Ihnen die Funktionen 118n und Aktivieren zur Verfügung.

Wenn Sie die Schaltfläche **HTML downloaden** aktivieren, können Sie das aktuelle Formular als HTML-Datei lokal auf den Rechner abspeichern. Dort kann es von Ihnen bearbeitet, und anschließend durch die Funktion *HTML ersetzen* (siehe Kapitel 6.4.2) wieder in das System geladen werden.

🕹 Formulare: hr_vacation (Vacation 1) - @e	nterprise -	Mozilla Firefo	x					
http://localhost:8380/wf/servlet.method/com.grois	s.storegui.Tab	bedWindow.show	/Dialog?node=a	dmin.formty	pe&foreigr	Key=applicatio	n&application=com.dec.av	w.core.Application:429665! 🏠
Allgemein Javaklasse Datenbanktabelle	Rechte	Standardbere	chtigungen	Historie	Zugriff	Vorschau	Ordnereinstellungen	Referenziert von
com.dec.avw.appl.hr_vacation_1								
Id							Тур	
approved			boolean					
approvedby			com.groiss.	org.User				
comments			java.lang.St	ring				
days			int					
employee			com.groiss.	org.User				
notapprovedreason			java.lang.St	ring				
0U			com.groiss.	org.OrgUn	it			
substitute			com.groiss.	org.User				
type			java.lang.St	ring				
vacfrom			java.util.Dat	е				
vacto			java.util.Dat	е				
Anzahl der Einträge: 11								
Java-Klassen neu generieren								
Löschen						Ok	Abbreche	en Übernehmen

Abbildung 6.15: Formular-Details: Javaklasse

6.4.4 Registerkarte: Javaklasse

Diese Registerkarte zeigt eine Auflistung aller Formularfelder des aktuellen Formulars (Name und Typ). Mit Hilfe der Schaltfläche *Java-Klassen neu generieren* kann eine neue Java-Klasse erzeugt werden, auch wenn keine bestehende Klasse vorhanden ist. In diesem Fall wird eine neue Klasse im *forms*-Verzeichnis der **@enterprise** Installation gespeichert. Bei einer bereits vorhandenen Klasse wird die bestehende ersetzt.

6.4.5 Registerkarte: Datenbanktabelle

Diese Registerkarte zeigt das SQL-Statement für die Erzeugung der Tabelle für das Formular.

6.4.6 Registerkarte: Rechte

siehe Kapitel 5.1.4

6.4.7 Registerkarte: Standardberechtigungen

siehe Kapitel 5.1.4

6.4.8 Registerkarte: Vorschau

Der Inhalt dieser Registerkarte ist die Darstellung des Formulars.

6.4.9 Registerkarte: Ordnereinstellungen

Ist dieses Formular ein Ordnerformular, so kann man hier das Design der Ordnerinhalttabelle verändern (siehe Abbildung 6.17). Es besteht die Möglichkeit

❷ Formulare: hr_vacation (Vacation 1) - @enter	rprise - I	Mozilla Firefox						_ 🗆 🗙
http://localhost:8380/wf/servlet.method/com.groiss.sto	oregui. Tabl	bedWindow.showDialog?node=a	admin.formtyp	e&foreignl	<ey=application< th=""><th>n&application=com.dec.avv</th><th>.core.Application:429</th><th>9665! 🏠 🛞</th></ey=application<>	n&application=com.dec.avv	.core.Application:429	9665! 🏠 🛞
Allgemein Javaklasse Datenbanktabelle F	Rechte	Standardberechtigungen	Historie	Zugriff	Vorschau	Ordnereinstellungen	Referenziert von	
<pre>create table form hr_vacation_1(oid DECIMAL(20) not null primary task DECIMAL(20), transactionId DECIMAL(20), acl DECIMAL(20), avwcreatedBy DECIMAL(20), avwcreatedBy DECIMAL(20), avwchangedBy DECIMAL(20), avwchangedBy DECIMAL(20), avwchangedBy DECIMAL(20), avwchangedBy DECIMAL(20), avwchangedBy DECIMAL(20), avwchangedBy DECIMAL(20), employee DECIMAL(20), employee DECIMAL(20), employee Class varchar(100), vactro DATE, vacto DATE, days decimal(10), comments VARCHAR(500), substitute_less varchar(100), approvedby DECIMAL(20), approvedby DECIMAL(20), notapprovedreason VARCHAR(500), "TYPE" VARCHAR(10))</pre>	, key,							
Löschen					Ok	Abbreche	n Überna	ahmen

Abbildung 6.16: Formular-Details: Datenbanktabelle

- Spalten hinzuzufügen, zu ändern, zu löschen bzw. die Anordnung zu verändern
- Funktionen hinzuzufügen, zu löschen bzw. die Anordnung zu verändern
- Formulare hinzuzufügen, zu löschen und deren Verwendung zu setzen (*Erlaubt* bzw. *Nicht erlaubt*)

Näheres dazu finden Sie im *Benutzerhandbuch* von @enterprise. Diesbezügliche Änderungen, die beim Stammdatenobjekt selbst vorgenommen werden, gelten für alle Instanzen dieses Objekts.

6.5 Prozesse

Ein Prozess beschreibt den Ablauf eines gesamten Geschäftsprozesses. Sie können Prozesse in @enterprise über den grafischen Prozesseditor (siehe Kapitel 7.2) oder in WDL (siehe Kapitel 7.1) erstellen.

In der Werkzeugleiste stehen Ihnen neben den in Kapitel 2.1 beschriebenen Funktionen, folgende Funktionen zusätzlich zur Bearbeitung von Prozessen zur Verfügung:

- Neuen Prozess mit Prozesseditor erstellen
- Prozess mit Prozesseditor bearbeiten

6.5. PROZESSE

onnatare	. stanuarur	order (orditer 1) -	wenterpri	se - mozina i fielox						
localhost:82	80/javaclient/s	ervlet.method/com.grois	s.storegui.Ta	abbedWindow.showDialog?node	admin.form=	type&foreig	gnKey=applical	ion&application=com.dec.a	vw.c 🏫 🛞	÷
lgemein	Javaklasse	Datenbanktabelle	Rechte	Standardberechtigungen	Historie	Zugriff	Vorschau	Ordnereinstellungen	Referenziert	t von
Ordnerei	nstellunge	n								
Spalten:	Nam Zusa Typ Größ Zulet gesp	e tzdaten Se zt geändert am errt durch		+ ∥ X t						
Funktione	en: new spac cut copy link pasti	e		+ × ₽ ₽						
Formular	e: ⊙ E	Erlaubt 🔘 Nicht erla	aubt	×						
Paging: Tabelleneir	nträge pro Se	eite:								
tandard wi	iederherstell	en								

Abbildung 6.17: Formular-Details: Ordnereinstellungen

- WDL / XWDL laden
- Prozessübersicht

Die Objekt-Detailansicht von Prozessen gliedert sich in die folgenden Bereiche:

- Allgemein
- Source
- Graph. Darstellung
- Komponenten
- Formular-Sichtbarkeiten
- Eskalation
- Funktionen
- Historie
- Zugriff
- Ordnereinstellungen
- Referenziert von

Ğ.

ËÞ.

6.5.1 Neuen Prozess mit Prozesseditor erstellen

Durch das Aktivieren dieser Funktion wird der Prozesseditor gestartet. Damit können Sie einen neuen Prozess erstellen und diesen abspeichern. Der Prozesseditor wird in Kapitel 7.2 beschrieben.

6.5.2 Prozess mit Prozesseditor bearbeiten

Durch das Aktivieren dieser Funktion wird der Prozesseditor gestartet. Der aktuell selektierte Prozess wird dargestellt, und kann bearbeitet werden.

6.5.3 WDL / XWDL laden

Mit Hilfe dieser Funktion können Sie Prozesse, deren WDL–Definition bzw. XWDL-Definition in einer Datei vorliegt, in das System laden und damit einen neuen Prozess erstellen.

🥹 @enterprise - Scr	🤌 @enterprise - Script kompilieren - Mozilla Firefox 📃 🗖							
http://localhost:838	📸 http://localhost:8380/wf/servlet.method/com.dec.avw.html.HTMLProcessDefinition.showLoadMask?&node=admin.procdefinition&foreignKey=applica 🏠 🛞							
Script kompilie	ren							
WDL 💌 Datei: jo	obproc.wdl	ja la						
	Tasks generieren							
	Rollen generieren							
			Kompilieren Abbrechen					
		_						

Abbildung 6.18: Script laden

Nach dem Aktivieren der Funktion öffnet sich ein HTML-Dialog (siehe Abbildung 6.18) mit den folgenden Informationen:

- **Datei** bzw. **Pfad:** Geben Sie hier den vollständigen Pfad der Datei an, die die zu ladende Prozess–Definition enthält. Mit Hilfe der Schaltfläche *Durchsuchen* wird das Auffinden der Datei erleichtert. Zuvor muss aus der Dropdown-Liste ausgewählt werden, ob ein WDL-Skript geladen werden soll.
- Tasks generieren: Ist diese Checkbox angekreuzt, werden alle Tasks, die im WDL-Script verwendet werden, aber dem System noch nicht bekannt sind, erzeugt.
- Rollen generieren: Ist diese Checkbox angekreuzt, werden alle Rollen, die im WDL-Script verwendet werden, aber dem System noch nicht bekannt sind, erzeugt.

Durch das Aktivieren der Schaltfläche **Kompilieren** wird das Übersetzungsprogramm (der Compiler) gestartet. Danach meldet das System, ob der Prozess eingetragen wurde oder ob Fehler aufgetreten sind.

۲

6.5.4 Prozessübersicht

Diese Funktion zeigt eine Übersicht über alle verwendeten Komponenten des Prozesses (siehe Abbildung 6.19). Die einzelnen Komponenten werden in Blöcke gegliedert und können auf- bzw. zugeklappt werden. Über diese Darstellung ist es möglich referenzierte Objekte (z.B. Tasks, Rollen, etc.) in der Detailansicht anzeigen zu lassen und diese zu ändern. Weiters besteht die Möglichkeit ein PDF zu generieren.

🕙 Prozessübersicht - Mo	zilla Firefox	[
http://localhost:8380/wf/	servlet.method/com.groiss.avw.html.ProcessOverview.show?node=admin.procdefinition&foreignKey=application&application	plicati 🏫	7 🛞					
Prozess: Urlaub		1	^					
🗶 1. Allgemein								
Name	Urlaub							
Applikation	Personalprozesse							
Beschreibung	Über diesen Prozess können Urlaube beantragt und genehmigt werden. Ein Antrag wird immer zu jeweiligen Vorgesetzten weitergeleitet, welche/r den Antrag genehmigen oder ablehnen kann.	um/r						
ld (Version)	hr_vacation (1)							
Übersetzungen	Deutsch: Urlaub, Englisch: Vacation							
Formulare	ld: form_vacation, Name: Urlaub, Typ: Vacation							
2. 1102								
process hr_vacation	n(form_vacation vacation "form_vacation")							
version 1; name "proc vacation								
description "desc p	proc vacation";							
subject "{form_vaca	ation.employee}, {form_vacation.vacfrom,date} - {form_vacation.vacto,da	te}";						
timeoutaction none;	-0781							
begin "New"								
all vacation_req	<pre>guest(form_vacation);</pre>							
manager vacation	<pre>1_approve(form_vacation); form_vacation/enproved = ""true""" then "Vacation enproved"</pre>							
andpar "New"	Torm_vacacion/approved = crue chen vacacion approved							
form_vacat	ion.employee vacation_informapproved(form_vacation);							
hr home we	vetion process/form vection);							
end "New";	cacion_process(iorm_vacacion);							
else								
form_vacation	<pre>n.employee vacation_informrefused(form_vacation);</pre>							
end "New"								
💌 3. Graph. Darste	llung							
C								
Ĭ								
¥								
	(1)							
Alle Anfo	raern							
¥	(2)							

Abbildung 6.19: Prozess Übersicht

6.5.5 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

🕑 Prozesse: l	Jrlaub (1)	- @enterprise - Moz	illa Firefox							
🐞 http://local	host:8380/w	f/servlet.method/com.gro	iss.storegui.TabbedV	/indow.showDialog?node=admin	.procdefinition&	foreignKey=appl	ication&applic	ation=com	.dec.avw.core.Application:	4296659231& 🏠 🌘
Allgemein	Source	Graph. Darstellung	Komponenten	Formular-Sichtbarkeiten	Eskalation	Funktionen	Historie	Zugriff	Ordnereinstellungen	Referenziert von
ld:		hr_vacation								
Name:		proc_vacation							118	3n: Urlaub
Version:		1								
Applikatio	n:	Personalprozesse	*							
Betreff:		{form_vacation.emplo	oyee}, {form_vaca	tion.vacfrom,date} - {form_v	acation.vacto	,date}				
Instanz-Id:										
Priorität:		0								
Orthogonale	es Routing	:								
Becchroibu	eichnen:	D T TT 1								
Descriteibu	iy.		» × 🚑 ×							
		uesc_proc_vacation								
Max. Dauer		D Tage C	Stunden 🔿 M	nuten						
Aktiv:										
Änderung d	urchführen	am:	1999 1911							
										())
Loscher	1						Uk		Abbrechen	Ubernehmen

Abbildung 6.20: Objekt-Detailansicht: Prozesse

- Id: Die eindeutige Kurzbezeichnung des Prozesses. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Name: Der Name des Prozesses. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1. Der I18n-Link neben diesem Feld ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9).
- Version: Die Versionsnummer des Prozesses. Genauere Informationen zu Ids und Namen finden Sie im Kapitel 3.1.
- Applikation: Die Applikation, in der der Prozess läuft.
- Betreff: Formularfeld oder Betrefftext, das/der zur Laufzeit der Prozessinstanz angezeigt wird. Dabei kann weiters ein regulärer Ausdruck angegeben werden.
- Instanz Id: Hier kann die Instanz Id festgelegt werden, die jede gestartete Prozessinstanz eindeutig identifiziert. Hierbei besteht auch die Möglichkeit einen regulären Ausdruck im folgenden Format anzugeben:

```
{ letter* "{" ("n" | "nn" | "ny" | "nny" | "y" | "yy" | "ou") "}" letter* }*
```

Erklärung:

- n.....nächste Nummer

- nn....nächste Nummer für diesen Prozess
- ny....nächste Nummer pro Jahr
- nny...nächste Nummer für diesen Prozess für dieses Jahr
- y.....Jahr mit 2 Stellen
- yy....Jahr
- ou....Organisationseinheit

Hinweis: Falls die Instanz Id Leerzeichen enthält und der Parameter *webdav.show.subject* (siehe Installationshandbuch - Kapitel *Parameters without GUI*) in der Konfigurationsdatei *avw.conf* auf 1 gesetzt wurde, können DMS-Dokumente, die der Prozessinstanz angehängt wurden, nicht geöffnet werden und folgender Fehler wird angezeigt: Fehler 1002: Das Dokument konnte nicht gefunden werden.

- Priorität: Die Priorität des Prozesses.
- Orthogonales Routing: Wenn diese Checkbox aktiviert ist, werden die Kanten des Prozesses im Prozesseditor orthogonal dargestellt.
- In BMPN zeichnen: Wenn diese Checkbox aktiviert ist, wird der Prozess in BPM-Notation geöffnet. Wenn ein in WD-Notation gezeichneter Prozess im Prozesseditor geöffnet ist und diese Checkbox aktiviert wird, dann wird der aktuelle Benutzer gefragt, ob er die Notation ändern will. Bestätigt der Benutzer dies, so wird der Prozess von der WD-Notation in die BPM-Notation konvertiert. In diesem Fall ist zu beachten, dass der Buffer für die Funktionen Rückgängig/Wiederherstellen und Ausschneiden/Kopieren/Einfügen im Prozesseditor geleert wird! Nähere Informationen dazu finden Sie im Kapitel 7.
- Beschreibung: Freier Text. @enterprise bietet hierbei die Möglichkeit, den Text in diesem Feld entsprechend zu formatieren wie es z.B. aus Microsoft Word bekannt ist.
- Max. Dauer: Hier bestimmen Sie die maximale Bearbeitungszeit des Prozesses in Tagen, Stunden oder Minuten.
- Aktiv: siehe Kapitel 2.2.1
- Änderung durchführen am: siehe Kapitel 2.2.1

In dieser Ansicht steht Ihnen weiters die Funktion 118n zur Verfügung.

6.5.6 Registerkarte: Source

In der Registerkarte *Source* wird das WDL–Script des ausgewählten Prozesses angezeigt (siehe Abbildung 6.21).

Wenn Sie die Schaltfläche **XWDL anzeigen** aktivieren, öffnet sich ein neues Fenster, in dem die XWDL-Definition des Prozesses angezeigt wird. Wenn Sie die Schaltfläche **XWDL downloaden** aktivieren, können Sie die XWDL-Definiton in Ihrem Dateisystem abspeichern.

Drozosso: Urlaub (1	Prozesse: Urlaub (1) - Genterprise - Mozilla Firefox								
Prozesse. Oriaub (1) - wenterprise - mozi	nia i fretox							
http://localhost:8380/	wf/servlet.method/com.groi	ss.storegui.TabbedV	/indow.showDialog?node=admir	proceefinition&	foreignKey=appli	cation&applic	ation=com	.dec.avw.core.Application:	42966592318fu 🏫 🛞
Allagmain Source	Graph Darctallung	Komponenten	Formular Sighthorkoiton	Eckolation	Funktionen	Historia	Zugriff	Ordnoroinstellungen	Poforonziart von
Aligement Source	Graph, Darstellung	Komponenten	T official-Sichibarkeiten	Lokalation	T dirktionen	Thistone	Zugini	Orunereinstenungen	Referenzien von
process hr_vaca	tion(form_vacatio	n vacation '	form_vacation")						
version 1;	nian".								
description "de	sc proc vacation"								
subject "{form	vacation.employee	; }, {form vac	ation.vacfrom,date}	- (form v	acation.va	cto,date	:)";		
timeoutaction n	one;			_					
application sta	ffprocs;								
begin "New"									
all vacation	_request(form_vac	ation);							
if ynath."Sf	cion_approve(iorm	vacation);	""true""" then "Ver	etion ennr	oved"				
andpar "N	ew"	, approved	orac onen vao	deren appr	oveu				
form v	acation.employee	vacation inf	ormapproved(form va	cation);					
I - I		_							
hr ! hom	e vacation_proces	s(form_vacat	ion);						
end "New"	;								
else									
iorm_vaca	cion.employee vac	acion_inform	rerused(form_vacati	on);					
end "New"									
								XWDL anzeigen	WDL downloaden
Löschen						Ok		Abbrechen	Übernehmen

Abbildung 6.21: WDL-Anzeige

6.5.7 Registerkarte: Graph. Darstellung

In der Registerkarte zeigt die grafische Darstellung des Prozesses, wie sie im Prozesseditor aussehen würde.

6.5.8 Registerkarte: Komponenten

Diese Registerkarte zeigt Ihnen, welche Tasks, Formulare, Rollen, Prozesse und Webservices (Webservice Operationen und durch das Webservice importierte Dateien) mit der aktuellen Prozessversion gelinkt sind (siehe Abbildung 6.22). Durch Aktivieren eines Links innerhalb dieses Reiters (z.B. eines Tasks) wird wiederum ein neues Fenster mit der Detailansicht des ausgewählten Objekts geöffnet.

6.5.9 Registerkarte: Formular-Sichtbarkeiten

Die Registerkarte *Formulare* zeigt Ihnen eine Übersicht über alle, dem Prozess zugeordneten Formulare. Für jedes Formular wird eine weitere Registerkarte dargestellt, in dem die Sichtbarkeiten der Formularfelder in den einzelnen Tasks aufgelistet werden. In dieser Übersicht scheint ein Task nur auf, wenn ihm bei der Prozessdefinition ein Formular zugeordnet wurde.

Wenn Sie die Sichtbarkeit eines Formularfeldes in einem Task ändern wollen, aktivieren Sie den Link des entsprechenden Tasks. Dadurch wird die HTML–Seite Anzeigemodi von Formularfeldern angezeigt.

Informationen der HTML-Seite Anzeigemodi von Formularfeldern:

Prozesse: Urla	ub (1) - @enterprise ·	- Mozilla Firefox							
🐁 http://localhost	:8380/wf/servlet.method/co	om.groiss.storegui.Tab	bedWindow.showDialog]?node=admin.procdefin	tion&foreignK	ey=application&applic	ation=com	.dec.avw.core.Application:	4296659231& 🏠 🄇
Allaemein Sou	rce Graph. Darstellur	na Komponente	Formular-Sich	tbarkeiten Eskala	ion Funk	tionen Historie	Zuariff	Ordnereinstellungen	Referenziert von
								5	
Formulare:	ld Name	Version Besch	eibung						
	hr_vacation Vacatio	n 1 Form fo	or vacations.						
Tasks:	Id	Name	Version Beschrei	bung					
	vacation_approve	Genehmigen	1 desc_task	_approve_vacation					
	vacation_informappr	oved Genehmigt	1 desc_task	<_informapproved_va	cation				
	vacation_informrefus	sed Abgelehnt	1 desc_tas	<_informrefused_vac	ation				
	vacation_process	Verbuchen	1 desc_tas	_process_vacation					
	vacation_request	Anfordern	1 desc_task	_request_vacation					
Rollen:	ld Name E	Beschreibung							
	manager manager								
	all Alle								
	home Home								
Prozesse:	ld Name Version I	Beschreibung							
Webservice Operationen:	Webservice Opera	ntion Art							
Importierte Dateien:	Pfad								
Löschen						Ok		Abbrechen	Ubernehmen

Abbildung 6.22: Komponenten

Allgemein Sour	ce Graph. Darstellung	Komponenten	Formular-Sichtbarkeit	en Eskalation	Funktionen	Historie	Zugriff	Ordnereinstellungen	Referenziert v
vacation (form_	vacation)								
				Tasks					
	Anfordern	Ge	enehmigen	Abgelehnt		Geneh	migt	Verbuc	hen
type	rw	ro	r	0	ro			ro	
notapprovedreaso	on ro	rw	r	0	ro			ro	
approvedby	ro	man	r	0	ro			ro	
substitute	rw	rw	r	0	ro			ro	
comments	rw	ro	r	0	ro			ro	
days	rw	rw	r	0	ro			ro	
vacto	man	ro	r	0	ro			ro	
vacfrom	man	ro	r	0	ro			ro	
employee	man	ro	r	0	ro			ro	
ou	man	ro	r	0	ro			ro	
approved	ro	man	r	0	ro			ro	
approveblock	inv	rw	r	W	rw			rw	

Abbildung 6.23: Formularsichtbarkeiten

- Formulartyp: Die unter *Formularfeld* aufgelisteten Formularfelder beziehen sich auf dieses Formular.
- Task: Es handelt sich um ein Formular, das diesem Task zugeordnet ist.
- Prozess: Diesem Prozess ist der vorher genannte Task zugeordnet.
- Sichtbarkeit übernehmen von...: Hier kann die Sichtbarkeit von einem anderen Task in diesem Prozess übernommen werden, sofern es Unterschiede bzgl. Formular-Sichtbarkeiten zwischen den Tasks gibt.

- Formularfeld: Name des Formularfeldes, dessen Sichtbarkeit festgelegt werden soll.
- Unsichtbar: Ist dieser Radio-Button aktiviert, wird das Formularfeld nicht angezeigt.
- Lesen:
 - 1. Gesperrt: Ist dieser Radio–Button aktiviert, kann das Formularfeld nicht geändert werden und wird grau hinterlegt.
 - 2. Text: Ist dieser Radio–Button aktiviert, kann das Formularfeld auch nicht geändert werden, jedoch wird es nicht grau hinterlegt. Bei XForms ist diese Einstellung nicht wirksam, da XForms diese nicht unterstützen!
- Schreibbar:
 - 1. Optional: Ist dieser Radio-Button aktiviert, kann das Formularfeld geändert werden.
 - Muss: Ist dieser Radio-Button aktiviert, ist das Formularfeld änderbar und es muss befüllt werden.

Gibt es zudem ein Subformular, dessen Sichtbarkeit gesetzt werden sollte, gliedert sich die Information der HTML–Seite *Anzeigemodi von Formularfeldern* wie folgt:

- Formulartyp, Task, Prozess und Sichtbarkeit übernehmen von sind gleich.
- Tabelle: Name des Subformulars, dessen Sichtbarkeit festgelegt werden soll.
- Unsichtbar: Ist dieser Radio–Button aktiviert, wird die Tabelle des Subformulars nicht angezeigt.
- Nur Lesen: Ist dieser Radio-Button aktiviert, wird die Tabelle des Subformulars angezeigt, jedoch kann diese nicht verändert werden.
- Optional: Ist dieser Radio–Button aktiviert, wird die Tabelle des Subformulars angezeigt und kann über den Button *Neuer Tabelleneintrag* verändert werden.
- Kein Einfügen/Löschen: Ist dieser Radio-Button aktiviert, können keine weiteren Einträge der Tabelle hinzugefügt bzw. gelöscht werden.

Weiters gibt es die Möglichkeit die Sichtbarkeit eines Formularfeldes für alle Tasks festzulegen. Dafür müssen Sie auf den Link des entsprechenden Formularfeldes klicken und es öffnet sich die HTML–Seite *Anzeigemodi von Formularfeldern*. Diese ist ähnlich der HTML–Seite für Formularfelder, jedoch werden hier die Sichtbarkeiten für Tasks anstatt für Formularfelder festgelegt.

Durch das Aktivieren der Schaltfläche **Vorschau** öffnet sich ein neues Fenster, in dem das Formular bei Anwendung der aktuell gespeicherten Sichtbarkeiten dargestellt wird.

Durch das Aktivieren der Schaltfläche **Ok** werden Ihre Änderungen bezüglich der Sichtbarkeiten der Formularfelder gespeichert.

Durch das Aktivieren der Schaltfläche **Schließen** werden Ihre Änderungen verworfen und die HTML–Seite *Anzeigemodi von Formularfeldern* wird geschlossen.

6.5.10 Registerkarte: Eskalation

Eskalationsobjekte können Sie verwenden, um auf Zeitüberschreitungen bzw. das vorzeitige Erreichen eines Zeitpunktes bei der Ausführung von Prozesse reagieren zu können. Es ist möglich verschiedene Aktionen zu definieren, die bei einer Zeitüberschreitung ausgeführt werden sollen. Der @**enterprise**-Timer *Escalations* ist für die Prüfung der Zeitüberschreitungen verantwortlich. Läuft dieser Timer nicht, erfolgt auch keine Prüfung der Zeitüberschreitungen!

Die Registerkarte *Eskalation* zeigt zunächst alle zu diesem Prozess definierten Eskalationsobjekte. Sie können diese mit den Funktionen der Werkzeugleiste bearbeiten.

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Eskalationstyp: Hier können Sie zwischen folgenden Eskalationstypen wählen:
 - 1. **Prozess Sollzeit**: Diese Eskalation bezieht sich auf das Fertigstellungsdatum des Prozesses, das beim Prozessstart angegeben wurde.
 - Aktivitäts Sollzeiten: Analog zu Aktivität Sollzeit von Tasks, mit dem Unterschied, dass nicht jeder einzelne Task im Prozess mit dieser Eskalation belegt werden muss (siehe Abschnitt 6.2.2). Wenn dieser Eskalationstyp gewählt wurde, wird die Eskalation für jeden Task im Prozess ausgeführt.
 - 3. Batch nicht fertig: Dieser Typ bezieht sich auf nicht beendete Batch-Schritte in einem Prozess. Der Schritt wird über den Step-Label spezifiziert.
 - 4. Sync nicht erfolgt: Analog zu *Batch nicht fertig*, jedoch wird hierbei auf Sync-Schritte reagiert.
 - 5. Eingangsnachricht nicht erhalten: Dieser Typ bezieht sich auf nicht beendete RECEIVE-Schritte (= Eingangsnachricht) in einem Prozess. Der Schritt wird auch hier über den Step-Label spezifiziert.
- Schritt: Falls mehrere Schritte im Prozess den gleichen Task verwenden, können Sie hier den Schritt auswählen, auf den die Eskalation angewendet werden soll. Ein Schritt ist nur auswählbar, wenn der Eskalationstyp *Batch nicht fertig, Sync nicht erfolgt* bzw. *Eingangsnachricht nicht erhalten* ist.
- Zeitabstand: siehe Abschnitt 6.2.2
- Aktion: Siehe Abschnitt 6.2.2. Bei der Aktion *E–Mail senden* an aktuelle Akteure werden jedoch alle aktiven Tasks (ausgenommen jene von Subprozessen) herangezogen und an die Akteure eine E–Mail versendet. Bei der Aktion *Task starten* besteht jedoch der Unterschied, dass bei Prozesseskalationen immer Task und Rolle angegeben werden müssen.
- Beschreibung: siehe Abschnitt 6.2.2

Sie können auch weitere Eskalationen für jeden einzelnen Task in diesem Prozess erstellen (siehe Abschnitt 6.2.2).



Hinweis: Die Prozesseskalation stellt eine verbesserte Form der Timeout–Behandlung des Prozess–Editors dar und sollte daher bei neu erstellen Prozessen verwendet werden.

6.5.11 Registerkarte: Funktionen

Diese Maske ist analog zu 6.2.3 und bezieht sich auf den gesamten Prozess anstatt auf einzelne Tasks.

6.5.12 Registerkarte: Ordnereinstellungen

In diesem Reiter können Sie die DMS-Ordnereinstellungen für diesen Prozess vornehmen. Diese Funktion ist dabei äquivalent zu den Einstellungen, die bei Ordnerformularen getätigt werden können (siehe Abschnitt 6.4.9). Es besteht die Möglichkeit

- Spalten hinzuzufügen, zu ändern, zu löschen bzw. die Anordnung zu verändern
- Funktionen hinzuzufügen, zu löschen bzw. die Anordnung zu verändern
- Formulare hinzuzufügen, zu löschen und deren Verwendung zu setzen (*Erlaubt* bzw. *Nicht erlaubt*)

6.6 Interface

Mit Interfaces können Sie Prozesse durch Ausfüllen und anschließendes Abschicken eines Formulars starten. Die URL dafür sieht folgendermaßen aus:

http://host:port/wf/servlet.method/

com.groiss.storegui.FormWrapper.showProcessStartForm?formclass=<Formularklasse>

In der Interfaceliste stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen zur Verfügung.

Die Objekt-Detailansicht von Interfaces gliedert sich in die folgenden Bereiche:

- Allgemein
- Historie
- Zugriff

6.6.1 Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Formular: Das Formular das ausgefüllt werden muss, um den Prozess zu starten. Dieses Formular muss für den Prozess (in dessen Argumentliste) als Eingangsformular definiert sein.
- **Prozess:** Der Prozess, der gestartet werden soll.

6.7. FUNKTIONSGRUPPE

🎒 Interfaces: Neu - @ent	erprise - Mozilla Firefox 📃 🗖 🔀
http://localhost:8380/wf/	servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.procinterface&func=nev 🏫 🛞
Allgemein Historie	Zugriff
Formular:	hr_vacation (Vacation 1)
Prozess:	Urlaub (1)
Organisationseinheit:	Groiss Informatics
Aktiv:	
Antwortmaske:	
·	
Löschen	Ok Abbrechen Übernehmen

Abbildung 6.24: Objekt-Detailansicht: Interface

- Organisationseinheit: In dieser Organisationseinheit soll der Prozess gestartet werden.
- Aktiv: siehe Kapitel 2.2.1.
- Antwortmaske: Der HTML-Text, der dem Benutzer angezeigt wird, nachdem der Prozess gestartet wurde. Die Felder %pid% und %process% werden dabei durch die Prozessnummer und den Prozessnamen ersetzt.

Beispiel: <h3> Danke für Ihren Auftrag!</h3> Ihre Auftragsnummer ist %pid%.

• Änderung durchführen am: siehe Kapitel 2.2.1

6.7 Funktionsgruppe

Funktionsgruppen dienen dazu, um Task-Funktionen zu gruppieren. Die Funktionsgruppe besteht aus einer eindeutigen *Id* und einem frei vergebbaren *Namen* und muss einer Applikation zugeordnet sein. Der I18n-Link neben dem Feld *Name* ermöglicht direkt die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9). Weiters kann zu der Funktionsgruppe ein Sortierattribut angegeben werden, nach dem sortiert werden soll. Wenn nun mehrere Task–Funktionen verschiedene Gruppen angehören, werden diese im Arbeitskorb unter *Funktionen* gruppiert dargestellt. Funktionsgruppen können nur dann gelöscht werden, wenn sie keiner *Task-Funktion* bzw. einer *Gespeicherten Anfrage* zugewiesen wurden.

6.8 GUI-Konfiguration

Die GUI Konfiguration erlaubt es, Masken (Arbeitskorb, DMS, etc.) einfach anzupassen und die entsprechenden Berechtigungen über den Reiter *Zuordnungen* an Benutzer bzw. Rollen zu vergeben. Die Maskenelemente werden in XML-Dateien im *classes*-Verzeichnis der aktuellen @**enterprise** Installation bzw. im Applikationsverzeichnis der jeweiligen Applikation gespeichert. @**enterprise** bietet die Möglichkeit

- neue Masken zu erzeugen
- bestehende Masken zu bearbeiten bzw. zu löschen
- bestehende Masken zu kopieren

6.8.1 Registerkarte: GUI-Konfiguration

Mit Hilfe der Registerkarte *GUI-Konfiguration* ist es möglich, Masken für Benutzer zu erzeugen und anzupassen. Neue Masken benötigen eine *Id* und optional einen *Namen*. Weiters kann noch eine *Beschreibung* hinzugefügt werden. Das Auswählen einer *Applikation* ist ebenso obligatorisch wie das Setzen einer *URL* auf eine Startseite oder dem Erzeugen eines XML-Baumes, der in einer XML-Datei gespeichert wird (siehe Abbildung 6.25). Die *Id* ist dabei gleichzeitig der Dateiname der XML-Datei, die im *classes*-Verzeichnis gespeichert wird.

Über den Toolbar können Sie nun folgende Aktionen zum Erstellen eines XML-Baumes durchführen:

- Neu: Hinzufügen von neuen Knoten wie z.B. Arbeitskorb, DMS, Funktionen, etc. (siehe Abschnitt 6.8.1)
- Bearbeiten: Bearbeiten von bestehenden Knoten; gleiche Funktionalität durch Doppelklick auf das Element
- Löschen: Löschen von bestehenden Knoten, Subknoten werden dadurch mitgelöscht
- Nach oben: Verschiebt den ausgewählten Knoten um eine Stelle nach oben in der gleichen Ebene
- Nach unten: Verschiebt den ausgewählten Knoten um eine Stelle nach unten in der gleichen Ebene
- Nach rechts: Verschiebt den ausgewählten Knoten in eine Ebene tiefer (optisch dargestellt nach rechts)

- Nach links: Verschiebt den ausgewählten Knoten in eine Ebene höher (optisch dargestellt nach links)
- Eigenschaften: In diesem Fenster können diverse Einstellungen zur Darstellung der Maske getätigt werden (siehe Abschnitt 6.8.1)
- Vorschau: Mit Hilfe dieser Funktion wird eine Vorschau der aktuell zusammengestellten Maske angezeigt wie sie auch die Benutzer sehen würden

실 GUI-Konfigura	tionen: myconfig3 - @enterprise - Mozilla Firefox 📃 🗖 🔯
http://localhost	:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.guiconf&foreignKey=application&application= 🏠 🚳
GUI-Konfiguratio	on Zuordnungen
ld:	myconfig3 Name: myconfig3
Beschreibung:	
Applikation:	Default
Datei:	file:/C:/Enterprise8D/ep_oracle/classes/myconfig3.xml
O Url:	
⊙ Tree:	
	MyConfig3
	MyWorklist
	P MyWorklist
	MyUserFolder
	MyStartProcess
	MyFunctions
	₽ ²² MyDMS
	L DMS
	A misc
	- MyLink
	MyReport
	Mutable2
	Mujable2
Löschen	Ok Abbrechen Ühernehmen

Abbildung 6.25: Registerkarte: GUI-Konfiguration

Knoteneigenschaften

Durch Auswählen der Funktion *Neu* bzw. *Bearbeiten* öffnet sich das HTML-Fenster *Knote-neigenschaften*, in dem neue Knoten hinzugefügt bzw. bestehende bearbeitet werden können (siehe Abbildung 6.26). Knoten bestehen immer aus einer *Beschriftung*, die übersetzt angezeigt werden kann oder auch nicht. Durch Aktivieren der Checkbox *Lokalisieren* wird der eingetragene Name übersetzt (ohne @), sofern der String im Default- bzw. in einem angegebene Resourcebundle vorhanden ist. Der I18n-Link neben diesem Feld ermöglicht direkt

die Anpassung der Ressource (falls in Applikationsdetailmaske - Registerkarte *Eigenschaften* festgelegt). Die Änderungen werden nach Aktivieren der Schaltfläche *Speichern* in der Ressource Datei (Strings) dieser Applikation gespeichert (siehe Abschnitt 6.9). Die Checkbox *XML* dient dazu, dass HTML-Tags im Namen nicht codiert, sondern sinnhaft dargestellt werden, d.h. MyNode führt dazu, dass in der Ansicht der Knoten in Fettschrift dargestellt wird.

Ein weiteres fixes Element ist die Checkbox *Klickbar*. Ist diese aktiviert, wird der bestehende Knoten und dessen Subknoten als Baum dargestellt, ansonsten werden alle Knoten angezeigt. Mit Hilfe der Selectlist *Zugriff* können Rollen festgelegt werden. Benutzer, die diese Rollen besitzen, haben damit auch Zugriff auf diesen Knoten, d.h. der Knoten ist auf der Maske sicht- und/oder auswählbar.

Weitere fixe Elemente eines Knotens sind noch

- Id: Eindeutige Identifizierung des Knoten. Wenn keine Id angegeben wurde, vergibt sie @enterprise automatisch.
- CSS-Klasse: Möglichkeit der Angabe einer CSS-Klasse.
- Default: Wenn diese Option aktiviert ist, dann ist der jeweilige Knoten die Startseite, die nach dem Login im Hauptfenster geladen wird (sollte nur 1x vergeben werden!).

Folgende Knotentypen stehen zur Auswahl:

- Text: Frei wählbarer Text, der im Navigationsbaum angezeigt wird; kann z.B. für Gruppenteilung verwendet werden.
- Link: Definition eines Links im Navigationsbaum. Dabei können folgende Attribute festgelegt werden:
 - Zielfenster: Der Inhalt dieses Feldes bestimmt, wo diese Ausgabe angezeigt wird (z.B. _top). Wird in dieses Feld nichts eingetragen, wird die Ausgabe im rechten Frame angezeigt.
 - URI: Dieses Feld bietet die Möglichkeit händisch eine URL einzutragen oder über das Such-Symbol neben dem Feld einen @enterprise-Link auszuwählen.
- Prozess starten: Dieser Knoten dient dazu, um eine Liste der zu startenden @enterprise-Prozesse zu erhalten. Dabei kann zum Beispiel festgelegt werden, wer diese Liste sehen darf und wo sie angezeigt wird (analog zu *Link*). Weiters kann eine *Arbeitskorb-Id* festgelegt werden (z.B. standard.wl), die der *wlid* im XML entspricht. Wenn eine solche Id festgelegt wurde, wird nach dem Starten eines Prozesses der Arbeitskorb mit der zutreffenden Id angezeigt.
- Funktionsliste: Mit Hilfe dieses Knoten können alle globalen Funktionen einer oder mehrerer Applikationen angezeigt werden.
- Funktion: Dieser Knoten erlaubt das Hinzufügen einzelner Funktionen zum Navigationsbaum. Da diese Funktion im Navigationsbaum den Kontext des Hauptfensters nicht kennt, sollten hier globale Funktionen (auf keinen Eintrag anwendbar) bevorzugt werden.

- Report: Durch Hinzufügen dieses Knoten kann der ausgewählte Report im Navigationsbaum ausgeführt werden (analog zu 8.4)
- Arbeitskorb: Hier kann der Arbeitskorb festgelegt werden. Dabei können folgende Attribute gesetzt werden:
 - Worklist-Adapter: Tragen Sie hier die Klasse Ihres Worklist-Adapters ein. Nähere Informationen zu Worklist-Adaptern finden Sie im *Programmierhandbuch* Kapitel *Customizing the Worklist*.
 - Typ: Festlegung des Arbeitskorbtyps (z.B. Arbeitskorb, Rollen-Arbeitskorb, Wiedervorlage, Rollen-Wiedervorlage, etc.). Dabei können mehrere Typen gleichzeitig für einen Knoten ausgewählt werden.
 - Spalten: Setzen der Spalten, die in der Tabelle angezeigt werden sollen. Dabei können Spalten hinzugefügt, bearbeitet, gelöscht und in ihrer Reihenfolge verändert werden. Spalten bestehen aus einer *Id* und einem *Namen*, der internationalisiert werden kann (Checkbox *Lokalisieren*), oder einem *Icon*. Wenn die Checkbox *Sichtbar?* aktiviert ist, wird die Spalte beim erstem Aufruf des Arbeitskorbs angezeigt, andernfalls ist sie über den Columnpicker hinzufügbar. Bei der Id kann weiters eine Option angegeben werden, welcher Reiter beim Aktivieren des Links im Arbeitskorb geöffnet werden soll, z.B. *process-form0* es wird die Spalte Prozess angezeigt, jedoch mit dem Link auf das 1. Formular in der Reiteransicht anstatt dem Standardlink auf die Prozesshistorie.
 - Funktionen: Hier können entweder vorhandene *Task-Funktionen* hinzugefügt oder *Action Keys* gesetzt werden (z.B. new, cut, copy, etc). Weiters gibt es die Möglichkeit Funktionen aus der Liste zu löschen und deren Reihenfolge zu verändern.
 - Ohne Dokumente/Notizen: Durch Aktivieren dieser Checkbox wird verhindert, dass Dokumente und Notizen der Prozesse beim Anzeigen des Arbeitskorbs bereits gelesen werden. Aktivieren Sie diese Checkbox nur, wenn keine Dokumente oder Notizen bei Prozesse verwendet werden und die Anzeige der Arbeitskorbtabelle optimiert werden soll!
 - Keine Benutzerordner Filterung: Durch Aktivieren dieser Checkbox wird verhindert, dass Einträge in Benutzerordner beim Anzeigen des Arbeitskorbs herausgefiltert werden. Aktivieren Sie diese Checkbox nur, wenn keine Benutzerordner verwendet werden und die Anzeige der Arbeitskorbtabelle optimiert werden soll!
- Benutzerordner: Definition eines Benutzerordners im Navigationsbaum, der als Platzhalter dient. Felder sind analog zu Knotentyp *Arbeitskorb*, jedoch ohne die Möglichkeit einen *Typ* festzulegen. Dieser Knoten kann nur 1x im Navigationsbaum vergeben werden!
- DMS: Dieser Knoten ermöglicht die Erstellung bzw. Anpassung eines DMS-Ordners. Dabei können folgende Attribute angepasst werden:
 - Spalten: Analog zu Knotentyp Arbeitskorb
 - Funktionen: Analog zu Knotentyp Arbeitskorb

- Formulare: In dieser Liste können Sie festlegen, welche Formulartypen für diesen Ordner erlaubt bzw. nicht erlaubt sind. Wenn z.B. die Liste leer und der Radio-Button Nicht erlaubt aktiviert ist, sind alle verfügbaren Formulartypen für den Benutzer in der DMS–Tabelle auswählbar. Die Auswahl des gewünschten Radio-Buttons wird immer auf alle Formulare in der Selectlist angewendet.
- Paging: Wenn diese Checkbox aktiviert wurde, wird der Pagingmechanismus von @enterprise auf DMS-Tabellen angewendet (analog zu Arbeitskorb-Paging).
- Tabelleneinträge pro Seite: Hier können Sie festlegen, wieviele Einträge pro Seite bei aktiviertem Paging angezeigt werden sollen. Der Parameter wirkt sich nur auf Tabelleneinträge dieses Ordnertyps aus!

Dieser Knoten kann, wie auch der Knoten *Benutzerordner*, nur 1x im Navigationsbaum vergeben werden!

- Tabelle: Mit Hilfe dieses Knotens kann ein Link auf eine Tabelle erzeugt werden. Diese Tabelle kann Formularklassen (auswählbar über Icon) anzeigen. Dabei können folgende Attribute angepasst werden:
 - Klassenname: Hier kann eine Formularklasse ausgewählt werden.
 - Spalten: Analog zu Knotentyp Arbeitskorb
 - Funktionen: Analog zu Knotentyp Arbeitskorb
 - Detailfenster-Eigenschaften: Setzen der Fenster-Eigenschaften wie z.B. width, height, scrollbars etc. getrennt durch Semikolons wie bei der Java-Script-Methode *window.open*.

Hinweis: Nur Formularklassen können mit dem Wizard korrekt erstellt und anschließend verwendet werden!

Eigenschaften

Mit Hilfe der GUI-Eigenschaften können spezifische Einstellungen für die Maske getroffen werden. Folgende Attribute stehen zu Verfügung:

- HTML-Maske: Angabe einer HTML-Maske, die der Anzeige dient (entspricht Attribut *framepage* in der XML-Datei *standard.xml*)
- HTML-Maske (rechts-nach-links): Analog zu *HTML-Maske* (entspricht Attribut *fra-mepageRTL* in der XML-Datei *standard.xml*)
- Schaltflächen für erste Baumebene: Alle Baum-Elemente in 1. Ebene werden als Schaltflächen dargestellt (Default @enterprise-Verhalten). Diese Funktion sollte nicht angewendet werden, wenn auf der 1. Ebene externe Links (z.B. Verweis auf andere Homepage) angelegt wurden - @enterprise-Links können jedoch verwendet werden!
- Ordnersymbol anzeigen: Bei jedem Element erscheint ein Ordnersymbol, abhängig davon ob die Checkbox *Klickbar* aktiviert wurde.

6.8. GUI-KONFIGURATION



Abbildung 6.26: Knoteneigenschaften

- Klickbar: Darstellung als Baumstruktur (auf- und zuklappbar); kann nicht parallel mit *Schaltflächen für erste Baumebene* betrieben werden!
- Aktualisierungsintervall (sek.): Zeitintervall in Sekunden, wann die Seite automatisch aktualisiert wird
- Methode für Baumaufbau: Hier kann eine Methode angegeben werden, die den Baumaufbau beeinflussen kann. Diese Methode muss einen Rückgabewert *Page* besitzen und hat als Parameter ein *com.groiss.ds.Pair*,

z.B. public Page createTree(Pair pair). Aufgerufen wird diese Methode ohne Angabe des Parameters, z.B. <package>.<class>.createTree. Ein kurzes Beispiel für solch eine Methode ist im *Programmierhandbuch* im Kapitel *Configuring the Worklist Client* zu finden.

6.8.2 Registerkarte: Zuordnungen

Wenn verschiedene Client–Konfigurationen verwendet werden, können Sie hier festlegen, welcher Benutzer bzw. welche Rolle die aktuelle Konfiguration verwenden soll. Werden bei der Anmeldung eines Benutzers für diesen mehrere passende Einträge gefunden, wird jener mit der höchsten Präferenz ausgewählt.

🥮 Zuordnungen: i	tsm - @enterprise - Mozilla Firefox 📃 🗖 🔀
http://localhost:8	3380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.defaulturl&foreigr 🏠 🛞
Allgemein	
Akteur	
Rolle:	Home (Default)
OE:	Service 🔹
🔘 Benutzer:	▼
Präferenz: 0	
Löschen	Ok Abbrechen Übernehmen

Abbildung 6.27: Objekt-Detailansicht: Zuordnungen

Folgend eine Beschreibung der Detailmaske:

- Akteur: Der Akteur, für den ein TREE bzw. eine URL festgelegt werden soll. Sie können als Akteur entweder einen *Benutzer* oder eine *Rolle* (mit *Organisationseinheit*) festlegen.
- **Präferenz:** Es können für einen Akteur mehrere TREEs bzw. URLs angegeben werden. Beim Anmelden wird jene Seite mit der höchsten Präferenz berücksichtigt.

6.9 Ressourcen Editor

Dieser Abschnitt beschreibt die Verwendung des @enterprise Ressourcen Editors. Dieses Tool ermöglicht neben der Adaption von Applikationsressourcen (Strings) die Ansicht

im schreibgeschützten Modus von @enterprise Standardressourcen. Der Ressourcen Editor kann nur verwendet werden, wenn in der Detailmaske der Applikation eine Ressource eingetragen wurde (siehe Abschnitt 6.1 - *Registerkarte: Eigenschaften*). Die Applikation *Default* bildet dabei eine Ausnahme, da hier die Standardressourcen von @enterprise immer im schreibgeschützten Modus angezeigt werden. Eine neue Sprache kann jedoch immer mittels der Toolbarfunktion *Neue Spalte* hinzugefügt werden (siehe Abschnitt 6.9.1).

Hinweis: Der Ressourcen Editor erzeugt beim Speichern jeweils eine CSV-Datei (Strings.xls) und Property-Dateien (abhängig vom eingetragenen Pfad in der Applikationsdetailmaske). Um im Editor bestehende Ressourcen editieren zu können müssen eine CSV- bzw. die Property-Dateien am Dateisystem verfügbar sein. Weitere Informationen über Ressourcen finden Sie im *Programmierhandbuch* von @enterprise.

Durch Aktivieren des Verweises *Ressourcen* im Navigationsbaum wird eine Tabellenansicht mit den Applikationsressourcen (Strings) geladen. Beim Aktivieren dieses Verweises in der Applikation *Default*, wird eine Auswahlseite für Strings und Errors angezeigt (Standard *@enterprise* Ressourcen). Die Toolbarfunktionen der Tabellenansicht werden im Abschnitt 6.9.1 beschrieben. Folgende Spalten sind verfügbar (siehe Abbildung 6.28):

- Znr: Entspricht der Zeilennummer
- Schlüssel: Diese Spalte beinhaltet alle Schlüssel, die übersetzt werden sollen. Existierende Schlüssel können nicht in dieser Ansicht geändert werden (nur gelöscht).
- **Sprachspalten:** Eine Menge von Spalten wird angezeigt, wobei eine Spalte eine Sprache repräsentiert. Die 1. Sprachspalte entspricht automatisch der Standardsprache (= *Strings.properties*).

Das Tabellenverhalten (Sortierung, Spaltenauswahl, etc.) ist analog zum Standard @enterprise Verhalten (siehe Abschnitt 2.1.1).

6.9.1 Toolbarfunktionen

Þ

In diesem Abschnitt werden die diversen Toolbarfunktionen zur Adaptierung der Ressourcen behandelt. Die meisten Funktionen können nicht ausgeführt werden, wenn die Ressourcen aus einer JAR-Datei geladen werden. Folgende Funktionen sind verfügbar:

- Neue Zeile: Diese Funktion fügt der Tabelle eine neue Zeile hinzu. Wenn keine Zeile aktiviert wurde, wird die erste leere Zeile auf der letzten Seite aktiviert oder, wenn keine leere Zeile vorhanden ist, eine neue Seite mit einer leeren Zeile angelegt. Wenn eine Zeile ausgewählt ist, so wird die neue Zeile an der entsprechenden Position, abhängig von der Sortierung, eingefügt.
- Zeile bearbeiten: Wählen Sie eine Zeile aus und aktivieren Sie diese Funktion um eine Übersicht über den ausgewählten Übersetzungsschlüssel zu erhalten. Diese Ansicht erlaubt weiters die Bearbeitung der Übersetzungen und das Vor- bzw. Zurückspringen in den Zeilen. Änderungen werden erst nach Betätigen der Schaltfläche

6.9. RESSOURCEN EDITOR

🥹 @enterprise-Administration	n - Mozilla F	irefox		
<u>D</u> atei <u>B</u> earbeiten <u>A</u> nsicht <u>C</u> hro	inik <u>L</u> esezeic	hen E <u>x</u> tras <u>H</u> ilfe		
🐌 @enterprise-Administration		+		
A A Real Marce	lle et e 2200 / u fu	loou date match address along some lateral 1.1784	ll Cui shawAdain 🔗 - 🔗 🖉 💴 -	Canada 🕥 🖉
	milliost:0300/wii/	serviet.methodycom.dec.avw.num.Hhm		aoogie 🔪 🔪
ontor	Admin	histration	ep80 (c	racle) - angemeldet: Markus Irrasc
(prise	-			
	+ 2	X 🖞 🖬 < I		
Organisation	ITC	I. Carlo an		14 4 4 3 3 4 5 4 4
Applikationen		A:Strings		< < 1 2 3 4 5 > >
Formulare	▲ Znr	Schlüssel	Englisch	Deutsch
Tasks	1	abortandarchive	Abort and Archive	Abbrechen und Archivieren
- Funktionen	2	addtoexistprob	Add to an existing problem	Zu einem bestehenden Problem hinzufügen
Rollen	3	addtonewprob	Add to a new problem	Zu einem neuen Problem hinzufügen
Rechte	4	archive_btn	Archive	Archivieren
	5	archivefollowingprocs	Archive the following processes	Archiviere folgende Prozesse
Funktionsgruppen	6	ask_selection_table	Attach email to:	E-Mail anhängen an:
	7	assign	Assign	Zuweisen
Ressourcen	8	assigner	Assigner	Zuweiser
Webservice Clients	9	auto.receipt	Auto receipt	Automatische Empfangsbestätigung
Webservice Server	10	auto.state.change	Auto change of state (PM)	Automatische Statusänderung (PM)
Default	11	bcc.recipient	BCC recipient	BCC Empfänger
demos	12	bug	Bug	
ITSM	13	cancel	Cancel	Abbrechen
Brozocco	14	category	Category	Kategorie
Formulara	15	change_req	Change Request	
Tolliale	16	changelogtext	changelog text	Changelog Text
Tasks	17	changespecific	Change specific	Change spezifischer
Funktionen	18	chgreqcomp	Change Request implemented	Change Request implementiert
Rollen	19	cltextinrelform	The following changelog text was saved in re	Der folgende Change-Log-Text wird im Release-I
Rechte	20	comment	comment	Kommentar
Objektklassen	21	commentto	Comment to	Kommentar an
Funktionsgruppen	22	component	Component	Komponente
GUI-Konfigurationen	23	components	Components	Komponenten
Ressourcen	✓ 24	compose	Compose	Erstellen
	25	comprev	Complete in revision	Beendet in Revision
Suche	26	confreicreation	Confirm Release Creation	Bestätigung der Releasegenerierung
Admin-Tasks	Anzahi d	ler Eintrage: 200 0 ausgewählt		
Konfiguration				

Abbildung 6.28: Ressourcen Editor Tabelle

Übernehmen in der Tabelle angezeigt (= temporär gespeichert); persistent gemacht werden die Änderungen durch Aktivieren der Toolbarfunktion *Speichern*.

- Zeile löschen: Diese Funktion löscht die ausgewählte Zeile aus der Tabelle.
- **Kopieren:** Die ausgewählte Zeile wird durch Aktivieren dieser Funktion kopiert. Die kopierte Zeile wird an der entsprechenden Stelle, abhängig von der Sortierung, eingefügt.
- Speichern: Durch Aktivieren dieser Funktion werden alle getätigten Änderungen in der Tabelle in die entsprechenden CSV- bzw. Property-Dateien gespeichert. Die CSV-Datei und die Property-Dateien werden im gleichen Verzeichnis gespeichert, das im Eigenschaften-Dialog der jeweiligen Applikation angegeben wurde (siehe Abschnitt 6.1). Ressourcen, die aus einer JAR-Datei gelesen werden, werden beim Speichern nicht berücksichtigt (diese Sprachen können bereits in der Tabelle nicht bearbeitet werden). Neue Sprachen (= neue Spalte) werden unter dem gleichen Pfad im classes-Verzeichnis der jeweiligen Applikation gespeichert (CSV-Datei und die jeweilige Property-Dateien), wie sie auch in der JAR-Datei vorhanden wären.

Hinweis: Die CSV-Datei wird als UTF-16LE-Datei gespeichert.

1
- Änderungen verwerfen: Aktivieren Sie diese Funktion, um alle getätigten Änderungen im Ressourcen Editor seit dem letzten Speichern (Toolbarfunktion *Speichern*) zu verwerfen.
- **Kurzsuche:** Geben Sie einen Wert in das Textfeld ein und aktivieren Sie diese Funktion, um ein eingeschränktes Ergebnis zu erhalten. Diese Suche verhält sich analog zu Standard Kurzsuche von @enterprise.
- Alle Einträge: Aktivieren Sie diese Funktion, um wieder alle Einträge der Tabelle anzuzeigen.
- **Tabelle für Ressourcedateien sortieren:** Aktivieren Sie diese Funktion, wenn die temporäre Sortierreihenfolge der Tabelle für die CSV- bzw. Property-Dateien verwendet werden soll. Die Änderungen werden erst durch Betätigen der Toolbarfunktion *Speichern* persistent gemacht.
- Neue Spalte: Ein neuer Dialog öffnet sich, wo eine neue Spalte hinzugefügt werden kann, indem aus der Dropdown-Liste ein Wert ausgewählt und anschließend die Schaltfläche *Erstellen* aktiviert wird. Es werden die Standardsprachen in dieser Liste angezeigt falls keine anderen Sprachen unter *Konfiguration* → *Lokalisierung* → *Liste der Sprachen* hinzugefügt wurden. Diese Funktion kann immer ausgeführt werden, auch wenn bestehende Ressourcen aus einer JAR-Datei gelesen werden.

6.9.2 CSV-Datei Konvertierung

Bei Verwendung von CSV-Dateien muss sichergestellt sein, dass sie UTF-16LE codiert sind. Folgende Funktion ermöglicht die Konvertierung von Cp1252 nach UTF-16LE:

http://'host':'port'/'context-root'/servlet.method/ com.groiss.reseditor.ResourceEditorService.convertXLS?resource=<reurl>

Der Parameter *resource* muss dabei der URL der CSV-Datei entsprechen, z.B. für Applikation *myappl* Strings:

http://'host':'port'/'context-root'/servlet.method/ com.groiss.reseditor.ResourceEditorService. convertXLS?resource=com/dec/myappl/resource/Strings.xls

6.10 Web-Dienste

a·z

Dieser Abschnitt behandelt die Erstellung von Webservice Server/Client Objekten, die beim Prozesseditor in einem der 3 Web-Dienste-Knoten definiert werden können (siehe Kapitel 7.2.15). Um überhaupt Webservice Server/Client Objekte erzeugen zu können, müssen entsprechende WSDL-Dateien im Klassenpfad von @enterprise vorhanden sein.

Zu beachten ist, dass die WSDL-Dateien dem WS-I Basic Profile 1.1 entsprechen müssen (http://www.ws-i.org/profiles/basicprofile-1.1.html).

Wenn Sie Web-Dienste anbieten wollen, welche nicht dem WS-I Basic Profile entsprechen (z.B. RPC Webservices,...), können Sie diese Dienste über die Funktion $Admin-Tasks \rightarrow Kommunikation \rightarrow Web Dienste \rightarrow Lokale Dienste hinzufügen/aktivieren (siehe Kapitel 9.5.5). Auf diese Art bereitgestellte Web-Dienste können jedoch nicht (automatisch) in Prozessen verwendet werden.$

Webservice Server/Client Objekte können pro Applikation definiert werden.

6.10.1 Webservice Clients

Mit Hilfe der Client Objekte kann definiert werden, welcher Web-Dienst mit welchen Parametern aufgerufen wird (OUT-Parameter) bzw. welche Parameter vom Web-Dienst erwartet werden (IN-Parameter). Client Objekte werden dazu benötigt, wenn man Daten einem anderen Server für eine weitere Verarbeitung bereitstellt.

Die Objekt-Detailansicht gliedert sich in 3 Registerkarten:

- Allgemein
- Operationen
- Historie

Registerkarte: Allgemein

🕙 Webservice Clien	ts: kclient (client?: true wsdl: wsdl/kelag.wsdl) - @enterprise - Mozilla Firefox 🛛 🖃 🔲 🔀
http://localhost:838	30/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.webservice_client&foreir 🏫 🛞
Allgemein Oper	ationen Historie
ld:	kclient
WSDL Datei:	wsdl/kelag.wsdl
Webservice:	sendMessageService 🔹
Port:	sendMessageSOAP 🔹
URL:	http://localhost:8380/wf/services.axis2/kserver
Benötigte Module:	
Applikation:	Kel-Applikation 👻
Löschen	Ok Abbrechen Übernehmen

Abbildung 6.29: Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

• Id: Die ID des Webservice Client Objekts, die frei vergeben werden kann. Die Id muss pro Applikation eindeutig sein.

- WSDL-Datei: Der Pfad zu einer im Klassenpfad befindlichen WSDL-Datei.
- Webservice: Hier wird eine Auswahl der Webservices angeboten, die in der WSDL-Datei definiert wurden. Eine Auswahl ist nur möglich, wenn zuvor der korrekte Pfad auf eine WSDL-Datei angegeben wurde.
- **Port:** Abhängig von dem ausgewählten *Webservice* kann hier der entsprechende Port ausgewählt werden, der in der WSDL-Datei definiert wurde.
- URL: Die URL des Web-Dienstes, der aufgerufen werden soll. Wenn keine URL angegeben wurde, wird jene aus der WSDL-Datei verwendet.
- Benötigte Module: Hier kann eine durch Beistrich getrennte Liste von AXIS2-Modulen angegeben werden, die für diesen Web-Dienst benötigt werden, z.B. rahas,rampart,scripting
- Applikation: Die Applikation, in der das Client Objekt gespeichert werden soll. Standardmäßig ist jene Applikation vorausgewählt, von der aus die Toolbarfunktion *Neu* aufgerufen wurde.

Nachdem die Informationen in der Registerkarte *Allgemein* gespeichert wurden, muss ein *Operation*-Objekt in der Registerkarte *Operationen* erzeugt werden. In diesem *Operation*-Objekt können dann die Ein- bzw. Ausgangsparameter festgelegt werden (IN-/OUT-Parameter).

Registerkarte: Operationen

In dieser Registerkarte wird eine Tabelle aller Operationen zu diesem Client Objekt angezeigt. Neben den Standard-Toolbarfunktionen gibt es noch eine Funktion *Webservice-Operation aufrufen*, mit der man die Web-Dienst Operation testweise mit den entsprechenden Ausgangsparameter aufrufen kann.

Um eine Operation zu erstellen, aktivieren Sie die Toolbarfunktion *Neu*. Anschließend wird ein neuer Dialog geöffnet, wo man eine *Operation* (definiert in der WSDL-Datei) auswählen kann (siehe Abbildung 6.30). Anschließend kann mittels der Funktion *XML generieren* ein XML generiert werden, das im gleichnamigen Feld gespeichert wird.

Nachdem ein *Operation*-Objekt angelegt wurde, können die Ein- bzw. Ausgangsparameter in den entsprechenden Registerkarten angegeben werden. Beim Client Objekt sind Ausgangsparameter (= OUT-Parameter) jene, die dem Web-Dienst zur Verarbeitung (beim anderen Server) übermittelt werden. Eingangsparameter (= IN-Parameter) sind jene, die vom Web-Dienst erwartet werden (z.B. Statusmeldung über Verarbeitung). Ein Parameter wird definiert durch eine *Id*, einen *Namen* und einen *Pfad (XPath)*, wobei alle 3 Felder Mussfelder sind. Als Namespace-Prefix im XPath-Ausdruck können alle Prefixes verwendet werden, welche im Root-Element der WSDL-Datei definiert sind. Parameter können entweder händisch mittels der Toolbarfunktion *Neu* erzeugt werden oder sie werden automatisch mittels der Funktion *Parameter generieren* erzeugt. Es ist dabei nicht möglich Parameter mit gleicher ID mehrfach anzulegen.

6.10. WEB-DIENSTE

🥹 Operationen: kclient.sendMessage - @enterprise - Mozilla Firefox 📃 🗖 🔀							
📸 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.ws_ 🏠 🎯							
Allgemein Out-Pa	rameter In-Parameter	Historie	Referenziert von				
Operation: Nachrichten-Handler XML	sendMessage /sendMessage/"> <ioptional:> <messagetemplate :<br=""><ioptional:> <messagetype></messagetype> <ioptional:></ioptional:></ioptional:></messagetemplate></ioptional:>	>					
	<pre><enterpriseld></enterpriseld> <!--Optional:--> <xeoxid></xeoxid> <!--Optional:--> <reporter></reporter> <!--Optional:--> <subject></subject> <!--Optional:--> <text></text> <text></text> <!--Optional:--></pre>						
Löschen	Ok		Abbrechen	Übernehmen			

Abbildung 6.30: Das Operation-Objekt

6.10.2 Webservice Server

Mit Hilfe der Server Objekte können Webservice am Server zur Verfügung gestellt werden. Andere Systeme können diese aufrufen.

Die Objekt-Detailansicht gliedert sich analog zu *Webservice Clients* auch hier in 3 Registerkarten:

- Allgemein: Beinhaltet die selben Felder wie *Webservice Clients*, nur mit dem Unterschied, dass keine URL angegeben werden kann.
- Operationen: Analog zu *Webservice Clients*, jedoch mit dem Unterschied, dass die Funktion nicht verfügbar ist. Als Eingangsparameter (= IN-Parameter) werden in diesem Fall jene bezeichnet, die vom Web-Dienst zur Verarbeitung empfangen werden und Ausgangsparameter (= OUT-Parameter) als jene, die dem Web-Dienst übermittelt werden (z.B. Statusmeldung über Verarbeitung). Optional kann noch ein *Nachrichten-Handler* angegeben werden, der das Interface *com.groiss.ws.server.MessageHandler* implementieren muss. Wenn ein *Nachrichten-Handler* eingetragen wurde, kann diese Operation nicht mehr in einem Prozess verwendet werden.
- Historie: Analog zu Webservice Clients.

7 Prozessdefinition

In diesem Kapitel wird das Vorgehen bei der Definition von Prozessen beschrieben. In @enterprise gibt es zwei Möglichkeiten der Prozessdefinition:

- 1. grafische Definition mittels Prozesseditor,
- 2. Definition des Prozesses als Script in der Workflow Definition Language (WDL)

Beide Formen haben die gleiche Mächtigkeit und sind austauschbar, d.h. sie können einen grafisch definierten Prozess als WDL–Script speichern, dieses weiter bearbeiten, das WDL–Script laden, und den Prozess im Prozesseditor weiter verändern.¹.

Im folgenden wird zuerst die Scriptsprache WDL beschrieben, da hier die einzelnen Sprachelemente leichter formalisierbar sind.

7.1 WDL

In diesem Abschnitt wird die Syntax der Sprache und die Semantik (Bedeutung) der einzelnen Sprachkonstrukte beschrieben. WDL (Workflow Definition Language) ist die Prozessbeschreibungssprache von @enterprise. Die Sprache ähnelt einer strukturierten Programmiersprache und ermöglicht die Definition von WDL-Prozessen. Ein WDL-Prozess besteht aus Prozesskopf, Deklarationsteil und Anweisungsteil.

Beispiel:

```
process jobproc()
name "jobproc";
description "simple process";
version 1;
subject f.subj;
forms f Jobform;
begin
<label_order> all order(f);
loop
f.recipient a_task(f);
```

¹Das grafische Layout und Annotationen werden bei Notationswechsel nicht beibehalten

exit when (f.finished = 1); end; label_order:user inform(f); end;

Die Prozessdefinition beginnt mit dem Schlüsselwort process, gefolgt von der Prozess-Id und einer Argumentliste (= Prozesskopf). Danach folgt der Deklarationsteil als Menge von Key-Value Paaren, z.B. version 1;.

Der Anweisungsteil beginnt mit dem Schlüsselwort begin und endet mit end. Dazwischen wird die Struktur des Workflows, bestehend aus Taskaufrufen, Subprozessen, automatischen Schritten und den verschiedenen Kontrollkonstrukten beschrieben.

7.1.1 Lexikalische Konventionen

In WDL gelten die folgenden lexikalischen Regeln:

• Ids

Ids sind Bezeichner für Tasks, Rollen, Benutzer u.ä. In WDL gelten dafür folgende Konventionen:

Ids beginnen mit einem Buchstaben oder \$ oder / oder \, danach können weitere der genannten Zeichen oder Ziffern kommen. Die Gesamtlänge von Ids darf 80 Zeichen nicht überschreiten.

• Strings

Strings sind in doppelte Anführungszeichen eingeschlossene Zeichenketten. Doppelte Anführungszeichen innerhalb von Strings müssen verdoppelt werden.

Beispiele:

```
"Das ist ein String."
"Das ist ein String mit zwei ""Anführungszeichen""."
```

• Kommentare

Zwischen " /*" und "*/" werden alle Zeichen ignoriert. Damit lassen sich insbesondere auch mehrzeilige Kommentare realisieren.

• Groß-/ Kleinschreibung

Zwischen Groß- und Kleinschreibung wird im allgemeinen unterschieden, d.h. if ist ungleich If.

Schlüsselwörter

Schlüsselwörter sind in der Tabelle 7.1 aufgelistet. Ein Schlüsselwort, das innerhalb eines einfachem Hochkommas steht wird nicht als Schlüsselwort interpretiert, jedoch als Id.

²Die Kommentare werden beim Laden des WDL-Scripts nicht ins System übernommen.

abort	adhocTasks	and	andpar	application
autofinish	baseform	batch	begin	branch
call	choice	corr	correlation	current_tx
days	description	do	else	elsif
end	exception	exit	for	forms
gobackonerror	goto	hours	if	in
instanceid	invoke	loop	maxtime	minutes
name	new_tx	newthread	none	not
null	or	orpar	out	owner
parallel	priority	process	raiseEvent	receive
registerForEvent	repeat	reply	skipable	start
startfunction	startnow	subject	success	sync
system	then	timeoutaction	timeouttask	unregister
until	version	when	while	

Tabelle 7.1: Schlüsselwörter in WDL

7.1.2 Prozesskopf

Syntax:

```
processdef =
  "process" id "(" [formdecl{ "," formdecl}] ")"
  { pdeclaration ";" }
  "begin" [nodename]
    statseq
  "end" [nodename] .
```

Beschreibung der Syntax:

- Id: interner Name des Prozesses
- Parameterliste: Formulare, die an den Prozess übergeben werden, wenn dieser als Subprozess aufgerufen wird. Die Übergabe erfolgt *by reference*, d.h. die Formulardaten werden nicht kopiert.
- pdeclaration: Deklarationen, siehe unten.
- statseq: Sequenz der Anweisungen (Taskaufrufe etc.).

7.1.3 Deklarationsteil

Im Deklarationsteil finden sich eine Reihe allgemeiner Informationen über den Prozess.

Syntax:

```
pdeclaration =
"name" string
| "description" string
```

```
| "version" number
| "subject" ( formfield | expressionstring )
| "maxtime" number ("days" | "hours" | "minutes" )
| "timeoutaction" ("none" | "abort" )
| "timeouttask" taskstmt
| "forms" formdecl { "," formdecl}
| "application" application
| "instanceid" string
| "priority" number
| "adhocTasks" adhoctask { "," adhoctask }
```

Beschreibung der einzelnen Deklarationen:

- name: Name des Prozesses, wie er in Benutzerdialogen auftaucht.
- description: freier Text
- version: Integer, gibt die Version dieser Prozessdefinition an.
- subject: Inhalt der Betreff-Feldes in der Worklist. Kann ein einzelnes Formularfeld (formid.fieldid) sein oder ein Ausdruck der mehrere Felder umfasst. Die Definition ist im Kapitel 6.5 beschrieben.
- maxtime: maximale Bearbeitungszeit, Angabe in Tagen, Stunden oder Minuten.
- timeoutaction: Reaktion im Timeout-Fall. Zwei Reaktionen sind möglich:
 - none: keine Reaktion
 - abort: Prozessabbruch
- timeouttask: Angabe einer Aktion, die im Timeout-Fall ausgeführt wird.³
- forms: Deklaration der Formulare als Prozesslokale Datencontainer: Die Definition eines lokalen Formulars lautet wie folgt:

formid formtype ["baseform" baseformid] ["formname"]

- formid: ist die Id des lokalen Formulars
- formtype: ist ein dem System bekannter Formulartyp
- formname: ist der Anzeigename des lokalen Formulars (optional)
- baseform: wenn das zu deklarierende Formular ein View-Formular ist, muss das Basisformular angegeben werden.

Beispiel: forms rg rechnung, ls lieferschein, rgkopf rechnungskopf baseform rg;

 $^{^3}$ Die Spezifikation von timeoutaction und timeouttask wird nicht mehre empfohlen. Der aktuelle Weg ist die Verwendung von Eskalationen (siehe Abschnitt 6.2.2 bzw. 6.5.10).

- application: Angabe der Id der Applikation, zu welcher der Prozess gehört.
- instanceid: Id, welche jede Instanz eindeutig identifiziert. Hierbei besteht auch die Möglichkeit einen Nummernschema über einen regulären Ausdruck anzugeben (siehe Kapitel 6.5).
- priority: Frei vergebbarer Wert, der die Priorität des Prozesses angibt.
- adhocTasks: Kann verwendet werden, um zusätzliche Tasks zu deklarieren, die zur Laufzeit des Prozesses programmatisch erzeugt werden können. Syntaktisch sind das Task Statements (siehe unten) ohne deklarierte Akteursliste. Die Akteure werden erst zur Laufzeit angegeben. Jeder adhoctask ist wie folgt definiert:

```
taskid "(" [ formlist ] ")" [ nodename ]
```

Alle Deklarationen bis auf name, version und application sind optional..

7.1.4 Einfache Anweisungen

Der Anweisungsteil ist der zentrale Teil einer Prozessspezifikation und befindet sich zwischen den beiden Schlüsselwörtern begin und end. Er enthält zumindest eine Anweisung. Anweisungen werden durch ein Semikolon abgeschlossen.⁴

Syntax:

```
statseq = { [ "<" labelid ">" ] statement [ nodename ] ";" }.
```

statement = (batchstmt branchstmt choice exitstmt gotostmt ifstmt invokestmt loopstmt par parforstmt raiseEvent receivestmt registerForEvent | repeatstmt

⁴ Die erste Anweisung sollte immer ein Taskaufruf sein, da die Akteurdefinition des ersten Schrittes festlegt, wer den Prozess starten darf. Beginnt ein Prozess z.B. mit einer Schleife (loop oder while), kann dieser Prozess nicht über das Benutzer-Interface gestartet werden. Solche Prozesse können allerdings über API gestartet oder als Subprozesse verwendet werden.

| replystmt | subproccall | sync | systemstmt | taskstmt | unregister | whilestmt)

labelid: Die Id dieses Schrittes innerhalb der Prozessdefinition. Sie muss eindeutig sein und kann als exakte Referenz für diesen Schritt verwendet werden. *nodename:* Ein String, der als Anzeigename für das Statement (und die entsprechenden

Knoten im Prozesseditor) verwendet wird. Dieser Name muss nicht eindeutig sein. Beispiel:

<ordertask> all order(form) "Bestellung anlegen";

Im folgenden werden die verschiedenen Anweisungen näher beschrieben:

Manueller Taskaufruf

Aufruf von manuellen Tasks.

Syntax:

```
taskstmt =
    ( "none" | agentlist ) taskid "(" [ formlist ] ")" [ "skipable" ].
```

```
agentlist = agent { , agent }.
```

Beschreibung der Syntax:

Akteure (agentlist): es stehen folgende Möglichkeiten für die Angabe von Akteuren zur Verfügung:

- Der Akteur wird direkt mit seinem Benutzer-Id angegeben. Dies sollte aber nur in wohl begründeten Fällen erfolgen, da die Prozessdefinition für gewöhnlich keine spezifischen Referenzen auf einzelne Benutzer beinhalten sollte.
- Der Akteur wird als Id einer Rolle (logische Benutzergruppe) angegeben. Jeder WF-Benutzer, der dieser Rolle angehört, kann als potentieller Akteur der Aktivität fungieren.
- Der Akteur wird über eine bestimmte Rolle in einer bestimmten Organisationseinheit (OE) definiert. Die Notation dafür lautet: org_einheit_id!roleid. Beispiel: einkauf!sek.

Die Organisationseinheit wird nur für den aktuellen Schritt geändert. Für den Gesamtprozess bleibt die alte OE bestehen.

- Akteur eines vorherigen Schrittes: labelid":user" Dieser Ausdruck gibt den Benutzer an, der die Aktivität mit der Label-Id (Kennzeichnung) labelid innerhalb der aktuellen Prozessinstanz zuletzt ausgeführt hat. Beispiel: antrag_stellen:user
- Akteurspezifikation aus einem Formularfeld. Im Formularfeld steht entweder direkt eine Benutzer-Id, eine Rollen-Id oder die Angabe einer bestimmten Rolle innerhalb einer bestimmten Organisationseinheit (OE) oder der Bearbeiter eines vorigen Schrittes.
- Ein leerer Akteur: "none". Zur Laufzeit muss der kann der Akteur programmtechnisch oder vom Bearbeiter des vorhergehenden interaktiven Schritts gesetzt werden.
- Setzen durch eine Methode: Angabe des Namens einer Java-Methode, die entweder eine Rollen-Id, eine User-Id, eine Kombination aus Organisationseinheit und Rolle oder die Angabe eines Akteurs eines vorherigen Schrittes liefert.
- Sequenz von Akteuren: Es kann auch eine durch Beistrich getrennte Liste von Akteuren angegeben werden, wobei die verschiedenen Formen gemischt werden können. Die Definition von mehreren Akteuren bei einem Task bedeutet, dass diese den Task hintereinander in einer Sequenz bearbeiten.⁵

Die in der Prozessdefinition angegebenen Akteure können zur Laufzeit durch eine Preprocessing– Methode überschrieben werden. Dafür muss beim entsprechenden Task ein entsprechender Methodenaufruf angegeben werden.

taskid: Die Id eines in der Applikation definierten Tasks. Wenn Sie eine Id festlegen, die nicht bereits einem anderen Task vergeben wurde, dann kann die Option *Tasks generieren* beim Laden der Prozessdefinition verwendet werden. Der Task wird dann mit der Id in der Prozessdefinition erzeugt.

formlist: Eine durch Beistrich getrennte Liste von formids, die in der Argumentenliste des Prozesses definiert oder als lokales Formular deklariert wurden.⁶

skipable: Der Task wird übersprungen, wenn zur Laufzeit kein Akteur definiert ist. Wird in der Prozessdefinition als Akteur *none* angegeben (also kein Akteur), wird der Task nur ausgeführt, wenn über eine Preprocessing-Methode ein Akteur gesetzt wurde.

Subprozess-Aufruf

Ein Prozess kann als Teil eines anderen bereits ausgeführten Prozesses aufgerufen werden. Somit können wiederverwendbare und modular aufgebaute Prozesses definiert werden.

Syntax:

subproccall = "call" subprocid "(" [formlist] ")".

⁵Eine etwaige Preprocessing-Methode wird vor dem 1. Akteur ausgeführt, die Postcondition wird erst nach dem letzten Akteur überprüft.

⁶Zur Laufzeit werden die Icons für diese Formulare im Arbeitskorb beim jeweiligen Task angezeigt. Der Formularinhalt ist somit sicht- und editierbar für Akteure dieses Tasks. Im Kapitel 6.2 wird die Möglichkeit der feingranularen Einschränkung auf Einzelfeldebene pro Task pro Task beschrieben.

Das Statement **call** instanziert einen Prozess, der bereits im System definiert wurde, durch Angabe von **subprocid** zur Laufzeit des aktuellen Prozesses. Die Ausführung erfolgt synchron. Der aufgerufene Prozess erhält die Kontrolle und wenn dieser beendet wird, fällt die Kontrolle zurück zum aufrufenden Prozess unmittelbar nach dem **call** Statement. Formulare können dem aufgerufenen Prozess als durch Beistrich getrennte Liste (formlist) von Formular-Ids übergeben werden. Die Übergabe erfolgt *by reference*, d.h. die Formulare werden nicht kopiert. Die formids beziehen sich auf im Aufrufer bekannte Formularvariable und müssen zu den im Prozesskopf deklarierten Formularen des aufgerufenen Prozesses passen.

Systemschritte

Ein Systemschritt wird dazu verwendet, um eine Java-Methode ohne interaktives Zutun auszuführen. Der Name der Methode wird nach dem Schlüsselwort system festgelegt. Eine Liste von Stringliteralen innerhalb der Klammern kann der Parametrisierung dienen. Diese Methode wird synchron abgearbeitet auf kurze Abarbeitungszeiten ist zu achten.

Syntax:

```
systemstmt =
    "system" methodname "(" [ string { "," string } ] ")".
```

Achten Sie auf vollständige Angabe des Klassennamens, also inkl. Package-Bezeichnung, z.b.: system com.groiss.demo.Step.exec("p1","p2");

Batch Schritte

Batch Schritte sind ähnlich wie Systemschritte, da sie auch automatisch ohne Interaktion ausgeführt werden. Der wesentliche Unterschied jedoch besteht darin, dass Batch Aktivität asynchron ausgeführt werden und somit eine längere Abarbeitungszeit beanspruchen können. Bei Batch Aktivitäten muss eine Handler-Klasse angegeben werden, damit auf Ereignisse während der asynchronen Abarbeitung reagiert werden kann.

Beschreibung der Syntax:

```
batchstmt =
    "batch" batchAdapterClassName "(" [ paramstring ] ")
    { "startnow" | "newthread" | "autofinish" | "gobackonerror" }.
```

Achten Sie auf vollständige Angabe des Klassennamens, also incl. Package-Bezeichnung, z.b.: batch com.groiss.demo.DemoBatchAdapter("param"). Weitere Informationen zu Batchjobs finden Sie im Programmierhandbuch und in der API-Dokumentation.

7.1.5 Kontrollstrukturen

Kontrollstrukturen sind notwendig, um den Kontrollfluss zu definieren. Mittels Kontrollstrukturen kann bestimmt werden, in welcher Reihenfolge Aktivitäten ausgeführt werden sollen. Die wichtigsten Kontrollelemente sind die Sequenz, die Alternative und die Parallelität.

Sequenz

Eine sequentielle Abarbeitung von Aktivitäten erreicht man, indem die einzelnen Aktivitätsaufrufe nacheinander im WDL-Programm aufgelistet werden. Als Trennzeichen wird das Semikolon ; verwendet.

Beispiel:

Zuerst soll die Aktivität auftrag_erfassen() von der Rolle sek ausgeführt werden. Sobald diese Aktivität beendet ist, soll die Aktivität erhebung() von der Rolle sachbearbeiter abgearbeitet werden. Danach soll ein Techniker in der Organisationseinheit waehlamt die Aktivität durchschalten() ausgeführt werden.

... sek auftrag_erfassen(auftrag); sachbearbeiter erhebung(auftrag); waehlamt!techniker durchschalten(auftrag); ...

Bedingungen

Bedingungen treten in der WDL an folgenden Stellen auf:

- Alternativen: if, choice
- Schleifen: while, repeat, loops exit when
- Postconditions bei Tasks

Es können Vergleiche zwischen Konstanten und Formularfeldern, Java-Methodenaufrufe, und beliebige logische Verknüpfungen von beiden angegeben werden. Weiters gibt es noch die Möglichkeit mittels Groovy und XPath Bedingungen anzugeben. Nähere Infos zur Verwendung von Groovy und XPath finden Sie im *Programmierhandbuch*.

Syntax:

```
cond = expr1 { "or" expr1 } .
expr1 = expr2 { "and" expr2 }.
expr2 = [ "not" ] expr3.
expr3 =
    "(" cond ")"
    | methodcall
    | booleanformfield
    | formfield relop (number | string | formfield | "null").
```

relop = ("=" | "<>" | "<=" | ">=" | "<" | ">").

```
formfield = formid "." fieldid.
```

Beispiele:

- f.recipient = null
- f.auftragswert > 100000
- com.groiss.Check.isAvailable("f.betrag") and f.klasse > 3
- (f.empfaenger <> null or f.wert > 10000) and f.klasse = 4
- groovy: form_f.betreff == "Book"
- xpath:\$form_f/subform[@id='1']/form/status = 'ok'

Java-Methoden haben 0 bis n konstante Strings als Parameter und den Rückgabetyp boolean. Näheres zur Programmierung von Java-Methoden als Bedingungen sind im *Programmier-handbuch* beschrieben.

If

if und elseifermöglichen die bedingte Ausführung von Prozesszweigen.

Syntax:

```
ifstmt =
   "if" cond
   "then" [nodename] statseq
   { "elsif" cond "then" [nodename] statseq }
   ["else" statseq ]
   "end".
```

Beschreibung der Syntax

- cond: eine Bedingung wie oben definiert.
- statseq: eine Sequenz von Statements.

Beispiel:

```
if auftrag.betrag <= 2000 then "kleiner Auftrag"
   sachbearbeiter bestaetigung_schreiben()
elsif auftrag.betrag <= 5000 and auftrag.klasse = 4 then "mittlerer Auftrag"
   direktor genehmigen()
elsif ...
...
else
...
end</pre>
```

Choice: gemischte manuelle und automatische Alternativenwahl

Choices ermöglichen dem Benutzer einen Prozessweg aus einer vorbestimmten, aber zur Laufzeit eingeschränkten Menge von möglichen Wegen zu selektieren.

Syntax:

```
choice =
  "choice" [ nodename ]
   { branchname [ "," cond ] ":" statseq }
   "end".
```

```
Beschreibung der Syntax:
```

Jeder Zweig hat einen Namen (branchname) — hier kann ein beliebiger String eingegeben werden – und eine optionale Bedingung. Zuerst werden die Bedingungen aller Zweige ausgewertet. Die Zweige mit wahren Bedingungen werden dem Benutzer zur endgültigen Auswahl angeboten. Wenn keine Bedingungen angegeben wurden, wird die Auswahl völlig manuell erfolgen.

Beispiel:

```
choice "manuelle Auswahl"
   "sofort bestellen", f.summe < 5000:
    sek bestellen(f);
   "nochmals prüfen":
    sachbearbeiter pruefen(f);
   "ablegen":
    system Archive.insert();
end;</pre>
```

While-Schleife

Syntax:

```
whilestmt =

"while" cond "do" [ nodename ]

statseq

"end"
```

```
•
```

Beschreibung der Syntax:

Die Anweisungen im Schleifenkörper einer While-Schleife (solange-bis Schleife) werden wiederholt, solange der Wert der Bedingung (cond) wahr ist. Da die Überprüfung der Bedingung vor Eintritt in den Schleifenkörper erfolgt, wird dieser Schleifentyp auch als *abweisende* Schleife bezeichnet. Beispiel:

```
while f.genehmigt = 0 do
  sek korrigieren(f);
end;
```

Repeat-Schleife

Syntax:

repeatstmt = "repeat" [nodename] statseq "until" cond.

Beschreibung der Syntax:

Die Statements im Schleifenkörper werden solange wiederholt, bis die Bedingung wahr ist. Schleifenanweisungen werden mindestens 1 mal ausgeführt.

Beispiel:

```
repeat
sachbearbeiter daten_erfassen(auftrag);
call daten_pruefen(auftrag);
until antrag.daten_ok = 1;
```

Loop-Schleife

Syntax:

```
loopstmt =

"loop" [ nodename ]

[ statseq1 ]

"exit" "when" cond;

[ statseq2 ]

"end".
```

Beschreibung der Syntax:

Allgemeine Schleife, bei Erreichen von end wird wieder zu loop gesprungen. Die Statements in statseq1 werden ausgeführt. Die Bedingung von exit when wird ausgewertet und falls das Ergebnis nicht wahr ist, die Anweisungen von statseq2 ausgeführt und die Schleife wird nochmals durchlaufen. Wenn die Auswertung wahr ergibt, wird die Schleife beendet ohne die Anweisungen aus statseq2 nochmals auszuführen.

Parallelität (andpar und orpar)

Innerhalb eines Prozesses können Teile auch parallel ausgeführt werden. Dies ermöglicht erhebliche Zeiteinsparungen, indem Aktivitäten nebenläufig abgearbeitet werden. Andpar und orpar ermöglichen die Angabe einer festgelegten Anzahl von parallelen Zweigen. Syntax:

```
par =
  ( "andpar" | "orpar" ) [ nodename ]
     statseq
     { "|" statseq }
     "end" [ "do" paramethod ].
```

Grundsätzlich gilt für die beiden Varianten:

- andpar: Prozess wird fortgesetzt, wenn alle Zweige beendet sind.
- orpar: Prozess wird fortgesetzt, wenn ein Zweig beendet ist, alle anderen Zweige werden abgebrochen.

Der optionale paramethod wird später noch genauer beschrieben.

Beispiel für andpar:

Für die Behandlung eines komplexen Schadensfalles ist immer die Einholung von 3 Gutachten notwendig. Erst danach kann eine Bewertung erfolgen:

```
...
andpar
gutachter1 gutachten_erstellen(s_form1);
| gutachter2 gutachten_erstellen(s_form2);
| gutachter3 gutachten_erstellen(s_form3);
end
s_abteilung!leiter bewerten(s_form1, s_form2, s_form3)
...
```

Beispiel für orpar:

Bei der Ermittlung eines Fluges werden immer Anfragen an mehrere Flugreservierungssysteme gerichtet. Sobald ein passender Flug von gefunden wurde, werden die anderen Abfragen abgebrochen (bzw. nicht weiter verfolgt).

```
...
sachbearbeiter flugdaten_erfassen(flugform);
orpar
clerk check_route1(flugform);
| clerk check_route2 (flugform);
end
...
```

•••

Eine Verallgemeinerung der Parallelität, wo n von m Zweigen abgeschlossen sein müssen, wird mit einem *andpar* realisiert, wo beim Endknoten (join) eine entsprechenden Methode *paramethod* angegeben wird:

com.groiss.wf.SystemAction.join(n,action)

Beide Parameter sind Strings:

- n: Die Anzahl der fertigzustellenden Zweige, damit die gesamte Parallelität als abgeschlossen betrachtet wird.
- action: Beinhaltet entweder *none* oder *cancel*. Beim Wert *cancel* werden die verbliebenen aktiven Zweige abgebrochen.

Beispiel für andpar mit n von m abgeschlossenen Zweigen: Für die Behandlung eines minderen Schadenfalles ist die Einholung von nur zwei Gutachten notwendig:

... andpar

```
gutachter1 gutachten_erstellen(s_form1);
| gutachter2 gutachten_erstellen(s_form2);
| gutachter3 gutachten_erstellen(s_form3);
end do com.groiss.wf.SystemAction.join("2","cancel");
...
```

Wenn die Fertigstellung der Parallelität nicht durch eine fixe Anzahl von fertigen Zweigen bestimmt ist, sondern zur Laufzeit bestimmt werden kann, ist dies durch eine Java-Methode im Endknoten (join) möglich. Nähere Informationen dazu finden Sie im *Programmierhandbuch*.

Parallel For

Ermöglicht die Aufspaltung in mehrere identische Zweige. Die Anzahl der Zweige ist erst zur Laufzeit und nicht schon zur Prozessdefinitionszeit bekannt. Syntax:

```
parforstmt =
    "parallel" "for" (localformid "in" formid"."subformtableid | iteratorclass)
    "do" [ nodename ]
        statseq
    "end" [ "do" parmethod ].
```

Mit dieser Kontrollstruktur kann entweder ein paralleler Zweig pro Zeile in einer Subformtabelle erzeugt werden, oder die Generierung der Zweige erfolgt über einen Iterator. Beschreibung der Syntax:

- localformid: neue lokale Variable (frei wählbarer Id), die auf das entsprechende einzelne Subformular im parallelen Zweig verweist.
- formid: Id des Hauptformulars.
- subformtableid: Id der Subformulartabelle wie im tablefield Eintrag bei der Definition des Formulartyps der Hauptformulars angegeben.
- iteratorclass: Name einer Klasse die das Interface com.groiss.wf.ParForIterator implementiert.

Im Endknoten kann durch Angabe einer Methode eine spezifische Bedingung für das Ende das Parfor Konstrukts angegeben werden. Fehlt diese Angabe, si wird auf das Ende aller Zweige gewartet.

Bei einer fixen Anzahl von Zweigen kann die Methode

com.groiss.wf.SystemAction.parforJoin(n,action)

verwendet werden. Die Parameter entsprechen dabei jenen, die bei der Methode *com.groiss.wf.SystemAction.join(n,action)* angegeben wurden (siehe Abschnitt *Parallelität* von Kapitel 7.1).

Wenn die Entscheidung über die Fertigstellung des gesamten Parfor Konstrukts völlig dynamisch erfolgen muss, kann eine Java-Methode angegeben werden. Nähere Informationen über Parallelitäten finden Sie im *Programmierhandbuch*.

Abzweigung

Abzweigungen erlauben die Aufsplittung des Ausführungspfads in einen Hauptpfad und in einen Nebenpfad (den Branch). Syntax:

branchstmt = "branch" [nodename] statseq "end".

Beschreibung der Syntax:

Anweisungen im Branch werden parallel zu den Anweisungen im Hauptzweig ausgeführt. Nebenzweige können die Beendigung des Hauptzweiges überdauern. Beispiel:

begin

```
bearbeiter eingabe(f);
manager kontrolle(f);
branch "Merkliste"
    pruefer inform(f);
    ...
end;
mitarbeiter erzeuge(f);
...
end
```

Goto

Goto's erlauben ein Abweichen des Prozessablaufes und Sprünge zu anderen Teilen der Prozessdefinition.

Syntax:

gotostmt = "goto" labelid.

Die Ausführung wird am Statement das durch labelid bezeichnet wird fortgesetzt. Beispiel:

```
<entry> clerk enter(f);
supervisor check(f);
```

```
if (f.quality <> "OK") then /* denotes exceptional case */
  goto entry;
end;
worker build(f);
```

In diesem Beispiel bewirkt die goto Anweisung, dass die Tasks enter und check nochmals wiederholt werden, wenn die Qualität nicht ausreichend ist. Eine repeat until Schleife wäre eigentlich die bessere Formulierung dieses Ablaufs, jedoch möchte der Designer wahrscheinlich mit diesem Konstrukt explizit zum Ausdruck bringen, dass die Wiederholung ein Ausnahmefall ist.

Die Verwendung von gotos sollte sparsam und nur mit Bedacht in wohlbegründeten Ausnahmefällen erfolgen. Exzessive Verwendung beeinträchtigt die Lesbarkeit und Wartbarkeit der Prozessdefinition nachhaltig.

7.1.6 Eventmechanismus

Der Eventmechanismus wird verwendet um innerhalb der Engine Events zu erzeugen und zu behandeln. Prozesse die ihr Interesse für Events registriert haben, empfangen solche Events von der Engine ausgelösten Aufruf der registrierten Eventhandler. Syntax:

Auslösen von Events durch:

```
raiseEvent =
"raiseEvent" "(" eventname "," ["," form] ")".
```

Warten auf Events:

```
sync =
"sync" "(" eventname "," eventhandler ["," form] ")".
```

Registrieren von Handlern für Events:

```
registerForEvent =
"registerForEvent" "(" eventname "," eventhandler ["," form] ")".
```

Die Registrierung von Eventhandlern aufheben:

unregister = "unregister" "(" eventname ")".

Beschreibung der Syntax

- eventname: Der Name des Events.
- form: Entweder ein Formular oder ein Formularfeld, das als Kontext–Objekt dienen soll.
- eventhandler: Eine Java-Klasse, die das Interface *com.groiss.event.EventHandler* implementiert.

Weitere Informationen zu Events finden Sie im Programmierhandbuch.

7.1.7 Web-Dienste

Die WDL bietet die Möglichkeit Web-Dienste in die Prozessdefinition einzubinden. Über invoke-Knoten können Webservices aufgerufen werden. Die Ausführung in einem Zweig kann mittels, receive-Knoten aufgehalten werden, bis ein Webserviceaufruf von außen erfolgt. Knoten vom Typ reply bezeichnen das Senden einer Antwort auf einen zuvor erfolgten Webservice-Aufruf.

Syntax:

Eingangsnachricht (RECEIVE):

```
receivestmt =
"receive" [ "start" "process" ] operationspec "(" [incorrparams] ")" ["end"].
```

Antwort (REPLY):

```
replystmt =
"reply" operationspec "(" [outparams] ")".
```

Ausgangsnachricht (INVOKE):

```
invokestmt =
  "invoke" [address "."] operationspec "(" [inoutparams] ")"
    ["success" statseq ]
    ["exception" statseq]
  ["end"].
```

Allgemeine Anweisungsteile, die von den Web-Dienste-Knoten verwendet werden:

```
operationspec = serviceid "." operationid.
```

```
incorrparams =
incorrparam {"," incorrparam}.
```

incorrparam = inparam | corrparam.

```
corrparam =
("corr" | "correlation") xpath "=" messagecomp.
```

```
inoutparams =
inoutparam {"," inoutparam}.
```

inoutparam = inparam | outparam.

inparam = ["in"] xpath "=" messagecomp.

```
outparams =
{"," outparam}.
```

```
outparam =
["out"] messagecomp "=" xpath.
```

Beschreibung der Syntax:

- serviceid: Die ID des Web-Dienstes.
- operationid: Die Operations-ID einer Operation des Web-Dienstes.
- address: Die URL des Webservise
- messagecomp: Die ID des (IN/OUT)-Parameters, der Nachricht.
- xpath: Ein XPath-Ausdruck, der das Formularelement bezeichnet, das auf die Nachrichtenkomponente abgebildet wird.
- statseq: Sequenz der Anweisungen.

```
invoke mywebservice.SendMessage(
```

```
MessageTemplate="form_ticket/messageTemplate",
MessageType="$form_ticket/messageType", enterpriseid="$pi/id",
xeoxid="$form_ticket/xeoxId", reporter="$form_ticket/reporter",
'subject'="$form_ticket/subject", """0"""=SendMessageResult)
exception
administrator inform(ticket);
```

```
end;
...
```

```
receive kserver.sendMessage(
```

```
corr "$pi/id"=enterpriseId,
```

```
"$form_ticket/messageTemplate"=messageTemplate,
```

```
"$form_ticket/messageType"=messageType,
```

```
"$form ticket/enterpriseId"=enterpriseId,
```

```
"$form ticket/xeoxId"=xeoxId,
```

```
"$form_ticket/reporter"=reporter,
```

```
"$form_ticket/subject"='subject',
```

```
"$form_ticket/text"=text,
```

```
"$form_ticket/analyst"=analyst
```

```
) waitforincomingmessage;
```

```
reply kserver.sendMessage('out'="""0""");
```

•••

Weitere Informationen zu Web-Diensten finden Sie im Programmierhandbuch.

7.2 Grafische Prozessdefinition

Zur grafischen Neudefinition bzw. Änderung von Prozessen steht Ihnen der @enterprise– Prozesseditor in der Prozesskomponente zur Verfügung. Der Prozesseditor unterstützt dabei die Notationen *BPMN* (Business Process Modeling Notation - siehe Abbildung 7.1) und *WDN* (Worklist Definition Notation - siehe Abbildung 7.2).

Je nachdem, ob Sie einen neuen Prozess definieren wollen oder einen bereits bestehenden abändern, gehen Sie wie folgt vor:

- Durch einen Mausklick auf den Link *Erstellen* öffnet sich der @enterprise-Prozesseditor immer standardmäßig in der BPM-Notation (kann jedoch in den Einstellungen des Prozesseditors geändert werden).
- Markieren Sie zunächst jenen Prozess, den Sie bearbeiten wollen und wählen anschließend den Link *Ändern*. Dadurch wird der @enterprise-Prozesseditor gestartet und der aktuell ausgewählte Prozess darin dargestellt. Nun können Sie damit beginnen, die Prozessdefinition zu ändern. Die Darstellung der Notation beim Öffnen hängt davon ab, in welcher Notation der Prozess gezeichnet und abgespeichert wurde.

7.2.1 Aufbau des Prozesseditors

Der Prozesseditor besteht aus den folgenden Komponenten:

- Titelleiste: Diese Leiste enthält den aktuellen Namen des zu bearbeitenden Prozesses.
- Menüleiste: In der Menüleiste befinden sich folgende Menüpunkte:
 - Prozess
 - Bearbeiten
 - Ansicht
 - Hilfe
 - Symbolleiste
- Zeichenfenster: In diesem Fenster werden die Prozesse grafisch modelliert.
- **Funktionsleiste**: Diese Leiste stellt die erforderlichen Funktionen für die Manipulation von Prozessen zur Verfügung.

Hinweis: Da es Probleme mit Popup-Blocker geben kann, empfehlen wir Ihnen diese für die Dauer der Prozessbearbeitung abzuschalten!

7.2.2 Die Funktionen der Menüleiste

Das Menü Prozess

• Neu: Mit dieser Funktion können Sie den gegenwärtigen Inhalt im Zeichenfenster löschen und neu mit der Modellierung eines Prozesses beginnen.



Abbildung 7.1: Prozesseditor in BPM-Notation

- Öffnen: Mit dieser Funktion können Sie einen bestehenden Prozess öffnen und ihn weiter bearbeiten.
- **Speichern:** Mit dieser Funktion speichern Sie Ihre bisher gemachten Änderungen unter dem aktuellen Prozessnamen ab. Das System informiert Sie, ob die Sicherung erfolgreich durchgeführt wurde oder nicht. Wenn Schritte im Prozess nicht ausreichend spezifiziert wurden (z.B. kein Task zu einer Aktivität zugewiesen), wird der Prozess zwar gespeichert, aber auf *inaktiv* gesetzt. Der Prozess muss vor seiner Verwendung wieder händisch aktiviert werden (siehe Kapitel 6.5).

Hinweis: Das Ausführen dieser Funktion ist nur möglich, wenn Sie zuvor über das



Abbildung 7.2: Prozesseditor in WD-Notation

Menü *Prozess* die Funktion *Eigenschaften* ausgeführt und für den Prozess einen eindeutigen Namen vergeben haben.

- **Speichern unter...:** Mit dieser Funktion können Sie die bestehende Prozessspezifikation unter einem neuen Namen abspeichern. Dazu erscheint der in Abbildung 7.3 dargestellte Dialog.
- Seite einrichten: Hier kann die Seite für den späteren Druck eingerichtet werden (Papierauswahl und -ausrichtung).
- Drucken: Prozess wie in Seitenformat definiert drucken (Standarddruckfenster).

🐸 Speicher	rn unter Mozilla Firefox 📃 🗖 🔀
http://lo	ocalhost:8380/wf/servlet.method/com.dec.avw.html.HTMLProcessD 🏫 🥪
Id:	hr_vacation
Name:	proc_vacation
Version:	1
Aktiv:	
	Ok Abbrechen

Abbildung 7.3: Speichern unter

- Eigenschaften: Mit dieser Funktion werden die Prozesseigenschaften geöffnet (siehe Abschnitt 7.2.3).
- **Tasks:** Mit Hilfe der Task–Maske legen Sie die Tasks fest, die Sie beim Ändern eines Akteurs im Arbeitskorb einem Empfänger in der Empfängerliste zuordnen können.
- **Timeout Task:** Die Reaktion des Prozesses auf Timeouts wird in dieser Funktion spezifiziert (siehe Kapitel 7.2.5).
- Schließen: Mit der Funktion *Schließen* steigen Sie aus dem Prozesseditor aus. Sollten Sie noch nicht gespeicherte Änderungen gemacht haben, so erscheint ein Dialogfenster, welches den Abbruch des Beenden-Vorgangs oder das Abspeichern des Prozesses ermöglicht.

Das Menü Bearbeiten

- **Rückgängig:** Mit dieser Funktion können die letzten n Schritte im Zeichenfenster rückgängig gemacht werden.
- Wiederherstellen: Diese Funktion ist analog zu Rückgängig.
- Ausschneiden: Diese Funktion erlaubt es, Elemente aus einer Stelle im Prozess auszuschneiden und an einer anderen Stelle im Prozess wieder einzufügen. Wenn Sie Elemente ausschneiden, können Sie diese nur 1x wieder einfügen. Es wird hierbei das Original verschoben und keine Kopie erzeugt, was zur Folge hat, dass alle Einstellungen (Akteure, Formular-Sichtbarkeiten, etc.) für die ausgeschnittenen Elemente erhalten bleiben.
- **Kopieren:** Diese Funktion erlaubt es, Elemente aus einer Stelle im Prozess zu kopieren und an einer anderen Stelle im Prozess wieder einzufügen. Wenn Sie Elemente kopieren, können Sie diese beliebig oft wieder einfügen. Dabei werden nicht alle Einstellungen (z.B. Formular-Sichtbarkeiten) für die kopierten Elemente übernommen.
- Einfügen: Mit Hilfe dieser Funktion können Sie vorher ausgeschnittene bzw. kopierte Elemente an der ausgewählten Stelle einfügen. Wählen Sie zuerst diese Funktion

und fügen Sie dann mit einem Klick auf die gewünschte Stelle das Element ein. Das Element in der Zwischenablage wird neben dem Mauszeiger dargestellt, bis Sie entweder das Element eingefügt haben, eine andere Funktion ausgewählt haben oder auf die Taste *Escape* drücken.

- Löschen: Mit dieser Funktion können einzelne Elemente gelöscht werden. Wenn unter diesem Knoten (z.B. Verzweigung) noch weitere Elemente enthalten sind, werden Sie gefragt, ob Sie diesen Knoten wirklich löschen wollen. Klicken Sie einfach auf das Element und wählen Sie dann diese Funktion aus.
- Aktivitätseigenschaften: Diese Funktion öffnet die Detailansicht zu dieser Aktivität (Batch), wo sie Akteure und Formulare hinzufügen können.
- **Taskeigenschaften:** Mit dieser Funktion werden die Task-Eigenschaften für diese Aktivität geöffnet (siehe Kapitel 6.2).
- Zeitmanagement: Wenn Sie für eine Aktivität oder einen Batch diese Funktion ausgewählt haben, können Sie eine vorher erstellte Dauerstatistik für diese Aktivität auswählen oder selbst Werte eintragen. Weiters kann eine Übersicht über die in Beziehung stehenden Histogramme des Schrittes angezeigt werden (siehe Abschnitt 7.2.7).
- Annotieren: Mit Hilfe dieser Funktion kann ein zuvor ausgewählter Knoten mit einer Anmerkung versehen werden, d.h. es wird eine Verlinkung zwischen Knoten und Anmerkung erzeugt. Machen Sie einen Doppelklick in das Textfeld, um einen Text hinzuzufügen und bestätigen Sie die Eingabe mit *Return*.
- Ausnahmebehandlung: Diese Funktion ist nur für den Knoten Ausgangsnachricht (INVOKE) verfügbar und erlaubt die Definition eines Exception Flows bzw. das Entfernen eines bestehenden. Der Exception Flow (Ausnahmebehandlung) wird dann aufgerufen, falls beim INVOKE-Knoten etwas schief läuft (z.B. Server antwortet nicht).
- Zusätzliche Kante: Mit Hilfe dieser Funktion können Sie den Elementen *Choice*, *UND-Parallelität* und *ODER-Parallelität* eine weitere Kante (Pfad) hinzufügen. Selektieren Sie dafür das gewünschte Objekt und wählen Sie danach diesen Menüpunkt aus.
- Alles markieren: Durch diese Funktion werden alle Elemente im Zeichenfenster markiert.
- Markierung umkehren: Diese Funktion markiert alle jene Elemente im Zeichenfenster, die vorher nicht markiert waren.

Das Menü Ansicht

 Miniaturansicht: Durch Aktivieren dieser Funktion öffnet sich ein kleines Popup-Fenster, wo eine Übersicht über das Zeichenfenster dargestellt wird. Damit kann bei größeren Prozessen ein besserer Überblick bewahrt werden. Hierbei haben Sie die Möglichkeit durch Ziehen des roten Vierecks die Position im Zeichenfenster zu verändern. Um den ausgewählten Bereich in der Miniaturansicht zu vergrößern bzw. zu verkleinern (verändert sich auch im Zeichenfenster), verwenden Sie bitte die nachfolgend erklärte Zoom–Funktion.

- Zoom: Diese Funktion beinhaltet folgende 3 Subfunktionen:
 - 1. *Normale Ansicht*: Das Zeichenfenster wird in der Größe angezeigt, die beim Start des Prozesseditors vorgegeben ist.
 - 2. Vergrößern: Der angezeigte Bereich wird vergrößert.
 - 3. Verkleinern: Der angezeigte Bereich wird verkleinert.
- Ausrichten: Mit dieser Funktion können die Elemente im Zeichenfenster entsprechend dem äußersten Element (links, rechts, oben, unten) ausgerichtet werden.
- Endknoten anzeigen: Diese Funktion markiert den Endknoten des ausgewählten Elements.
- Automatisch Routen: Sie haben beim orthogonalen Routen die Möglichkeit Kanten zu verschieben. Wählen Sie die Kante aus und ziehen Sie sie in die entsprechende Richtung. Damit die Kante wieder automatisch geroutet wird, wählen Sie zuerst die Kante und anschließend diese Funktion aus.
- Einstellungen: Mit Hilfe dieser Funktion können Sie folgende Einstellungen tätigen:
 - Am Raster ausrichten: Wenn diese Checkbox aktiviert ist, werden die Elemente und Pfade nach den Rasterlinien ausgerichtet.
 - Rasterart: Hier können Sie zwischen 4 möglichen Varianten des Rasters wählen:
 - 1. Unsichtbar: Raster wird ausgeblendet.
 - 2. Punktraster: Punktdarstellung des Gitters.
 - 3. Karoraster: Karodarstellung des Gitters.
 - 4. Linienraster: Liniendarstellung des Gitters.
 - Lineale: Im Zeichenfenster wird am oberen und linken Rand ein Lineal angezeigt. Dabei kann folgendes ausgewählt werden:
 - 1. keine: Es werden keine Lineale angezeigt.
 - 2. in Zentimeter: Die Lineale erhalten als Maßeinheit Zentimeter.
 - 3. in Zoll: Die Lineale erhalten als Maßeinheit Zoll.
 - Seitenränder anzeigen: Diese Funktion zeigt die Seitenränder im Zeichenfenster an.
 - Neue Prozesse in BPMN zeichnen: Wenn diese Checkbox aktiviert ist, so werden neue Prozesse automatisch in der BPM-Notation erzeugt.
 - Neue Prozess orthogonal routen: Wenn diese Funktion aktiviert ist, werden die Pfade bei neu erstellten Prozessen orthogonal dargestellt.
 - Hilfskanten nicht anzeigen: Wenn diese Checkbox aktiviert ist, dann werden Hilfskante von z.B. einem GOTO nicht angezeigt. Hilfskanten werden im Prozesseditor als hellgraue strichlierte Linie angezeigt.

- Goto-Hilfe nicht anzeigen: Wenn diese Checkbox aktiviert ist, wird das Hilfefenster beim Einfügen des Gotos nicht angezeigt.
- Drucker-Zoom: Hier können Sie einstellen, wie vergrößert bzw. verkleinert der Prozess ausgedruckt werden soll.
- Applet Look-and-Feel: Legen Sie hier das Look-and-Feel f
 ür den Prozesseditor fest:
 - gemäß Konfiguration: Es wird das Schema verwendet, dass unter Administration → Konfiguration → Lokalisierung eingestellt wurde (siehe Installationshandbuch - Kapitel 3).
 - 2. metal
 - 3. windows

Das Menü Hilfe

- Hilfe: Hier wird die Hilfe-Seite von @enterprise angezeigt (siehe Kapitel 2).
- Über @enterprise Prozesseditor: Zeigt Informationen über den Prozesseditor und die verwendeten Bibliotheken an.

Die Symbolleiste

Über die Symbole haben Sie die Möglichkeit, die am häufigsten verwendeten Funktionen schnell zu erreichen, ohne in den vorher beschriebenen Menüs danach zu suchen:

- Neu: siehe Abschnitt 7.2.2
- Öffnen: siehe Abschnitt 7.2.2
- Speichern: siehe Abschnitt 7.2.2
- Rückgängig: siehe Abschnitt 7.2.2
- Wiederherstellen: siehe Abschnitt 7.2.2
- Ausschneiden: siehe Abschnitt 7.2.2
- Kopieren: siehe Abschnitt 7.2.2
- Einfügen: siehe Abschnitt 7.2.2
- Löschen: siehe Abschnitt 7.2.2
- Miniaturansicht: siehe Abschnitt 7.2.2
- Normale Ansicht: siehe Abschnitt 7.2.2
- Vergrößern: siehe Abschnitt 7.2.2
- Verkleinern: siehe Abschnitt 7.2.2

- Aktivitätseigenschaften bei Doppelklick auf Knoten anzeigen: Wenn diese Schaltfläche aktiviert ist und Sie auf einen Knoten einen Doppelklick ausführen, werden die *Aktivitätseigenschaften* angezeigt.
- **Taskeigenschaften bei Doppelklick auf Knoten anzeigen:** Wenn diese Schaltfläche aktiviert ist und Sie auf einen Knoten einen Doppelklick ausführen, werden die *Taskeigenschaften* angezeigt.
- Zeitmanagement bei Doppelklick auf Knoten anzeigen: Wenn diese Schaltfläche aktiviert ist und Sie auf einen Knoten einen Doppelklick ausführen, wird das Zeitmanagement angezeigt.

Das Kontextmenü

Das Kontextmenü stellt eine schnelle und bequeme Art der Bedienung im Zeichenfenster dar. Sie erreichen dieses Menü, indem Sie im Zeichenfenster ein Element auswählen und dabei die rechte Maustaste betätigen. Das Kontextmenü beinhaltet dabei zum Teil die gleichen Elemente wie die Menüleiste:

- Ausschneiden: siehe Abschnitt 7.2.2
- Kopieren: siehe Abschnitt 7.2.2
- Löschen: siehe Abschnitt 7.2.2
- Aktivitätseigenschaften: siehe Abschnitt 7.2.2
- Taskeigenschaften: siehe Abschnitt 7.2.2
- Zeitmanagement: siehe Abschnitt 7.2.2
- Annotieren: siehe Abschnitt 7.2.2
- Ausnahmebehandlung: siehe Abschnitt 7.2.2
- Zusätzliche Kante: siehe Abschnitt 7.2.2



Hinweis: Um schneller im Prozesseditor arbeiten zu können, werden *Shortcuts* und *Mnemonics* unterstützt. Der jeweilige Shortcut zu einer Funktion wird gleich daneben angezeigt.

7.2.3 Prozesseigenschaften

Mit Hilfe der Prozesseigenschaften–Maske können Sie Einstellungen bezüglich des Prozesses treffen. Die einzelnen Reiter werden hier nun beschrieben:

- Allgemein: Analog zu 6.5, jedoch ist das Feld Änderung durchführen am nicht vorhanden.
- Formulare: Hier können Sie die Formulare des Prozesses, den Sie gerade bearbeiten (oder bearbeiten möchten), bestimmen. Durch einen Mausklick auf die Schaltfläche *Hinzufügen*, öffnet sich ein neues Fenster (siehe Abbildung 7.4), in dem Sie zunächst ein Formular auswählen und anschließend weitere Optionen zu einem Formular angeben können. Der Dialog *Formular hinzufügen* beinhaltet folgende Informationen:

- Id: Hier muss die Id f
 ür die eindeutige Identifizierung des Prozessformulars eingegeben werden.
- Name: Der Name für das Prozessformular kann hier festgelegt werden.
- Typ: In dieser Tabelle werden sämtliche Formulartypen des Systems aufgelistet. Mit dem Button *Neu* rechts neben der Liste können Sie weitere Formulare hinzufügen (siehe Kapitel 6.4).
- Mode: Hier können Sie den Verwendungszweck des hinzufügenden Formulars näher spezifizieren.

lokal: Das Formular wird nur innerhalb des Prozesses verwendet und wird somit innerhalb des Prozesses erzeugt.

inout: Das Formular wird dem Prozess beim Aufruf übergeben, im Prozess bearbeitet und danach wieder nach außen weitergeleitet.

- Basisformular f
 ür View: W

 ählen Sie hier das Basisformular aus, wo eine View
 hinzugef

 ügt werden soll.
- Ok: Durch einen Mausklick auf diese Schaltfläche wechseln Sie wieder in das Fenster *Prozesseigenschaften*, wobei das von Ihnen eingegebene Formular nun in der Formularliste des Prozesses aufscheint.
- Abbrechen: Durch einen Mausklick auf diese Schaltfläche wechseln Sie wieder zum Fenster *Prozesseigenschaften*, jedoch ohne ein Formular eingefügt zu haben.

Durch einen Mausklick auf die Schaltfläche *Entfernen* können Sie ein Formular aus der Liste dieses Prozesses löschen. Mit der Schaltfläche *Bearbeiten* haben Sie die Möglichkeit, das Formular zu bearbeiten. Bei Änderung einer bestehenden Formular-Id werden bei bestehenden Objekten wie z.B. Aktivitäten die Id's automatisch ersetzt. Bei Strukturen, die z.B. eine Bedingung beinhalten, müssen Sie die Id selbst ändern, anderenfalls werden Sie beim Speichern des Prozesses darauf hingewiesen. Vor dem Aufruf der Funktionen *Bearbeiten* bzw. *Löschen* muss ein Formular ausgewählt sein.

- Source: Analog zu 6.5.
- Komponenten: Analog zu 6.5.
- Formular-Sichtbarkeiten: Analog zu 6.5.
- Eskalation: Analog zu 6.5.
- **Historie:** Analog zu 6.5.
- **Zugriff:** Analog zu 6.5.
- **Referenziert von:** Analog zu 6.5.

7.2.4 Tasks

Mit Hilfe der Task–Maske (siehe Abbildung 7.6) legen Sie die Tasks fest, die Sie beim Ändern eines Akteurs im Arbeitskorb einem Empfänger in der Empfängerliste zuordnen

Formular hinzufügen - Mozilla Firefox					
ld:	form_vacation				
Name:	Urlaubsformular				
Тур:	hr_costcenter (Kostenstelle 1) Image: Costcenter (Kostenstelle 1) hr_dailyrateitem (Tagessatz 1) hr_evaluation (Bewertung 1) hr_expense (Expense 1) hr_recruiting (Personalauswahl 1) hr_sickness (Sickness 1) hr_timeitem (Time Item 1) hr_vacation (Vacation 1)				
Mode:	💿 lokal 🔘 inout				
Basisformula	r. 💌				
Ok	Abbrechen				

Abbildung 7.4: Formular hinzufügen

🥹 Eigenschaften des Prozesses: Urlaub (1) - @enterprise - Mozilla Firefox					
1 http://localhost:8380/wf/servlet.method/com.dec.avw.html.HTMLProcessDefinition.showEditorConsole	%node=admin.procd	efinition&fore	ignKey=ap	plication&application=com.	dec.avw.core.At 🏠 🛞
Allgemein Formulare Source Komponenten Formular-Sichtbarkeiten Eskalat	ion Funktionen	Historie	Zugriff	Ordnereinstellungen	Referenziert von
<pre>process hr_vacation(form_vacation vacation "form_vacation") version 1; name "proc_vacation"; description "desc_proc_vacation"; subject "(form_vacation.employee), (form_vacation.vacfrom,date) timeoutaction none; application staffprocs; begin "New" all vacation_request(form_vacation); manager vacation_approve(form_vacation); if xpath:"\$form_form_vacation/approved = ""true""" then "Vacation_approve(form_vacation); hr!home vacation_process(form_vacation); form_vacation.employee vacation_informapproved(form_vacation_vacation_vacation_vacation_vacation); else form_vacation.employee vacation_informrefused(form_vacation_vaca</pre>	<pre>- (form_vacat tion approved ation););</pre>	ion.vac	to,date	s)";	
				XWDL anzeigen	WDL downloaden
		Ok		Abbrechen	Übernehmen

Abbildung 7.5: WDL Anzeige

können. Diese Funktion wird standardmäßig im Arbeitskorb nicht angeboten und muss erst in der GUI-Konfiguration beim Knotentyp *Arbeitskorb* \rightarrow *Funktionen* hinzugefügt werden (*Action Key:* adHoc). Weiter Details zur GUI-Konfiguration \rightarrow Knoteneigenschaften finden Sie im Kapitel 6.8.1.

🕽 Tasks - Mozilla Firefox		
http://localhost:8380/wf/servlet.method/com.dec.avw.html.HTMLProc	ess	;Definition.showEditorConsole?object=com.dec.avw.core.ProcessDefinition 🏠 🤇
Tasks:		Hinzugefügte Tasks:
a_tranch (1) a_task (3) Abgelehnt (1) adm_task (1) andpar1_task (1) andpar2_task (1) andpar3_task (1) Anforderm (1) Arbeitszeit erfassen (1) Auswahl (1) Autorenkorrektur (1)		adHoc (1)
Formulare		Hinzugefügte Formulare:
form_vacation		
		Ok Abbrechen

Abbildung 7.6: Tasks

Task hinzufügen

Folgende Schritte sind notwendig:

- 1. Wählen Sie im Menü *Prozess* den Eintrag *Tasks* aus. Es öffnet sich der in Abbildung 7.6 dargestellte Dialog.
- 2. Wählen Sie in der Liste Tasks einen Task aus.
- 3. Aktivieren Sie die Schaltfläche Hinzufügen (Pfeil nach rechts).
- 4. Wenn Sie einem Task ein Formular hinzufügen wollen, dann sind folgende Schritte notwendig:
 - (a) Wählen Sie aus der Liste Hinzugefügte Tasks einen Task aus.
 - (b) Wählen Sie aus der Liste Formulare ein Formular aus.
 - (c) Aktivieren Sie die Schaltfläche *Hinzufügen* (Pfeil nach rechts). Nun erscheint das hinzugefügte Formular in der Liste *Hinzugefügte Formulare*. (Wenn Sie dem Task mehrere Formulare hinzufügen wollen, dann wiederholen Sie die Schritte b und c entsprechend oft.)
- 5. Aktivieren Sie die Schaltfläche OK.

Wenn Sie mehrere Tasks hinzufügen möchten, wiederholen Sie einfach die Schritte 2 bis 4 entsprechend oft, bevor Sie Schritt 5 ausführen.

Task entfernen

Folgende Schritte sind notwendig:

- 1. Wählen Sie im Menü *Prozess* den Eintrag *Tasks* aus. Es öffnet sich der in Abbildung 7.6 dargestellte Dialog.
- 2. Wählen Sie in der Liste Hinzugefügte Tasks einen Task aus.
- 3. Aktivieren Sie die Schaltfläche Entfernen (rotes Kreuz).
- 4. Aktivieren Sie die Schaltfläche OK.

Wenn Sie mehrere Tasks entfernen möchten, wiederholen Sie einfach die Schritte 2 und 3 entsprechend oft, bevor Sie Schritt 4 ausführen.

Löschen eines Formulars, das einem Task zugeordnet ist

Die folgenden Schritte sind notwendig:

- Wählen Sie in der Liste *Hinzugefügte Tasks* jenen Task aus, dem das Formular, das Sie löschen möchten, zugeordnet ist. Nun können Sie das Formular, das Sie löschen möchten, in der Liste *Hinzugefügte Formulare* sehen.
- 2. Wählen Sie das Formular in der Liste Hinzugefügte Formulare aus.
- 3. Aktivieren Sie die Schaltfläche Löschen (neben der Liste Hinzugefügte Formulare).
- 4. Aktivieren Sie die Schaltfläche Ok.

Wenn Sie mehrere Formulare löschen wollen, wiederholen Sie die Schritte 1 bis 3 entsprechend oft, bevor Sie Schritt 4 ausführen.

7.2.5 Timeout Task

Mit Hilfe der Timeout-Maske können Sie eine Aktivität definieren, die im Falle der Zeitüberschreitung (Feld *Max. Dauer* - siehe Kapitel 6.5) automatisch vom System aktiviert wird und in den Arbeitskorb des angegebenen Bearbeiters gestellt wird. Dieser kann dann entscheiden, wie der Prozess weiter abgewickelt wird.

Folgende Schritte sind zum Festlegen einer Timeout-Behandlung notwendig:

- 1. Wählen Sie im Menü Prozess den Eintrag Timeout Task aus.
- Wählen Sie den Namen der auszuführenden Aktivität direkt oder über das Auswahlfenster aus.
- 3. Wählen Sie jenen Akteur aus, der mit der Timeout-Behandlung konfrontiert werden soll.
- 4. Fügen Sie jene Formulare hinzu, die für die Timeoutbehandlung notwendig sind.
- 5. Bestätigen Sie Ihre Eingabe mit der Schaltfläche *Ok* oder brechen Sie den Vorgang mit *Abbrechen* ab.

Eigenschaften des Timeout-Tasks - O - Mozilla Firefox					
🐞 http://localho	ost:8380/wf/servlet.meth	od/c	om.dec.avw.html.HTMLProcessDefinition.showEditorConsole?ob 😭 🤅		
Task:	timeout_task		2 × ±		
Knotenname:					
Akteur(e):	Alle		<u> </u>		
			×		
			✓ ±		
Verfügbare Fo	ormulare:		Hinzugefügte Formulare:		
form_vacatio	n 🔼		▲ <i>P</i>		
			1		
		Þ	*		
	~				
Löschen			Ok Abbrechen		

Abbildung 7.7: Timeout–Maske



Hinweis: Diese Form der Timeout–Behandlung ist nur noch aus abwärtskompatiblen Gründen vorhanden. Verwenden Sie daher für neu erstellte Prozesse entweder die Prozess– Eskalationen (siehe Abschnitt 6.5.10) für den ganzen Prozess oder die Task–Eskalationen (siehe Abschnitt 6.2.2) für einzelne Tasks. Die Prozess–Eskalationen sind unter *Prozess* \rightarrow *Eigenschaften* \rightarrow *Eskalation* zu finden. Die Task–Eskalationen können durch Selektieren einer Aktivität und anschließend unter *Bearbeiten* \rightarrow *Taskeigenschaften* erreicht werden.

7.2.6 Eigenschaften einer Aktivität definieren

Wählen Sie eine Aktivität aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus, wenn die Funktion *Aktivitätseigenschaften bei Doppelklick auf Knoten anzeigen* aktiviert ist oder klicken Sie mit der rechten Maustaste auf den Knoten und rufen Sie im Kontextmenü *Aktivitätseigenschaften* auf. Es erscheint nun am Bildschirm die Maske *Eigenschaften für Aktivitäten* (siehe Abbildung 7.8), mit deren Hilfe Sie die Defaultwerte einer Aktivität überschreiben und bestimmte Eigenschaften definieren können.

Folgende Eigenschaften können bestimmt werden:

- Aktivität: Legt die Aktivität fest durch Einfügen eines Tasks.
- Knotenname: Bezeichnung des Knotens, der auch internationalisiert werden kann, indem der zu übersetzende Wert mit @@@ beginnt und mit @@ aufhört, z.B. @@@myname@@.
- Label: Dieser muss innerhalb des Prozesses eindeutig sein und unterliegt den gleichen

) Eigenschafte	n des Tasks 'Anfordern' - 4 - Mozilla Firefox	_ 0 >
🐁 http://localho:	st:8380/wf/servlet.method/com.dec.avw.html.HTMLProcessDefinition.showEditorCons	ole?ob 🏠 🔞
Task: Knotenname:	Anfordern 🖉 🎾	÷
Label: Icon:	label_4	
Akteur(e):	Alle	
Überspringbar Verfüqbare Fo	rmulare: Hinzugefügte Formulare:	
	form_vacation	
	Ok Abb	rechen

Abbildung 7.8: Eigenschaften für Aktivitäten

syntaktischen Bedingungen wie eine @enterprise-Id. Ein eingegebener Label wirkt sich bei der Akteurauswahl für andere Aktivitäten aus.

- Icon: Das Icon, durch das die Aktivität im Prozesseditor dargestellt wird. Wird hier kein Wert angegeben, so wird das Default Icon von @enterprise verwendet. Ein Icon wird wie eine Ressource behandelt, die im Klassenpfad vorhanden ist. Beispiel: Pfad *lang/default/images/pred/nodes/event_register.png* liefert Icon des Knoten *Event registrieren*.
- Akteur(e): Die Namen der Akteure (Rolle, Benutzer, Task, Formularfeld, Methode).
- Überspringbar: Ist diese Checkbox aktiviert und kein Akteur eingetragen, wird diese Aktivität ausgelassen.
- Formulare: Gibt an, welche Prozessformulare in dieser Aktivität verwendet werden. Die Reihenfolge der Prozessformulare kann über die Sortierbuttons neben der Liste verändert werden, d.h. das oberste Formular wird in der Reiter-Ansicht im Arbeitskorb ganz links angezeigt.

Folgende Funktionen stehen zur Verfügung:

• Neue Aktivität: Hierbei öffnet sich die Detailansicht zu einem neuen Task, wo Sie der Aktivität diesen neu erstellten Task hinzufügen können (siehe Abschnitt 6.2).
- Auswahl: Durch einen Mausklick auf diese Schaltfläche wird ein Fenster angezeigt, das es Ihnen ermöglicht eine bereits in der Taskverwaltung definierte Aktivität auszuwählen. Dafür gehen Sie wie folgt vor:
 - 1. Wählen Sie eine Aktivität aus der Liste aus.
 - 2. Klicken Sie auf die Schaltfläche Ok.
 - 3. Wenn Sie auf die Schaltfläche *Abbrechen* klicken, kehren Sie zum vorigen Fenster zurück und Ihre Eingaben werden verworfen.
- Akteur hinzufügen: Durch einen Mausklick auf die Schaltfläche Akteur hinzufügen wird ein neues Fenster angezeigt, das es Ihnen ermöglicht einen Akteur für die aktuelle Aktivität auszuwählen.

Wählen Sie aus der Liste, die Ihnen dann angeboten wird, einen Akteur aus. Bevor Sie dies tun, müssen Sie angeben wie der Akteur definiert wurde, d.h. ein Akteur kann sein:

- Benutzer: Registrierter Benutzer.
- *Rolle:* Spezifiziert eine Rolle, die Organisationseinheit eines Prozesses wird nicht verändert.
- Org.-Einheit und Rolle: Es kann eine Organisationseinheit und eine Rolle ausgewählt werden.
- Org.-Einheit und Benutzer: Es kann eine Organisationseinheit und ein Benutzer ausgewählt werden.
- Task: Es wird der Akteur einer früheren Aktivität genommen.
- *Formularfeld:* Die Rolle wird zur Laufzeit einem Formularfeld entnommen, dessen Inhalt dann einer Rollen- oder Benutzer-Id entsprechen muss.
- Methode: Angabe einer Java-Methode, die als Akteur dient (kein Groovy-Skript).
 Rückgabewert muss ein Agent oder ein String in WDL-Syntax sein.

Durch einen Mausklick auf die Schaltfläche *OK* wird der ausgewählte Akteur in die Liste der Akteure der vorigen HTML–Seite aufgenommen, und Sie kehren dorthin zurück.

Durch einen Mausklick auf die Schaltfläche *Abbrechen* kehren Sie zur vorigen HTML-Seite zurück, ohne einen Akteur zu übernehmen.

- Akteur entfernen: Um einen bestimmten Akteur zu entfernen, müssen Sie diesen aus der Liste der Akteure durch einen Mausklick auswählen, und anschließend mit der Maus auf die Schaltfläche X klicken.
- Formular hinzufügen: In der Tabelle *Verfügbare Formulare* werden alle Formulare aufgelistet, die für den konkreten Prozess (im Menü *Prozess* Menüpunkt *Eigenschaften/Attribute*) definiert wurden. Durch Auswählen eines Formulars und Aktivierung der Pfeil-Schaltfläche können Sie Formulare zur Aktivität hinzufügen. Die Formulare, die einer Aktivität hinzugefügt wurden, stehen rechts in der Tabelle *Hinzugefügte Formulare*.

- Formular entfernen: Aus der Liste der hinzugefügten Formulare können Sie ein Formular entfernen, wenn Sie die Schaltfläche *X* rechts neben dieser Liste betätigen.
- Formular-Sichtbarkeiten: Mit Hilfe des Schraubenschlüssel–Symbols auf der rechten Seite neben der Liste *Hinzugefügte Formulare* können Sie die Formular–Sichtbarkeiten für diese Aktivität festlegen, indem Sie ein Formular aus der Liste auswählen und auf dieses Symbol klicken (Analog zu Kapitel 6.5).
- **OK:** Durch Aktivierung des Befehls *OK* bestätigen Sie Ihre Eingaben und beenden das Eingabefenster.
- Abbrechen: Durch Aktivierung des Befehls *Abbrechen* verwerfen Sie Ihre Eingaben und beenden das Eingabefenster.

7.2.7 Zeitmanagement

Mit Hilfe des Zeitmanagements ist es möglich, Dauer und Resultat über den im Kapitel 9.1.15 erstellen Zeitgraphen anzusehen. Jeder interaktive Task (Aktivität oder Batch) sollte diese Daten beinhalten (TimeNode). Wenn nicht, besteht die Möglichkeit im Reiter *Dauer* die Werte selbst einzutragen oder die entsprechende Statistik aus der Drop–Downliste auszuwählen. Zuvor müssen Sie wählen, ob Sie die Werte händisch eintragen wollen oder ob Sie über den Link *Erstellen* den Mining-Task aufrufen und eine Dauerstatistik für diesen Prozess generieren wollen. Der Reiter *Resultat* beinhaltet dann eine Übersicht über die in Beziehung stehenden Histogramme des Schrittes. Für jeden Schritt gibt es 5 Arten von Zeithistogramme:

- 1. Dauer (blau): Zeigt an, wie lange es braucht, diesen Schritt durchzuführen.
- Frühest mögliche Startzeit (grün): Zeigt den bestmöglichen Startzeitpunkt dieses Tasks an. Dieses Histogramm wird nur generiert, wenn der Parameter Starthistogramme erzeugen unter Konfiguration → Zeitverwaltung zum Zeitpunkt der Graphgenerierung gesetzt war.
- 3. Frühest mögliche Endzeit (gelb): Zeigt den bestmöglichen Endzeitpunkt dieses Tasks an.
- 4. **Spätest erlaubte Startzeit (orange):** Beinhaltet Informationen über den spätest erlaubten Startzeitpunkt (aus der Sicht der Prozessfertigstellung). Negative Werte werden von der Prozess–Deadline errechnet.
- Spätest erlaubte Endzeit (magenta): Beinhaltet Informationen über den spätest erlaubten Endzeitpunkt (aus der Sicht der Prozessfertigstellung).

Positive Werte ergeben sich vom Prozessstart, negative Werte von der Prozess-Deadline.

Jedes Histogramm kann auf folgende Arten dargestellt werden:

- Normale Ansicht
- Akkumulierte Ansicht

Die akkumulierte Ansicht ermöglicht ein einfacheres Herausfinden von Werten wie: Wie lang werden Y% der Schritte dauern oder wieviel Prozent wurden bist zum Zeitpunkt X erledigt (siehe Abbildung 7.9).

7.2. GRAFISCHE PROZESSDEFINITION



Abbildung 7.9: Resultat des Zeitgraphen

7.2.8 Die Funktionsleiste

In der Funktionsleiste befinden sich sämtliche Funktionen zur grafischen Modellierung von Prozessen in der Zeichenfläche. Mit der Maus können Sie in der Funktionsleiste den gewünschten Modus auswählen, um z.B. Aktivitäten anzulegen, Verbindungen zwischen Aktivitäten herzustellen oder grafische Objekte zu löschen.

Zum Einfügen einer Aktivität, eines Subprozesses oder Kontrollflusses wählen Sie die entsprechende Funktion aus, und klicken dann auf eine Kante im Prozessgrafen. Wenn das orthogonale Routing aktiviert wurde, können die Knoten durch Gedrückthalten der *Shift-Taste* und Ziehen mit der Maus horizontal bzw. vertikal verschoben werden. Ausgangspunkt für die Verschiebungsrichtung ist der ursprüngliche Standpunkt des Knotens.

Die Funktionen im Einzelnen:

• Markierung: Den Zeigermodus aktivieren Sie über die Funktion Markierung. In die-

sem Modus können Sie grafische Objekte (Aktivitäten) am Bildschirm neu positionieren und bearbeiten.

Wählen Sie dazu mit der Maus eine Aktivität aus und halten Sie die linke Maustaste gedrückt. Sie können nun die Aktivität an eine beliebige Stelle im Zeichenfenster bewegen. Weiters haben Sie auch die Möglichkeit nach dem Drag und Drop Prinzip Objekte im Zeichenfenster neu zu positionieren bzw. zu löschen.

- **Task:** Im Aktivitätsmodus können Sie neue Aktivitäten anlegen und die Eigenschaften von Aktivitäten spezifizieren. Folgende Vorgangsweise ist dazu erforderlich:
 - Gehen Sie in den Aktivitätsmodus, indem Sie mit der linken Maustaste in der Funktionsleiste auf die Schaltfläche Aktivität klicken.
 - Positionieren Sie nun den Cursor im Zeichenfenster auf jene Kante, wo Sie eine neue Aktivität einfügen wollen.
 - Drücken Sie die linke Maustaste und eine neue Aktivität wird angelegt.
 - Führen Sie unmittelbar danach einen Doppelklick auf der Aktivität aus, um die Eigenschaften der Aktivität zu definieren. Diese Aktion können Sie auch nachträglich im Zeigermodus ausführen (siehe Zeigermodus oben).
- Subprozess: Analog dem Vorgehen beim Einfügen einer Aktivität können Subprozesse eingefügt werden.
- **Systemschritt:** Im Systemschrittmodus können Sie neue Systemschritte anlegen und die Methode für den jeweiligen Systemschritt angeben. Folgende Vorgangsweise ist dazu erforderlich:
 - Gehen Sie in den Systemschrittmodus, indem Sie mit der linken Maustaste in der Funktionsleiste auf die Schaltfläche *Systemschritt* klicken.
 - Positionieren Sie nun den Zeiger im Zeichenfenster an jener Stelle, wo Sie einen neuen Systemschritt einfügen wollen.
 - Drücken Sie die linke Maustaste und ein neuer Systemschritt wird angelegt.
 - Führen Sie unmittelbar danach einen Doppelklick auf den Systemschritt aus, um die Methode, die in diesem Systemschritt ausgeführt werden soll, zu definieren. Diese Aktion können Sie auch nachträglich im Zeigermodus ausführen (siehe Zeigermodus oben).
- **Batchschritt:** Analog zu *Systemschritt*. Allerdings wird hier der Name einer BatchAdapter Klasse eingetragen und es besteht die Möglichkeit, das Verhalten des BatchJobs zur Laufzeit detaillierter zu bestimmen.
- Verzweigung: Eine Verzweigung besteht aus einem If-Knoten (dargestellt durch ein grünes Kästchen), der die Bedingung enthält, und einem End-Knoten, der die Wege wieder zusammenführt. Vom If-Knoten zum End-Knoten gehen eine rote und eine grüne Kante. Bei der Ausführung des Workflows wird zuerst die Bedingung überprüft. Wenn diese wahr ist, wird die grüne Kante weiterverfolgt, andernfalls die rote.

Wenn Sie im Fall einer falschen Bedingung (rote Kante) nicht gleich eine Aktivität ausführen möchten, sondern die Nachfolgeaktivität wieder von einer Bedingung abhängig machen wollen, so können Sie wieder eine Verzweigung einführen, indem Sie in den Verzweigungsmodus wechseln und auf den roten Pfeil klicken, der die Verzweigungsaktivität mit der Nachfolgeaktivität verbindet.

Durch diese Vorgehensweise wird kein neues *if-end* erzeugt, sondern eine zusätzliche Bedingung in das bereits bestehende *if-end* eingefügt, wodurch im WDL-Script folgende Anweisungen entstehen:

```
if Bedingung 1 Then
Aktion 1
elsif Bedingung 2 Then
Aktion 2
elsif Bedingung 3 Then
Aktion 3
else
Aktion 4
end
```

• Auswahl: Jeder Auswahlzweig hat einen Namen und eine optionale Bedingung. Zuerst werden die Bedingungen aller Zweige ausgewertet, die Zweige mit wahren Bedingungen werden dem Benutzer zur Auswahl angezeigt.

Um einen Choicezweig einzufügen, ist folgende Vorgangsweise erforderlich:

- Gehen Sie in den Choicemodus, indem Sie mit der linken Maustaste in der Funktionsleiste auf die Schaltfläche Choice klicken.
- Positionieren Sie nun den Cursor im Zeichenfenster auf jener Kante, wo Sie einen neuen Choicezweig einfügen wollen.
- Drücken Sie die linke Maustaste und eine neue Choice wird angelegt.
- Sie sehen nun einen schwarzen Pfeil, wobei der schwarze Pfeil für einen möglichen Choicezweig steht, zu dem Sie, wie oben beschrieben, Aktivitäten hinzufügen können. Wenn Sie eine weitere Alternative hinzufügen wollen, wählen Sie die Choice aus und aktivieren Sie entweder in der Menüleiste unter *Bearbeiten* den Punkt *Zusatzpfad* oder klicken Sie mit der rechten Maustaste auf die Choice und wählen Sie im Kontextmenü *Zusatzpfad* aus.
- While-Schleife: Mit dem While-Konstrukt können Sie eine Schleife erzeugen, die ausgeführt wird, solange eine Bedingung wahr ist. Ähnlich wie bei der Verzweigung muss auch hier eine Bedingung angegeben werden. Aus dem *While* führt, wenn die Bedingung falsch (geworden) ist, ein roter Pfeil zur nächsten Anweisung. An den grünen Pfeil können Sie z.B. eine Aktivität setzen, die den Körper der While-Schleife bildet und bei wahrer Bedingung ausgeführt wird.

- Schleife: Eine Loop-Schleife führt eine Anweisung aus und prüft danach, ob eine Bedingung wahr ist. An den schwarzen, aus der Loop führenden Pfeil können Sie z.B. eine Aktivität setzen, die mindestens einmal ausgeführt wird. Ist die nachfolgende Bedingung wahr, wird die Schleife verlassen, andernfalls wird die Aktivität nochmals ausgeführt. Die Aktivität nach dem roten Pfeil wird nur ausgeführt, wenn die Bedingung für den Schleifenabbruch falsch ist. Die Aktivität nach dem grünen Pfeil wird ausgeführt, sobald die Abbruchbedingung für die Schleife erfüllt ist.
- **Parallel for:** Mit diesem Kontrollkonstrukt kann man über Subformulare oder über einen Iterator iterieren. Das Kontrollkonstrukt beginnt mit einem *parfor*–Knoten und endet mit einem *end*–Knoten in der WD-Notation. In der BPM-Notation wird dieses Kontrollkonstrukt als BPMN-Subprozess dargestellt. Wenn in diesem Fall innerhalb des Rahmens geklickt wird, aber nicht auf ein Element darin, dann wird der Rahmen selektiert. So kann z.B. das Konstrukt verschoben oder gelöscht werden. Ein Doppelklick öffnet denselben Dialog wie ein Doppelklick auf den Startknoten des *Parallel for*.

Weitere Informationen zu diesem Konstrukt entnehmen Sie dem WDL-Teil des Kapitels 7.

- UND-Parallelität: Dieses Konstrukt ermöglicht die Synchronisation paralleler Prozesspfade. Zwischen den speziellen Knoten *par* und *andjoin* können beliebig viele parallele Exekutionspfade erzeugt werden. Wenn Sie weitere Pfade hinzufügen wollen, wählen Sie die Parallelität (par) aus und aktivieren Sie entweder in der Menüleiste unter *Bearbeiten* den Punkt *Zusatzpfad* oder klicken Sie mit der rechten Maustaste auf die Parallelität und wählen Sie im Kontextmenü *Zusatzpfad* aus. Die Weiterführung des Prozesses erfolgt erst, wenn alle Pfade abgearbeitet sind.
- **ODER-Parallelität:** Dieses Konstrukt dient zur Formulierung einer ODER-Parallelität. Es unterscheidet sich von der UND-Parallelität dadurch, dass nur einer der parallelen Pfade beendet werden muss. Die Aktivitäten auf den anderen Pfaden werden dann abgebrochen.
- Abzweigung: Mit einer Abzweigung (Branch) können Sie einen Zweig im Prozess definieren, der unabhängig davon, ob der Prozess bereits beendet wurde oder nicht, abgearbeitet wird.
- Goto: Mit Goto können Sie zu einer beliebigen Aktivität eines Prozesses springen, ohne dass dabei eine Bedingung geprüft wird und erfüllt sein muss. Um auf eine andere Aktivität zu verweisen, klicken Sie auf die Pfeilspitze der durchgezogenen Kante und ziehen Sie diese bei gedrückter Maustaste über den gewünschten Zielknoten und lassen Sie dann die Maustaste los. Die gestrichelte Linie vom Goto aus zeigt den ursprünglichen Verlauf des Prozesses an. Wenn die durchgezogene Kante auf eine Aktivität zeigt, wird der Label dieser Aktivität im Goto-Eigenschaftsfenster angezeigt. Im Goto-Eigenschaftsfenster können Sie den *Label des Zielknoten* angeben. Wenn die durchgezogene Kante bereits auf ein Element im Prozesseditor zeigt und sie den Label im Goto verändern, wird die Änderung im Zielknoten auch übernommen.

Hinweis: In der BPM-Notation gibt es den Sonderfall, dass die durchgezogene Kante nicht die Grenzen (= den Rahmen) eines *Parallel for* überschreiten darf und

zwar weder ein- noch ausgehen (auch nicht bei geschachtelten Parallel for).

Achtung: Durch die Verwendung von Goto kann es vorkommen, dass Sie, ohne es zu wollen, Endlos-Schleifen einbauen und der Endzustand Ihres Prozesses nie erreicht wird!

• Ereignisse: Zu den Ereignissen gehören *Raise-Ereignis*, *Sync-Ereignis*, *Register-Ereignis* und *Unregister-Ereignis*. Der Eventmechanismus wird verwendet, um innerhalb der Engine Events zu erzeugen und zu behandeln. Dabei kann ein Event durch die Ausführung eines Prozesses oder via API erzeugt werden. Ein Event wird von allen Prozessinstanzen, die sich dafür registriert haben, empfangen. Dadurch wird der Event Handler, der durch den Empfänger spezifiziert wurde, gestartet.

Ein Event wird durch einen Namen und ein optionales Kontext–Objekt identifiziert. Wenn die Prozessinstanz, die den Event auslöst, ein Kontext–Objekt definiert, dann stimmt die Handler–Registrierung nur, wenn bei der Registrierung des Handlers:

- das gleiche Kontext-Objekt oder
- kein Kontext-Objekt angegeben wurde.

Als Kontext–Objekt kann entweder ein Formular oder ein Formularfeld angegeben werden. Enthält ein angegebenes Formularfeld ein *com.groiss.store.PersistentObject*, so wird beim Prüfen des Kontexts auf Gleichheit von oid und Klassenname geprüft, ansonsten wird als Kontext das Ergebnis der *toString*–Methode des übergebenen Kontextobjektes herangezogen.

Weitere Informationen zu Events finden Sie im Programmierhandbuch.

- Web-Dienste: Bei den Web-Diensten können folgende Knoten ausgewählt werden:
 - Ausgangsnachricht (INVOKE): Ist dieser Knoten ausgewählt, so wird zur Laufzeit der definierte Web-Dienst aufgerufen und die entsprechenden Daten mitgesendet. Falls es dabei zu einem Fehler kommt und eine Ausnahmebehandlung hinzugefügt wurde, so wird diese ausgeführt (siehe Definition einer Ausnahmebehandlung in Abschnitt 7.2.2).
 - Eingangsnachricht (RECEIVE): Ist dieser Knoten ausgewählt, so wird zur Laufzeit auf Daten (des zuvor aufgerufenen) Web-Dienstes gewartet. Bekommt dieser die Daten, werden diese entsprechend der Definition abgearbeitet.
 - Antwort (REPLY): Ist dieser Knoten ausgewählt, so wird im Falle einer positiven Abarbeitung des Knoten *Eingangsnachricht*, eine Antwort an den Gegenserver gesendet.

Die Eigenschaften-Dialoge der einzelnen Knoten werden im Abschnitt 7.2.15 beschrieben.

• Anmerkung: Wenn Sie diese Funktion ausgewählt haben, können Sie an einer beliebigen Stelle im Zeichenfenster ein Textfeld positionieren, wo Sie Anmerkungen hinzufügen können. Machen Sie einen Doppelklick in das Textfeld um einen Text hinzuzufügen und bestätigen Sie die Eingabe mit *Return*.

7.2.9 Bedingungen für Ifs, Choice, Schleifen

Für die genannten Konstrukte kann nach Doppelklick auf das grafische Symbol eine Bedingung eingegeben werden. Die Syntax dieser Bedingungen ist im Abschnitt 7.1.5 der textuellen Prozessdefinition beschrieben.

Zur Laufzeit wird die Bedingung vom System automatisch ausgewertet.

7.2.10 Die Methode eines Systemschrittes bestimmen

Wählen Sie einen Systemschritt bzw. Batchschritt aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus.

Es erscheint nun am Bildschirm die Maske *Eigenschaften Systemschritt*, mit deren Hilfe Sie die Methode, die bei der Ausführung eines Systemschrittes ausgeführt werden soll, definieren können.

7.2.11 Die Eigenschaften eines Batchschrittes bestimmen

Wählen Sie einen Batch aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus.

Es erscheint nun am Bildschirm die Maske *Eigenschaften Batchschritt*, mit deren Hilfe Sie die BatchAdapter Klasse, deren Methoden bei der Ausführung eines Batchschrittes aufgerufen werden sollen, definieren können. Variationen des Verhaltens der Laufzeit eines BatchJobs können über die Checkboxes angegeben werden. Mehr dazu im Programmierhandbuch und in der API-Dokumentation.

7.2.12 Die Eigenschaften eines Subprozesses bestimmen

Wählen Sie einen Subprozess aus und führen Sie mit der linken Maustaste einen Doppelklick aus.

Es erscheint nun am Bildschirm die Maske *Eigenschaften* für Subprozesse, mit deren Hilfe Sie die Eigenschaften eines Subprozesses definieren können.

Folgende Eigenschaften können definiert werden:

- **Prozess:** Der Name des Subprozesses.
- Knotenname: Frei wählbarer Name für diesen Knoten, der dann angezeigt werden soll; kann auch internationalisiert werden, indem der zu übersetzende Wert mit @@@ beginnt und mit @@ aufhört, z.B. @@@myname@@.
- Icon: Pfad zum Anzeigen eines alternativen Icons, das im Klassenpfad von @enterprise liegt und als Ressource behandelt wird.
- Formulare: Formulare, die dem Subprozess übergeben werden.

Folgende Funktionen stehen zur Verfügung:

7.2. GRAFISCHE PROZESSDEFINITION

- Auswahl: Durch das Anklicken der Schaltfläche mit der Lupe wird Ihnen eine Liste aller zur Verfügung stehenden Prozesse angezeigt, aus der Sie einen auswählen können. Sie erreichen dies, indem Sie:
 - einen Prozess mit der Maus auswählen, und anschließend die Schaltfläche OK anklicken.
 - Durch das Anklicken der Schaltfläche Abbrechen wird der Auswahlvorgang abgebrochen und das Auswahlfenster für Prozesse geschlossen.
- Formular hinzufügen: In der Tabelle *Verfügbare Formulare* werden alle Formulare aufgelistet, die für den konkreten Prozess (im Menü *Prozess* Menüpunkt *Eigenschaften/Attribute*) definiert wurden. Durch Auswählen eines Formulars und Aktivierung der Pfeil-Schaltfläche können Sie Formulare zur Aktivität hinzufügen. Die Formulare, die einer Aktivität hinzugefügt wurden, stehen rechts in der Tabelle *Hinzugefügte Formulare*.
- Formular entfernen: Aus der Liste der hinzugefügten Formulare können Sie ein markiertes Formular entfernen, wenn Sie die Schaltfläche X rechts neben dieser Liste betätigen.
- OK: Durch Aktivierung des Befehls *OK* bestätigen Sie Ihre Eingaben und beenden das Eingabefenster.
- Abbrechen: Durch Aktivierung des Befehls *Abbrechen* verwerfen Sie Ihre Eingaben und beenden das Eingabefenster.

7.2.13 Eigenschaften eines Parallel For definieren

Wählen Sie einen *parfor*-Knoten aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus. Es erscheint nun am Bildschirm die Maske *Parallel For* mit deren Hilfe Sie bestimmte Eigenschaften definieren können.

Folgende Eigenschaften können bestimmt werden:

- für jedes Subformular in: Wenn dieser Radio–Button aktiviert ist wird das Parallel For für die Subformular–Einträge eines Formulars ausgeführt, wie es im WDL–Teil beschrieben ist (siehe Fall 1, Abschnitt *Parallel For* unter 7.1). Aus der Dropdown-Liste können Sie nun das entsprechende Subformular (Hauptformular.Subformular-Id) auswählen.
 - Id des Formulars in der Schleife: Die Id des zuvor ausgewählten Subfomulars innerhalb des Parallel For Konstrukts.
 - Name des Formulars in der Schleife: Name des Subformulars innerhalb des Parallel For Konstrukts.
- **Iterator:** Wenn dieser Radio–Button aktiviert ist, wird das Parallel For für diese Klasse ausgeführt, wie es im WDL–Teil beschrieben ist (siehe Abschnitt *Parallel For* unter 7.1).

• Label: Dieses muss innerhalb des Prozesses eindeutig sein und unterliegt den gleichen syntaktischen Bedingungen wie eine @enterprise-Id.

Im Endknoten eines Parallel For kann zudem eine Methode angegeben werden (siehe Abschnitt *Parallel For* unter 7.1).

7.2.14 Eigenschaften eines Events definieren

Wählen Sie einen Event aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus. Es erscheint nun am Bildschirm die Maske des ausgewählten Events mit deren Hilfe Sie bestimmte Eigenschaften definieren können.

Dabei gibt es folgende vier Events:

- Auslösen (Raise): Ist dieses Ereignis ausgewählt, so wird ein Event ausgelöst.
- Synchronisieren (Sync): Ist dieses Ereignis ausgewählt, so wartet der Prozess an dieser Stelle auf den Event.
- Registrieren (Register): Ist dieses Ereignis ausgewählt, so registriert sich der Prozess für den Event.
- Deregistrieren (Unregister): Ist dieses Ereignis ausgewählt, so wird an dieser Stelle im Prozess die Registrierung für den Event aufgehoben.

Folgende Eigenschaften können bestimmt werden:

- Event Name: Der Name des Events.
- Event Handler: Eine Java-Klasse die das Interface *com.groiss.event.EventHandler* implementiert.
- Kontext: Das Kontextobjekt.
- Knotenname: Frei vergebbarer Name für den Knoten, der auch internationalisiert werden kann, indem der zu übersetzende Wert mit @@@ beginnt und mit @@ aufhört, z.B. @@@myname@@.. Wenn nichts eingegeben wurde, wird der Standard-Knotenname verwendet.
- Label: Dieses muss innerhalb des Prozesses eindeutig sein und unterliegt den gleichen syntaktischen Bedingungen wie eine @enterprise-Id. Das Label ist auch für Prozesseskalationen vom Typ *Sync nicht erfolgt* relevant (siehe Abschnitt 6.5.10).

Folgende Funktionen stehen zur Verfügung:

- **OK:** Durch das Aktivieren der Schaltfläche *Ok* werden Ihre Eingaben übernommen und der Dialog wird geschlossen.
- Abbrechen: Durch das Aktivieren der Schaltfläche *Abbrechen* werden Ihre Eingaben verworfen und der Dialog geschlossen.

7.2.15 Eigenschaften eines Web-Dienste-Knoten definieren

Wählen Sie einen beliebigen Web-Dienste-Knoten aus und führen Sie mit der linken Maustaste einen Doppelklick darauf aus. Es erscheint nun am Bildschirm die Maske des ausgewählten Knotens mit deren Hilfe Sie bestimmte Eigenschaften definieren können. Für jeden Knoten gibt es einen *Knotenname*, der frei vergebbar ist und der auch internationalisiert werden kann, indem der zu übersetzende Wert mit @@@ beginnt und mit @@ aufhört, z.B. @@@myname@@. Wenn nichts eingegeben wurde, wird der Standard-Knotenname verwendet. Weiters gibt es ein *Label*. Dieses muss innerhalb des Prozesses eindeutig sein und unterliegt den gleichen syntaktischen Bedingungen wie eine @enterprise-Id. Das Label ist beim Knoten *Eingangsnachricht* für Prozesseskalationen vom Typ *Eingangsnachricht nicht erhalten* relevant (siehe Abschnitt 6.5.10).

Folgende Eigenschaften können für den Knoten Ausgangsnachricht bestimmt werden:

- Webservice Operation: Sie können aus der Auswahlliste eine bestehende Webservice Client Operation auswählen, die zuvor (für die Applikation) definiert wurde (siehe Kapitel 6.10.1).
- Adresse: Auswahl einer Adresse, um den Web-Dienst aufzurufen. Sie haben entweder die Möglichkeit die Adresse aus der WSDL-Datei zu beziehen, einen XPath-Ausdruck anzuwenden oder die URL direkt einzugeben.

Beispiel für XPath-Ausdruck: Lesen aus dem Konfigurationsparametern der Applikation *myappl*:

string(\$configuration_myappl/property[@name='webservice.address'])

- **Out-Parameter:** Hier können Sie eine Liste von Parametern als XPath-Ausdruck angeben, die dem Web-Dienst übergeben werden. Die Parameter sind in der WSDL-Datei definiert und wurden bei der Erstellung des Webservice-Client angegeben.
- In-Parameter: Analog zu Out-Parameter, jedoch für Daten, die aus dem Web-Dienst gelesen werden.

Folgende Eigenschaften können für den Knoten Eingangsnachricht bestimmt werden:

- Webservice Operation: Sie können aus der Auswahlliste eine bestehende Webservice Server Operation auswählen, die zuvor (für die Applikation) definiert wurde (siehe Kapitel 6.10.2).
- **Prozess starten:** Wenn diese Checkbox aktiviert ist und dieser Knoten der 1. Schritt in der Prozessdefinition ist, dann wird eine neue Instanz des Prozesses erzeugt. Ist diese Checkbox in diesem Fall nicht aktiviert, kann keine Instanz gestartet werden. Wenn dieser Knoten nicht der 1. Schritt ist, dann darf diese Checkbox nicht aktiviert werden!
- **In-Parameter:** Hier können Sie eine Liste von Parametern als XPath-Ausdruck angeben, die vom Web-Dienst gelesen werden. Die Parameter sind in der WSDL-Datei definiert und wurden bei der Erstellung des Webservice-Server angegeben.

7.2. GRAFISCHE PROZESSDEFINITION

• Korrelation: Auch in diesem Reiter können Sie eine Liste von Parametern als XPath-Ausdruck angeben, die bei der Erstellung des Webservice-Server angegeben wurden. Hier können Mappings definiert werden, mit denen dann eine automatische Zuordnung einer *Eingangsnachricht* zu einer Prozessinstanz gemacht werden können.

Folgende Eigenschaften können für den Knoten Antwort bestimmt werden:

- Webservice Operation: Sie können aus der Auswahlliste eine bestehende Webservice Server Operation auswählen, die zuvor (für die Applikation) definiert wurde (siehe Kapitel 6.10.2).
- **Out-Parameter:** Hier können Sie eine Liste von Parameter als XPath-Ausdruck angeben, die dem Web-Dienst übergeben werden. Die Parameter sind in der WSDL-Datei definiert und wurden bei der Erstellung des Webservice-Server angegeben.

Folgende Funktionen stehen bei allen Web-Dienste-Knoten zur Verfügung:

- **OK:** Durch das Aktivieren der Schaltfläche *Ok* werden Ihre Eingaben übernommen und der Dialog wird geschlossen.
- Abbrechen: Durch das Aktivieren der Schaltfläche *Abbrechen* werden Ihre Eingaben verworfen und der Dialog geschlossen.

Für weitere Informationen über die WDL-Syntax sehen Sie bitte im Kapitel 7.1 nach.

8 Die Suche in @enterprise

8.1 Prozesssuche

Siehe Benutzerhandbuch.

8.2 Dokumentensuche

Siehe Benutzerhandbuch.

8.3 Erweiterte Suche

Siehe Reporting-Handbuch.

8.4 Gespeicherte Anfragen

Siehe Reporting-Handbuch.

9 Administrationsfunktionen

Die Administrationsfunktionen von @enterprise werden im Bereich Admin-Tasks aufgelistet. Dieser Bereich enthält die Ordner Server, Benutzer, Import/Export, Reorganisation und Kommunikation. Wenn Sie den Link Konfigurierte Admin-Tasks aktivieren, werden die von Ihnen hinzugefügten Admin-Tasks im Bearbeitungsbereich dargestellt.

9.1 Server

In diesem Bereich der Admin-Tasks sind alle Administrationsfunktionen, die den Server betreffen, zusammengefasst. Zu diesen gehören:

- Servermonitor
- Serversteuerung
- Ereignisse
- Worklist-Cache
- Klassenpfad
- Timer
- Objekthistorie
- Interface Formulare
- Ausstehende Änderungen
- Event Registrierungen
- Zertifikatsmanagement
- Clustermonitor
- Volltextsuche
- Query-Tool
- Dauerstatistik

9.1.1 Servermonitor

Der Servermonitor bietet Ihnen einen detaillierten Überblick über:

- die Datenbankverbindungen
- die HTTP Threads die aktuelle Auslastung des Servers, d.h. wieviele Threads und Datenbankverbindungen gerade in Verwendung sind
- die Benutzer-Sessions (HTTP-Sessions)
- den Speicherverbrauch
- die Anzahl und Art der aufgetretenen Fehler (Logfile-Auszug)
- den Zeitpunkt, an dem der @enterprise-Server gestartet wurde
- die Anzahl der Anfragen an den Server (seit dem letzten Start) sowohl HTTP als auch RMI
- die Anzahl der abgesetzten DB-Anweisungen
- den Inhalt des Logfiles Je nach eingestelltem Log-Level werden die Zugriffe auf das System mitprotokolliert. Mit der Funktion *Log-Dateien* haben Sie die Möglichkeit, die aktuellsten Log- und Error-Log-Dateien anzusehen. Im Installationshandbuch ist im Kapitel *Configuration* genauer beschrieben, welche Information bei welchem Log-Level in das Log-File geschrieben wird.
- die System-Properties und Thread-Dumps
- zusätzliche Serverinformationen wie Version, Konfiguration, Lizenz
- die Anzahl der aktiven Prozessschritte im System wird nur angezeigt, wenn Worklist-Cache aktiviert bzw. gestartet wurde

Mit Hilfe des Feldes *Aktualisierungsintervall* können Sie angeben, in welchen Zeitabständen die dargestellten Informationen aktualisiert werden sollen.

In der Werkzeugleiste steht Ihnen außerdem die Aktualisierungsfunktion zur Verfügung. Mit dieser können Sie eine Aktualisierung der dargestellten Seite selbst durchführen.

Benutzer-Sessions

199

Mit Hilfe der Administrationsfunktion *Benutzer-Sessions* können Sie feststellen, wann welche Benutzer im System angemeldet waren.

Sobald sich ein Benutzer am System anmeldet, wird die Anzahl der angemeldeten Benutzer mit der Lizenz gegengeprüft (Concurrent-User). Ist das Anmelden möglich, wird eine sogenannte *Benutzer-Session* generiert. Diese Benutzer-Session ist solange gültig, bis sich der Benutzer selbst abmeldet (*Logout*). Wird das Logout nicht durchgeführt ist die Benutzer-Session 24 Stunden gültig und wird danach automatisch beendet. Zur Lizenzüberprüfung

werden allerdings nur Benutzer-Sessions herangezogen, die höchstens seit vier Stunden inaktiv sind.

Sie können sich die Benutzer-Sessions entweder als Benutzerliste oder in Form eines Histogramms ausgeben lassen. Damit die Ausgabe übersichtlich bleibt, können Sie weiters den Zeithorizont einstellen. Als Ergebnis werden Ihnen alle Benutzer-Sessions ausgegeben, welche in diesem Zeithorizont aktiv waren, d.h. wo das Login nach dem Datum (und Uhrzeit) im Datumsfeld **Von**, und das Logout vor dem Datum (und Uhrzeit) im Datumsfeld **Bis**, liegt.

Benutzerliste: Für die Ausgabe einer Benutzerliste, müssen Sie eine weitere Einschränkung vornehmen.

- Angemeldete Benutzer: Liefert Ihnen alle Benutzer-Sessions, die aktuell aktiv sind.
- Alle Benutzer: Liefert Ihnen alle Benutzer-Sessions.
- Ausgewählte Benutzer: Liefert Ihnen alle Benutzer-Sessions des ausgewählten Benutzers.

Mit der Option **letzter Zugriff (Minuten)** können Sie abhängig von der ausgewählten Option bei *Benutzerliste* festlegen, welche Benutzer angezeigt werden, wo der letzte Zugriff vor x Minuten durchgeführt wurde.

Sie können die Benutzerliste mit Hilfe des Zeithorizonts weiter einschränken! Folgende Informationen finden Sie in der Ergebnistabelle:

- Benutzer: Enthält den Vor- und Nachnamen des Benutzers.
- Client IP: Die IP-Adresse des Benutzers.
- initialisiert: Das Initialisierungsdatum der Benutzer-Session, d.h. der Zeitpunkt an dem sich der Benutzer angemeldet hat.
- letzter Zugriff: Der Zeitpunkt an dem der Benutzer das letzte Mal im System aktiv war. Falls gerade ein Thread ausgeführt wird (eine Aktion des Benutzers), kann auf das Datum geklickt werden, wodurch sich ein neues Fenster öffnet, in welchem Sie Detailinformationen zu dieser Aktivität erhalten (u.a. HTTP–Sessions).
- Abmeldung: In dieser Spalte sehen Sie entweder den Abmeldungszeitpunkt oder den Link *Abmelden*, wenn gerade ein Thread ausgeführt wird. Über diesen Link wird die Session beendet und der Benutzer ausgeloggt (auch Clusterweit).

Histogramm: Wenn Sie sich für die Ausgabe des Histogramms entscheiden, wird für das jeweilige Intervall die Anzahl der Benutzer-Sessions ermittelt und jeweils der höchste Wert ausgegeben. Liegen in einem Intervall keine Messwerte vor, d.h. es hat kein Login stattgefunden, wird für das Intervall kein Wert ausgegeben. Im Histogramm steht in jeder Zeile der Startzeitpunkt des Intervalls.

- Stunde: Das Zeitintervall beginnt jeweils mit 0 Minuten.
- Tag: Das Zeitintervall beginnt jeweils um 0 Uhr.
- Woche: Die Woche beginnt jeweils mit dem Starttag des Zeithorizonts um 0 Uhr.

ഄ

(U)

4

3

67

63

9.1.2 Serversteuerung

In der Serversteuerung stehen Ihnen die folgenden Funktionen zur Steuerung und Kontrolle des Servers zur Verfügung:

- Server abschalten: Wenn Sie diese Funktion aktivieren, wird der Server abgeschaltet.
- Server neu starten: Mit dieser Funktion starten Sie den Server neu.
- Dateien im aktuellen Patch anzeigen: Hier können Sie überprüfen, ob Ihr System (inkl. Applikationen) mit den aktuellen Patches übereinstimmt. Ist dies nicht der Fall, wird Ihnen eine Tabelle mit den möglichen Änderungen bzw. den betroffenen Elementen angezeigt.

Nähere Informationen zum Patch-Mechanismus entnehmen sie dem Installationshandbuch.

- Datenbank Upgrade durchführen: Hier können Sie ein ev. notwendiges Update des Datenbank-Schemas durchführen.
- Datenbankverbindungen neu aufbauen: Alle Datenbank-Verbindungen (des aktuellen Knotens) werden als alt markiert. Vor jeder Zuordnung einer Verbindung zu einer Transaktion wird auf das Alter geprüft und anschließend die alten Verbindungen geschlossen und neue aufgebaut. Das Markieren umfasst also alle Verbindungen. Der Neuaufbau wird aber verzögert, je nach Verwendung von Verbindungen.
- Log-Datei initialisieren: Nach Aktivieren dieser Funktion wird die bestehende Log-Datei wegkopiert und eine neue Log-Datei initialisiert.
- Konfiguration neu laden: Diese Funktion hat den Sinn, nach Änderungen in der Datei *avw.conf* bzw. *appl.prop* einer Applikation zu suchen (nicht Konfigurationsparameter-Definition in XML-Datei!), die nicht über das GUI erfolgt sind, und neu einzulesen bzw. geänderte Werte in das Configuration Objekt zu übernehmen. Anschließend wird für jedes Services (und jede Applikation, deren Applikationsklasse das Interface *com.groiss.component.Service* implementiert) die Methode *reconfigure()* aufgerufen. Nähere Informationen dazu finden sie im *Programmierhandbuch* - Kapitel *The Configuration File*.
- Alle Passwörter müssen beim nächsten Login geändert werden.
- **Reporting Schema anzeigen:** Zeigt das Reporting Schema im XML-Format in einem eigenen Fenster an.
- Reporting Schema neu laden: Diese Funktion lädt das Reporting Schema neu.

9.1.3 Ereignisse

Unter dem Menüpunkt *Ereignisse* können Sie nach protokollierten Systemereignissen von @enterprise suchen:

• Von und Bis: Zeitliche Einschränkung der Ereignisse

- Typ: Hier können Sie den Ereignistyp startup bzw. shutdown festlegen
- Suche: Mit dieser Funktion können Sie nach allen protokollierten Systemereignissen suchen, abhängig von den Suchoptionen, die getätigt wurde. Wurden keine Einschränkungen festgelegt, werden alle Systemereignisse in einer Tabelle angezeigt.
- Alle löschen: Diese Funktion erlaubt das Löschen aller protokollierten Systemereignisse, die in der Tabelle angezeigt werden.

9.1.4 Worklist-Cache

In der Worklist-Cache-Administration (siehe Abbildung 9.1) können Sie folgende Tätigkeiten durchführen:

- Cachezustand: Hier wird Ihnen angezeigt, in welchem Zustand der Cache sich aktuell befindet. Sie können diesen Zustand ändern, in dem Sie eine der folgenden Cachezustände auswählen und anschließend die Schaltfläche Ändern aktivieren. Diese Änderungen des Status haben keinerlei Auswirkungen auf den initialen Cachezustand nach dem Systemstart:
 - Ausgeschaltet: Der Cache ist nicht gefüllt, und wird nicht verwendet.
 - Gestartet (aber nicht aktiviert): Der Cache ist gefüllt und wird ganz normal aktualisiert, jedoch werden die Daten aus dem Cache vom System nicht verwendet. Dies ist beispielsweise sinnvoll, wenn Sie prüfen möchten, wie schnell das System mit bzw. ohne Cache arbeitet oder ob der Cache konsistent ist.
 - Aktiviert: Der Cache ist gefüllt und wird vom System verwendet.
- **Cache aktualisieren:** Das Aktualisieren der Cache-Strukturen ist in den folgenden Fällen nötig: Eintrag von neuen Applikationen, Abteilungen oder Rollen und bei Änderungen im Organisationsbaum. Dafür werden folgende Funktionen angeboten:
 - Aktualisieren der Org.Strukturen: Dadurch werden die Organisationsdatenstrukturen des Caches manuell aktualisiert.
 - Aktivitätsinstanzen aktualisieren: Die Arbeitskorbeinträge werden neu von der Datenbank gelesen; dies kann aber durch Ausschalten und wieder Einschalten des Caches bewerkstelligt werden.
 - WLCache reparieren: Falls Inkonsistenzen im Worklist-Cache eines Knoten N aufgrund der Java Garbage Collection auftreten, können diese mit dieser Funktion repariert werden. Durch Angabe eines Zeitintervalls (z.B. Start und Ende der Garbage Collection Phase des inkonsistenten Knotens) und Auswahl eines anderen Knoten M bekommt Knoten N Informationen von M über die geänderten Stepinstances dieses Zeitraums. Knoten N nutzt diese Informationen, um seinen internen Stand auf den Stand der aktuellen Datenbankdaten dieser Stepinstances zu bringen.

Bei beiden Funktionen wird der Zeitpunkt der letzten Ausführung in eckigen Klammern angezeigt.



Der Worklistcache liest beim Einloggen eines Users dessen aktuelle Rollenzuordnungen und Vertretungen. Damit werden Änderungen bei Rollen eines Users nach dem Einloggen wirksam. Änderungen bei Vertretungen (bei denen ja automatisch durch den CurrentSubstitutesTask Fristbeginn oder Fristablauf berücksichtigt werden) spiegeln sich sofort im Cache wieder, ein neuerliches Einloggen des Users oder ein Aktualisieren der Cachestrukturen ist nicht erforderlich.

Das Aktualisieren der Cachestrukturen instantiiert einige der Strukturen doppelt (die alte und die neue Version der Struktur). Es ist mit zusätzlichem Speicherplatzbedarf während der Operation bis zur nächsten Garbage-Collection zu rechnen.

Das API stellt zum Aktualisieren der Cachestrukturen zwei Methoden zur Verfügung:

- com.dec.avw.wlcache.WLCache.getInstance().refresh() Aktualisiert die Cachestrukturen (alle Änderungen an Organisationsdaten); entspricht der manuellen Auslösung der Funktion. Empfohlen wird die Benutzung z.B. nach dem Import eine Menge von Benutzern und deren Rollen.
- com.dec.avw.wlcache.WLCache.getInstance().refreshUser(User u) Geänderte Rollenzuordnungen und Vertretungen eines Benutzers werden im Cache berücksichtigt. Andere Modifikationen der Organisationsdaten (neue Applikationen, Abteilungen, Rollen ...) bleiben unberücksichtigt.
- Cacheüberprüfung für Benutzer: Damit können Sie die Konsistenz des Worklist-Caches für einen bestimmten Benutzer testen. Treten für Einträge im Arbeitskorb, Rollen-Arbeitskorb, Wiedervorlage, Rollen-Wiedervorlage bzw. für Ausstehende Aktionen Inkonsistenzen auf, werden Ihnen diese angezeigt, und Sie können diese einzeln oder gesammelt (für den Benutzer mittels Button *Cache für User aktualisieren*) aktualisieren. Wenn Inkonsistenzen für den jeweiligen Bereich vorhanden sind, werden ein roter Haken und die betroffenen Elemente angezeigt, ansonsten, wenn es keine Inkonsistenzen gibt, ein grüner Haken.

9.1.5 Klassenpfad

Mit dieser Funktion können Sie Klassen neu laden, Klassen auffinden und Shadow Klassen anzeigen. Diese Seite zeigt Ihnen außerdem, welche Verzeichnisse aktuell im Klassenpfad eingebunden sind.

Klassen neu laden

Aus Gründen des Benutzerkomforts für Applikationsprogrammierer wird das Neu laden von Klassen unterstützt. Eine Vorbedingung dafür ist, zwischen Systemklassen und Applikationsklassen (und Ressourcen) zu unterscheiden.

Systemklassen werden vom System-Klassen-Lader geladen und können nicht entladen werden. Diese Klassen befinden Sich im Verzeichnis *lib* der Installation.

Die Applikationsklassen befinden sich in den Verzeichnissen *lib* und *classes* der Applikation. Die Formularklassen, die vom System erzeugt wurden, befinden sich im Verzeichnis *forms* der Installation. Diese Klassen werden vom Applikations–Klassen–Lader geladen und können neu geladen werden, ohne den Server neu starten zu müssen. Dies kann auf zwei Arten erfolgen:

orklist-Cache		
achezustand: Aktiviert euer Zustand: O Ausgeschaltet O Gestartet (aber	nicht aktiviert)	Cache aktualisieren Aktualisieren der Org. Strukturen [2012-07-31 14:51: Aktivitätsinstanzen aktualisieren [2012-07-31 14:51:0 WLCache reparieren
	ndem	
ncheüberprüfung für Benutzer- nutzer-ld: markus ØPersonal Items	Cache überprüfen	Cache für User aktualisieren)
ncheüberprüfung für Benutzer nutzer-ld: markus ØPersonal Items ØRole Items	Cache überprüfen (Items: 48 Items: 5	Cache für User aktualisieren)
ocheüberprüfung für Benutzer nutzer-ld: markus Personal Items Role Items Personal Suspended Items	Cache überprüfen (Items: 48 Items: 5 Items: 1	Cache für User aktualisieren)
Cheüberprüfung für Benutzer nutzer-ld: markus Personal Items Role Items Personal Suspended Items Role Suspended Items	Cache überprüfen (Items: 48 Items: 5 Items: 1 Items: 0	Cache für User aktualisieren)

Abbildung 9.1: Worklist-Cache-Administration

- 1. In der Serverkonfiguration unter Punkt *Tuning* ist die Checkbox *Klassen neu laden* angekreuzt. Soll nun eine Klasse verwendet werden, leert der Applikations–Klassen– Lader seinen Cache, und lädt die Klasse danach automatisch neu.
- 2. Ist die unter Punkt 1 erwähnte Checkbox nicht angekreuzt, können Sie durch das Aktivieren der Schaltfläche *Klassen neu laden* die Klassen manuell neu laden.

Anmerkung: Das Neu laden einer Applikationsklasse ist nicht möglich, wenn die Instanz einer Klasse aus dem Systemklassenpfad auf die Instanz der Applikationsklasse verweist. In diesem Fall muss der Server neu gestartet werden, damit die entsprechende Applikationsklasse neu geladen wird.

Klassen auffinden

Wenn Sie nicht mehr wissen, wo eine Klasse, die Sie verwenden, definiert ist, dann geben Sie den Klassennamen (inkl. Package) in das Feld *Klasse oder Ressource suchen* ein, und aktivieren Sie die Schaltfläche *Ok*. Wenn die Klasse gefunden wird, wird der Pfad bei *URL* ausgegeben.

Shadow Klassen anzeigen

Diese Funktion zeigt Ihnen alle Klassen und Ressourcen, die im Klassenpfad mehr als einmal zu finden sind. Die Anzeige erfolgt in folgender Syntax: relativer Pfad der Ressource Anzahl der gefundenen Ressourcen mit diesem relativen Pfad

Absoluter Pfad der vom System verwendeten Ressource Absoluter Pfad der vom System nicht verwendeten Ressourcen ...

9.1.6 Timer

Der Timer in @enterprise löst zeitgesteuerte Ereignisse aus. Er wird für systeminterne Zwecke verwendet, ist aber auch für applikationsspezifische Tasks verwendbar. Durch Aktivieren des Links **Timer** in der Liste der Administrationstasks gelangen Sie in die Verwaltungsmaske, die eine Liste der eingetragenen Ereignisse darstellt. Es stehen Ihnen die in Kapitel 2.1 beschriebenen Funktionen und zusätzlich die Funktion *Ausführen* zur Verfügung.

Die Objekt-Detailansicht von Timern gliedert sich in die folgenden Bereiche:

- Allgemein
- Zugriff
- Historie

Übersicht über Standard-Timer

In @enterprise gibt es eine Reihe vordefinierter Timer, die hier nun näher beschrieben werden:

- ArchiveTimer: Archiviert abgeschlossene Prozesse. Für nähere Informationen sehen Sie bitte im *Installationshandbuch* nach.
- **BatchManager:** Startet und beendet Batch-Jobs. Wird nur verwendet, wenn Batch-Job Schritte in der Prozessdefinition verwendet werden.
- CalendarReminder: Überprüft auf Kalendereinträge, die ihren Erinnerungszeitpunkt erreicht haben und sendet Hinweis-Emails für diese Einträge. Aktivieren Sie diesen Timer, wenn Sie den Kalender verwenden.
- CleanUpDMS: Löscht leere Ordner und ebenso Rechte, die sich auf DMS-Objekte beziehen, aber nicht mehr erreichbar sind. Aktivieren Sie diesen Timer, wenn Sie das DMS verwenden.
- **ClusterCheck:** Überprüft, ob andere Knoten laufen und teilt eine neuen Cluster Timer zu. Dieser Timer wird nur dann benötigt, wenn der @enterprise Cluster verwendet wird. Für nähere Informationen über @enterprise Cluster sehen Sie bitte im *Installationshandbuch* nach.
- **CurrentSubstitutes:** Überprüft, ob irgendwelche Vertretungseinstellungen aktiviert oder deaktiviert werden müssen aufgrund der eingetragenen Zeitdauer. Dieser Timer wird nur dann benötigt, wenn Benutzer- oder Rollenvertretungen verwendet werden.

- **DeferredUpdate:** Dieser Timer überprüft, ob es aufgeschobene Updates von Masterdaten gibt, für die der Ausführungszeitpunkt erreicht wurde. Wenn dieser Zeitpunkt erreicht wurde, wird ein Update durch diesen Timer ausgelöst. Lassen Sie diesen Timer aktiviert.
- **DeleteUserSessions:** Dieser Timer löscht User-Sessions, die bereits abgelaufen sind. Dabei kann mit dem Parameter *avw.keep.user.sessions* (in Tagen) angegeben werden, wie lange eine User-Session aktiv sein darf, bevor sie der Timer löscht.
- Escalations: Dieser Timer überprüft bei jedem Lauf, ob Timeout-Tasks gestartet oder Eskalationen durchgeführt werden sollen. Für nähere Informationen sehen Sie bitte im Abschnitt 6.2.2 oder 6.5.10 nach. Dieser Timer wird benötigt, wenn Eskalationen oder Timeouts in der Prozessdefinition verwendet werden.
- **Expiration:** In BPEL ist es möglich für eine bestimmte Aktivität (nicht ein bestimmter Task) ein Fertigstellungsdatum zu setzen. Wenn dieser Zeitpunkt überschritten wurde, wird die Aktivität automatisch abgeschlossen (weitergeleitet), sobald dieser Timer läuft.
- HeartBeat: Informiert den Cluster, dass dieser Knoten noch aktiv ist. Dieser Timer wird nur dann benötigt, wenn der @enterprise Cluster verwendet wird. Für nähere Informationen über @enterprise Cluster sehen Sie bitte im *Installationshandbuch* nach.
- IndexRefresh: Aktualisiert den Index für die Volltextsuche in ORACLE. Dieser Timer wird nur dann benötigt, wenn Sie die Volltextsuche unter ORACLE verwenden.
- LDAPDirSyncTask: Synchronisiert mit LDAP Verzeichnis-Servers. Für detaillierte Informationen sehen Sie bitte im Abschnitt 9.5.2 nach. Wird nur dann benötigt, wenn die periodische LDAP Synchronisierung konfiguriert ist.
- Log: Dieser Timer entfernt alle Logeinträge (außgenommen den aktuellsten Eintrag/die letzte Änderung), die älter sind als unter Konfiguration → Lokalisierung → Lösche Objektänderungen nach (Tage): eingetragen. Alternativ ist es möglich im Feld Parameter des Timers einen Wert anzugeben, der den Konfigurationsparameter überschreibt. Wenn der Parameter des Timers einen positiven Wert D enthält, dann werden alle Logeinträge die älter als D Tage sind entfernt. Wenn der Parameter einen Property-String enthält, dann kann für jede Klasse ein positiver Wert (in Tage) festgelegt werden. Der Property-String besteht dabei aus Elementen (durch Zeilenumbruch getrennt) mit folgender Struktur: classname=Dn. Hat Dn den Wert 0, dann werden keine Logeinträge entfernt. Bei Verwendung des Zeichen * als Klassenname wird der Wert Dn für alle Logeinträge aller Klassen, die nicht explizit angeführt werden, angewendet, z.B.

*=30

com.groiss.org.User=0 com.groiss.org.OrgUnit=1000 com.groiss.org.Role=1000 com.groiss.org.UserRole=365

- **MailGetter:** Ladet die Nachrichten herunter und führt die konfigurierte Aktion aus. Für nähere Informationen sehen Sie bitte im Abschnitt 9.5.1 nach. Wird nur benötigt, wenn Mailboxinhalte automatisch verarbeitet werden sollen.
- SeenObjectCleaner: Entfernt alle Seen-Informationen, die nicht mehr benötigt werden. Die Seen-Informationen werden dazu benutzt, um anzuzeigen, ob ein Arbeitskorbeintrag neu ist (=unseen) oder nicht. Lassen Sie diesen Timer aktiviert.
- **Suspension:** Dieser Timer überprüft alle Einträge der Wiedervorlage, ob diese zurück in den Arbeitskorb kommen. Lassen Sie diesen Timer aktiviert.
- WfXMLTask: Sendet WfXML-Nachrichten vom ausgehenden Buffer und erhält Nachrichten von passiven Partnern. Dieser Timer wird nur dann benötigt, wenn WfXML verwendet wird. Für nähere Informationen sehen Sie bitte im Abschnitt 9.5.4 nach.

Registerkarte: Allgemein

•		Timer: BatchManager - @enterprise - Mozilla Firefox – 🗖 🗙		
🛞 localhost:	8180/wf/sen	et.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.timer (📓 🐠 🕶		
Allgemein	Zugriff	Historie		
ld: Klassenn	ame:	BatchManager com.groiss.wf.batch.BatchManager		
Parameter	r:			
Erster Lau	ıf:	09-12-2008 15:57		
Periode:		600 🖉 🕑		
Aktiv:				
Beim Star ausführen	t :			
Auf jedem ausführen:	Knoten			
Thread Id:		BatchManager		
Beschreib	oung:	start batch jobs		
Letzter La passiviert	uf: 2013 um: -	-10-09 12:27:15 - 2013-10-09 12:27:15 (0,016 s) Ausführen Reaktivieren		
Lösche	en	Ok Abbrechen Übernehmen		

Abbildung 9.2: Objekt-Detailansicht: Timer

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Id: eine Id (Kurzbezeichnung) des Eintrags
- Klassenname: der Name der Klasse, die die Aktion enthält. Die Klasse muss das Java-Interface com.groiss.timer.TimerTask implementieren.
- Parameter: ein String-Parameter als Argument für die run Funktion.
- Erster Lauf: Wann soll die Funktion das erste Mal ausgeführt werden?
- Periode: Wiederholungsintervall in Sekunden oder in Form von Cron-Pattern.
- Aktiv: Nur aktive Einträge werden vom Timer berücksichtigt.
- Beim Start ausführen: Soll die Aktion beim Start von @enterprise ausgeführt werden?
- Auf jedem Knoten ausführen: Dieser Timer wird lokal auf jedem Knoten im Cluster ausgeführt.
- Thread Id: Mit Angabe einer Thread Id können die Tasks zu einzelnen Threads gruppiert werden.
- Beschreibung: Freier Text.
- Letzter Lauf: Zeigt an, wann der Timer das letzte Mal ausgeführt wurde (Start- und Endzeit) und wie lange er gebraucht hat (in Sekunden).
- **passiviert um:** Zeitpunkt, wann der Timer temporär inaktiv gesetzt wurde als keine Verbindung zum Server bestand.

Wenn Sie die Schaltfläche Ausführen aktivieren, wird der Timer sofort ausgeführt.

Wenn Sie die Schaltfläche **Reaktivieren** aktivieren, wird der Timer aus dem passiven Zustand gelöst.

Durch einen Klick auf dieses Symbol wird ein Popup-Fenster geöffnet, wo Sie entweder die Periode in Sekunden eingeben oder das Cron-Pattern verwenden können (siehe Abschnitt 9.1.6).

Wenn Sie auf dieses Symbol klicken, werden in einem Popup-Fenster die nächsten 5 Läufe des Timers angezeigt.

Cron-Pattern

ø

æ

Das Cron-Pattern kommt ursprünglich aus der UNIX-Welt und wird für Aufgaben verwendet, die in wiederkehrenden Abständen automatisiert durchgeführt werden sollen. @enterprise verwendet dieses Pattern um Timer nach Belieben starten zu können. Dabei hält sich @enterprise an den V7-Standard von cron. Eine Zeile besteht grundsätzlich aus fünf festgelegten Spalten. Diese enthalten die Zeitangaben (Minuten, Stunden, Tage, Monate, Wochentage), wobei die Spalten durch Leerzeichen getrennt werden. Folgend sind die Einträge für die Zeitangaben dargestellt:

Minuten0-59 und * für alle MinutenStunden0-23 und * für alle StundenTage1-31 und * für jeden TagMonate1-12 und * für jeden MonatWochentage0-7 und * für jeden Wochentag (0 und 7 für Sonntag)

Es gibt zudem erweiterte Funktionen, die hier kurz erwähnt werden:

- Ein Komma, lässt mehrere Zeitangaben zu
- Ein Bindestrich gibt einen Zeitraum an
- Ein Schrägstrich / teilt einen Zeitraum ein

Beispiele:

- Jeden Tag um 09:00 und 15:00 Uhr soll der Timer ausgeführt werden: 0 9,15 * * *
- Am 15. jedes Monats um 09:50 Uhr soll der Timer ausgeführt werden: 50 9 15 * *
- Jeden Samstag um 00:00 Uhr soll der Timer ausgeführt werden: 0 0 * * 6
- Der Timer soll jeden Tag alle 30 Minuten ausgeführt werden: */30 * * * *
- Jeden Tag von 8 bis 20 Uhr wird alle 20 min. der Timer ausgeführt: */20 8-20 * * *

Um nähere Informationen über cron zu erhalten, sehen Sie bitte unter

http://de.wikipedia.org

nach.

9.1.7 Objekthistorie

Diese Funktion ermöglicht es Ihnen, die Veränderungen in den einzelnen Daten anzusehen. Es ist auch möglich sich die Änderungen der einzelnen Objekte, bezogen auf ein bestimmtes Zeitintervall, anzusehen.

Wählen Sie zunächst aus der Liste **Objektklasse** jene Objektklasse aus, deren Änderungen Sie ansehen möchten. Wenn Sie *Alle* auswählen, findet hier keine Einschränkung statt und Sie sehen eine Liste aller Änderungen in allen Objektklassen. Optional können Sie auch einen Zeitraum eingeben, in dem Sie die Änderungen verfolgen wollen. Starten Sie die Suche durch Aktivieren der Schaltfläche **Suche starten**.

In einer Tabelle werden Ihnen nun alle Änderungen der jeweiligen Objektklasse angezeigt. Sie können sich die einzelnen Einträge durch Aktivieren der Funktion **Anzeigen** im Detail ansehen.

٩

۲

67)

ത

9.1.8 Interface Formulare

Hier sehen Sie eine Liste aller Formulare, für die ein Interface definiert wurde (siehe Kapitel 6.6). Durch Anklicken eines Formulars und Ausfüllen desselben können Sie den damit assoziierten Prozess starten (Schaltfläche **Prozess starten**)

9.1.9 Ausstehende Änderungen

Hier werden alle Objekte aufgelistet, für die bevorstehende Änderungen definiert wurden (mit der Funktion *Änderungen durchführen am*, siehe Kapitel 2.2.1).

Sie können sich die Details zum jeweiligen Objekt mit der Funktion **Anzeigen** ansehen. Sie können die ausstehende Änderung entweder direkt in der Detailansicht oder mit der Funktion **Löschen** verwerfen.

9.1.10 Event Registrierungen

Durch das Anklicken des Links *Event Registrierungen* werden Ihnen im Arbeitsbereich alle Event Registrierungen von laufenden Prozessen angezeigt. Die Tabelle enthält die folgenden Informationen:

- Id: Diese Spalte enthält die Id der Prozessinstanz, bei welcher ein Event registriert wurde.
- Registrant: Die Id des Prozesses, der sich für diesen Event registriert hat.
- Event Name: Der Name des Events für den die Registrierung erfolgt ist.
- Kontext: Der Kontext für den Event.
- Handler: Der Event Handler.

In der Werkzeugleiste stehen Ihnen die folgenden Funktionen zur Verfügung:

- Aktualisieren: Wenn Sie diese Funktion aktivieren, wird die Tabelle aktualisiert.
- *Ansehen:* Wenn Sie diese Funktion aktivieren, wird die zum Prozess gehörende Prozesshistorie angezeigt. Diese Funktion wird auch beim Doppelklick auf einen Tabelleneintrag ausgelöst.
- *Weiterleiten:* Nach dem Aktivieren dieser Funktion wird der Prozess weitergeleitet (wie im Arbeitskorb).

9.1.11 Zertifikatsmanagement

Um SSL verwenden zu können brauchen Sie ein Zertifikat. Zertifikate und die dazugehörigen Private und Public Keys werden im sogenannten Keystore gespeichert. Bevor Sie nun ein Zertifikat generieren können, müssen Sie den Keystore konfigurieren und den SSL–Port festlegen. Dies geschieht in der Systemkonfiguration, Schirm *SSL* (siehe *Installationshandbuch - Kapitel Configuration*). Als nächster Schritt muss die Klasse com.groiss.ssl.SSLHttpd in die Liste der geladenen Klassen aufgenommen werden. Dies geschieht ebenfalls in der Systemkonfiguration, Schirm *Klassen* (siehe Installationshandbuch). Nach einem Restart können Sie die Funktionalität des Zertifikatsmanagement nutzen.

Zunächst müssen Sie ein KeyPair bestehend aus Private und Public Key generieren, das gleichzeitig als selbst signiertes Zertifikat gilt. Dieses ist aber nur für Testzwecke zu empfehlen. Für sensible Daten benötigen Sie ein Zertifikat, das von einer Certification Authority (kurz CA) ausgestellt wurde. Um zu einem solchen Zertifikat zu gelangen, müssen Sie allerdings zuerst einen entsprechenden Zertifikations-Request (CR) zu einem KeyPair erzeugen und an diese CA schicken. Diese schickt Ihnen dann ein fertiges Zertifikat, das Sie in @enterprise integrieren und somit eine sichere Kommunikation sicherstellen können.

Selbst signiertes Zertifikat generieren

🥹 @enterprise - Mozilla Firefox				
[http://localhost:8380/wf/servlet.method/com.groiss.ssl.HTMLCertTable.showGenerateSelfCert?&comingFrom=%2Fwf%2Fservlet.method 🏠 🄕				
Selbst signiertes Zertifikat generieren				
Alias Name:	mycert			
Land:	AT			
Firmenname:	Groiss			
Organisationseinheit:	IT			
E-Mail:	office@groiss.com			
Hostname:	cairo.groiss.com			
Firmensitz:	Klagenfurt			
Bundesland:				
Gültigkeitsdauer des Keys in Tagen (Default=90)				
Schlüssellänge in bit:	1024			
	0 2048			
	Ok	Schließen		
		.:		

Abbildung 9.3: Formular für selbstsigniertes Zertifikat

Folgende Informationen sind hier von Ihnen anzugeben:

- Alias Name: Geben Sie hier eine eindeutige Id für diesen Eintrag ein.
- Land: Hier geben Sie den Ländercode des jeweiligen Landes ein. Für Österreich ist dies AT.
- Firmenname: Sie müssen hier die vollständige Bezeichnung Ihrer Firma angeben.
- Organisationseinheit: Hier wird der Name der Abteilung, für die das Zertifikat erstellt werden soll, verlangt.
- E-Mail: Die E-Mail Adresse des verantwortlichen Mitarbeiters ist hier einzutragen.

- Hostname: Hier müssen Sie den Hostnamen jenes Rechners angeben, auf dem der @enterprise Server läuft, für den das Zertifikat ausgestellt werden soll.
- Firmensitz: Der Name der Stadt, in der sich ihre Firma befindet, ist hier einzutragen.
- **Bundesland:** Der Name des Bundeslandes, in der sich ihre Firma befindet, ist hier einzutragen.
- Gültigkeitsdauer in Tagen: Ein Zertifikat hat eine gewisse Gültigkeitsdauer. Defaultwert ist hier 90 Tage.
- Schlüssellänge in bit: Hier können Sie die Länge der zu erzeugenden Schlüssel auswählen.

Nach dem Sie dieses Zertifikat generiert haben, ist SSL prinzipiell betriebsbereit. Es wird jedoch empfohlen, ein offizielles Zertifikat einer CA zu beantragen.

Zertifizierungsrequest erzeugen

Um einen CR zu erzeugen, müssen Sie auf der Seite für das Zertifikatsmanagement das selbst signierte Zertifikat mit einem Doppelklick auswählen. Sie erhalten den Zertifizierungsrequest als Download. Wurde der CR von Ihnen erzeugt, so können Sie damit bei der CA Ihres Vertrauens ein Zertifikat anfordern. Wie Sie das machen, entnehmen Sie bitte der Dokumentation der jeweiligen CA. Testzertifikate erhalten Sie zum Beispiel bei www.secude.com/trustfactory/ oder www.trustcenter.de.

Importieren eines Zertifikats

Wenn Sie von der CA Ihr Zertifikat im PEM-Format oder DER-Format erhalten haben, so müssen Sie dieses in den Keystore importieren. Klicken Sie auf Zertifikat importieren. Folgende Informationen sind hier von Ihnen anzugeben:

- Alias Name: Geben Sie hier die Id des Eintrages an, zu dem der Zertifizierungsrequest erzeugt wurde. Ist das Zertifikat ein fremdes vertrauenswürdiges Zertifikat, so geben Sie eine noch nicht verwendete ID an.
- Kodierung: Zertifikate können im PEM (Base64 codiert) oder DER (binär) Format von der CA zugesendet werden.
- **Zertifikatsfeld:** Liegt Ihr Zertifikat im PEM Format vor, kopieren Sie das Zertifikat in dieses Textfeld.
- Zertifikatsdatei: Wenn Sie ein DER Zertifikat vorliegen haben, so geben Sie hier den Dateipfad Ihres Zertifiakts an.
- Art des Zertifikats: Ist dieses Zertifikat jenes, mit dem sich der @enterprise Server authentifizieren soll, so geben Sie hier Serverzertifikat an. Ist das zu importierende Zertifikat ein Zertifikat einer vertrauenswürdigen Organisation (z.B: Mail-Server, CA für Clientzertifikate) so geben Sie bitte dies an.

🥹 @enterprise - Mozilla Firefox 📃 🗖		
http://localhost:8380/wf/servlet.r	nethod/com.groiss.ssl.HTMLCertTable.showImportCert?&comingFrom=%2Fwf%2Fservlet.method%2Fcc 🏠 🛞	
Zertifikat importieren		
Alias Name: Kodierung des Zertifikates : Datei mit Zertifikat:	Image: Binar (DER, PKCS#7)	
Kodierung des Zertifikates : Fügen Sie Ihr Zertifikat hier	◯ Base64 kodiert	
ein (inkl. der Kopf- und Fusszeile):		
Zertifikatstyp:	 Zertifikat einer vertrauenswürdigen Organisation Serverzertifikat 	
	Ausführen Schließen	

Abbildung 9.4: Zertifikate importieren

Um Zertifikatsketten zu importieren, muss das eigentliche Zertifikat an erster Stelle im Textfeld oder der DER-Datei stehen, gefolgt von dem Zertifikat der CA, die das Zertifikat ausgestellt hat. Diese Kette kann beliebig lang sein, bis hin zu der sogenannten Root-CA!

9.1.12 Clustermonitor

Informationen über die Cluster–Architektur von @enterprise sind im Installationshandbuch zu finden. Dort werden auch die einzelnen Attribute eines Clusters bzw. Knotens erklärt.

9.1.13 Volltextsuche

Wird die Volltextsuche in der Systemkonfiguration (siehe Installationshandbuch) aktiviert, so gilt die Volltextindizierung nur für Objekte, die nach dem Aktivieren erstellt bzw. geändert werden. Um ältere Objekte in die Volltextsuche aufzunehmen, können Sie über diese Funktion alle Formulare und Dokumente, die nach dem unten angegebenen Datum erzeugt oder geändert wurden (bzw. für alle, wenn kein Datum angegeben wird), indizieren.

9.1.14 Query-Tool

Diese HTML–Seite bietet Ihnen die Möglichkeit SQL–Queries auf das Datenbank–Schema von @enterprise abzusetzen. Da das Ausführen dieser Funktion mit enormen Sicherheits-

risiken verbunden sein kann, steht diese Funktion nur unter zwei Voraussetzungen zur Verfügung:

- Der Konfigurationsparameter *database.direct.access* hat den Wert 1. Es gibt kein Interface, mit dem dieser Parameter gesetzt werden kann. Daher müssen Sie diesen Parameter per Hand direkt im Konfigurationsfile eintragen.
- Der Benutzer muss das Recht Objekt ausführen auf alle Objekte besitzen (jeder Benutzer mit der Rolle sys hat dieses Recht). Vertretungen werden hier nicht berücksichtigt.

Geben Sie eine beliebige Query im Textfeld **Anweisung** ein und schicken Sie diese Anweisung durch Aktivieren der Schaltfläche **Anfrage abschicken** ab. Die Anweisung wird direkt auf der Datenbank ausgeführt. Ergebnisse der Anweisung werden Ihnen im unteren Bereich der Seite angezeigt.

9.1.15 Dauerstatistik

Die Dauerstatistik enthält eine Liste aller aufgenommenen Zeitdaten des Prozessminings. Das sammeln der Zeitdaten wird durch die Funktion *Neu* im Toolbar gestartet. Der Zweck des Minings ist, dauerhafter Statistikinformationen von der Prozess-History zu extrahieren. Diese dauerhaften Statistiken werden später dazu genutzt, um Laufzeitstrukturen zu generieren (Zeitgraphen).

🎱 Mozilla Firefox	
🛸 http://localhost:8380/wf/	servlet.method/com.groiss.timemgmt.gui.MiningGui.showMiningMask?node=admin.durst; 🏠 🎯
Prozessmining	
Name	Change-Request
Beschreibung	
Prozesse	boolproc (1)
	bpelEscalations (1)
	bpelSequence (1)
	bpelWait (1)
	branch_proc (1)
	Candidature Process (1)
	Candidature Process (2)
	Change Request (1)
Implementierung	com.groiss.timemgmt.ProbabilisticTimeMgmt 🥪
Prozess-Mining durchfü	hren 🗹
Prozessstart Einschrän	kung
Von	
Bis	
Start	Abbrechen

Abbildung 9.5: Prozessmining

Um Zeitdaten aufzunehmen, muss ein Prozesstyp und optional ein Zeitintervall angegeben werden, in dem die Prozessinstanz gestartet wurde. Sie haben auch die Möglichkeit keine Zeitdaten aufzunehmen, indem Sie die Checkbox *Prozess-Mining durchführen* nicht auswählen. Zusätzlich sollte ein Name angegeben werden, damit diese Zeitaufnahme in der

Dauerstatistik-Tabelle eindeutig zu finden ist. Das Mining erfasst die Dauer jeder Aufgabe (Task) in diesem Prozess. Nachdem das Mining beendet wurde, werden Sie über den Status des Minings informiert.

Nach Fertigstellung des Minings erscheint ein Eintrag mit dem zuvor eingegebenen Namen in der Tabelle *Dauerstatistiken*. Durch Auswahl eines Eintrags können Sie nähere Details darüber erfahren oder diesen Eintrag löschen, falls Sie ihn nicht mehr benötigen. Nach einem Doppelklick auf einen Eintrag öffnet sich die Detailansicht, die Ihnen neben *Namen*, *Beschreibung*, *Implementierung* und *Prozessdefinition* im Reiter *Allgemein* noch 3 weitere Reiter *Dauerhistogramme*, *Abzweigungswahrscheinlichkeiten* und den *Zeitgraph* anzeigt (siehe Abbildung 9.6). Im Feld *Implementierung* können Sie Ihre eigene Klasse angeben, die die Adapterklasse *com.groiss.timemgmt.DefaultTimeManagementImpl* erweitern muss. Nähere Informationen dazu entnehmen Sie bitte der @enterprise API.

Hinweis: Bitte beachten Sie, dass bei Verwendung einer eigenen Implementierungsklasse kein Zeitgraph erzeugt werden kann und dadurch auch die Funktion *Graph neu erzeugen* nicht mehr verfügbar ist!

🥹 Dauerstatistike	n: Change-Reque	st - @enterprise - Mozilla Fir	efox		_ 🗆 🛛
🖹 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.durstat&func=edit&comingFrom=%2Fwf%2f 🏠 🧉					
Allgemein Da	uerhistogramme	Abzweigungswahrscheinlichk	eiten Zeitgraph		
Dauerstatisti	k				
Name: Beschreibung: Implementieru Prozesse	Change-Requ Ing: com.groiss.ti	est nemgmt.ProbabilisticTimeMgm	it Aktiv Chance Request (1)		
			Change Request (1)		
Löschen			Ok	Abbrechen	Übernehmen

Abbildung 9.6: Dauerstatistik

Im Reiter *Dauerhistogramme* wird für jeden Task im Prozess die Wahrscheinlichkeit angezeigt. Der Reiter *Abzweigungswahrscheinlichkeiten* beinhaltet die Wahrscheinlichkeit für jeden Zweig und zeigt, wie oft dieser durchlaufen wurde. In diesem Beispiel besitzt der Prozess ein IF–Konstrukt, das nur im TRUE–Zweig (grüne Linie im Prozesseditor) eine Aktivität hat. Dieser Zweig wurde aber nie durchlaufen, wodurch es zu einer Wahrscheinlichkeit von 0 kommt. Der ELSE–Zweig (rote Linie im Prozesseditor) hat dagegen eine Wahrscheinlichkeit von 1, da dieser immer durchlaufen wurde. Der Reiter *Zeitgraph* beinhaltet eine visuelle Darstellung aller gesammelten Daten, indem man auf einen Link klickt (analog zu Ansicht beschrieben in Abschnitt 7.2.7).

Dauerstatistiken des Minings können erst nach Generierung eines Zeitgraphen zur Laufzeit verwendet werden. Dieser Task ist getrennt vom Mining, weil jeder Prozess nur eine bestimmte Menge von Zeitdaten verwenden kann. Sie können deshalb so viele Minings erstellen wie sie wollen und anschließend entscheiden, welches Sie zur Laufzeit verwenden möchten.

Die dazugehörende Funktion ist in der Detailansicht eines Elements in der Dauerstatistik verfügbar. Sie müssen dafür nur die *Prozessdefinition* aus *inaktiv* nach *aktiv* übernehmen. Daraufhin werden Sie gefragt, ob Sie einen Zeitgraphen erstellen wollen. Nach der Berechnung wird der Status des Generierungsprozesses angezeigt (siehe Abbildung 9.7). Um das Ergebnis zu sichern, steht der Button *Sichern* zur Verfügung. Weiters können Sie zu dem ausgewähltem Prozess in der Liste der aktiven Prozesse einen neuen Zeitgraphen über den Button *Graph neu erzeugen* generieren.

🕙 Mozilla Firefox 📃 🗖 🔁
http://localhost:8380/wf/servlet.method/com.groiss.timemgmt.GraphGenerator2.startGene 🏠 🧕
Zeitgraph wird für Prozess 'Change Request' generiert
itsm_insert_chg 1,0000,
itsm_reviewdesign 1,0000,
if 1,0000,
end 0,7500,
end 0,7500,end node
itsm_implement U,2500,
itsm_test_cngi0,2500,
nsm_reviewimpi 0,2000, nar 0.2500
itsm integrate 0.2500
andioin 0.2500
itsm integrationtest 0,2500.
itsm release 0,2500,
end 0,2500,
end 0,2500,end node
itsm_document 0,2500,
End prob: 1.0
Schließen

Abbildung 9.7: Status des Generierungsprozesses

9.2 Benutzer

In diesem Bereich der Admin-Tasks sind alle Administrationsfunktionen, die die Benutzer betreffen, zusammengefasst. Zu diesen gehören:

- Anmelden de-/aktivieren
- Berechtigungen überprüfen
- Abgelaufene Passwörter

9.2.1 Anmelden de-/aktivieren

Diese Funktion ermöglicht es Ihnen, das Anmelden für normale Benutzer zu deaktivieren bzw. wieder zu aktivieren. Ist das Anmelden deaktiviert, ist es nur einem Systemadministrator, der das Recht *Konfiguration* besitzt, möglich, sich anzumelden.

nmelden de-/aktivieren	
O Anmelden aktiviert	
⊙ Anmelden deaktiviert	
lachricht: Login derzeit nicht möglich wegen Wartungsarbeiten	
Ändern	

Abbildung 9.8: Anmelden deaktivieren

In der Maske stehen Ihnen die folgenden Informationen zur Verfügung:

- Anmelden aktiviert: Ist dieser Radio–Button aktiviert, gibt es keine Einschränkung beim Anmelden.
- Anmelden deaktiviert: Ist dieser Radio-Button aktiviert, können sich nach Neustart des Servers nur Systemadministratoren anmelden. Gleichzeitig werden falls für die Applikation definierte Upgrades von Applikationen durchgeführt. Für nähere Informationen darüber, sehen Sie bitte im API von @enterprise nach (*ApplicationAdapter.getVersion()* und *ApplicationAdapter.upgrade()*).
- Nachricht: Wenn das Anmelden deaktiviert ist, können Sie hier eine Nachricht angeben, die dem Benutzer angezeigt wird, wenn er versucht sich an das System anzumelden.

9.2.2 Berechtigungen überprüfen

Diese HTML–Seite ermöglicht es Ihnen festzustellen, ob für einen Benutzer ein bestimmtes Recht vergeben wurde.

Wählen Sie zunächst einen **Benutzer** aus, dessen Berechtigungen Sie überprüfen wollen. Danach wählen Sie das **Recht** aus, das Sie überprüfen wollen. Sie können optional das Recht auf die *Objektklasse* (und *Objekt*), *Formularklasse* oder auf ein *Dokument* noch weiter einschränken. Nach dem Sie die Schaltfläche **Überprüfen** aktiviert haben, wird die Überprüfung gestartet. Das Ergebnis der Überprüfung wird Ihnen im unteren Bereich der Seite angezeigt.

enutzer:	Maier Franz maier		•	
echt:	Administration		•	
nwenden auf				
) Objektklasse:			-	
Objekt / Berechtig	gungsliste:		Ŧ	
) Formularklasse:			*	
) Dokument:				
Überprüfen	rüfung ergibt: DB: 🎈 erlaubt	/ ACLCache: nicht gefunder	n	
Berechtigungsüberp				
3erechtigungsüberp Berechtigung erh	alten über			
Berechtigungsüberp Berechtigung erh Akteur	alten über Recht	Objektklasse	Objekt	Positiv

Abbildung 9.9: Berechtigungen überprüfen

9.2.3 Abgelaufene Passwörter

Ist in der Passwortpolicy in der Systemkonfiguration (siehe Installationshandbuch) eine maximale Gültigkeitsdauer für Passwörter definiert, liefert Ihnen diese Funktion eine Liste mit allen Benutzern, deren Passwort bereits die maximale Gültigkeitsdauer überschritten hat.

6 Benutzer mit abgelaufenem Passwort gefunden!		
Id	Name	
arb	Andreas Reichenberger'	
bush	Bush	
danet	Danet Admin	
mobi	Mobi	
user_with_all_rights	Benutzer mit allen Rechten	
user_without_rights Benutzer ohne Rechte		

Abbildung 9.10: Abgelaufene Passwörter

9.3 Import/Export

9.3.1 Import/Export im XML Format

Die Import/Export Funktionalität erlaubt es Ihnen, Daten von einem @enterprise Server zu exportieren und auf einem anderen zu importieren. Die Daten werden hierbei in einer XML Datei gespeichert.

Export

Sie können verschiedene Arten von Daten exportieren. Der XML Export zeigt Ihnen eine Auswahl der zur Verfügung stehenden Daten und lässt Sie zusätzliche Optionen wählen. Abbildung 9.11 zeigt die verfügbaren Arten von Daten. Wenn es zusätzlich auswählbare Optionen gibt, wird ein Bereich mit Einstellmöglichkeiten sichtbar (wie in der zuvor genannten Abbildung für Organisationseinheiten zu sehen). Im ersten Teil der Export Maske können Sie eine optionale Beschreibung des Exportes eingeben. Diese Beschreibung wird in die XML Datei integriert und bei einem späteren Import angezeigt.

Export im XML-Format
Beschreibung des Exports: 2012-04-10
O Applikationen (inkl. alle zugehörigen Objekte)
O Prozesse (inkl. alle zugehörigen Objekte)
○ Formulartypen
Organisationseinheiten
O OE-Bäume
O Benutzer
O Berechtigungslisten
O Gespeicherte Anfragen (Auswahl von Anfragen im nächsten Schritt)
O Timer (Auswahl von Timern im nächsten Schritt)
O LDAP Einstellungen
O Mail Einstellungen
O Dashboard (Default Elemente)
O Prozessinstanzen (Auswahl von Prozessdefinitionen im nächsten Schritt)
O DMS Ordner
○ Formulare
O Dauerstatistik
Export

Abbildung 9.11: **Export im XML–Format**

Sie können die folgenden Daten exportieren:

- Applikationen Export jeweils einer Applikation mit allen Daten. Dazu zählen die Daten, die mit Prozessen exportiert werden (siehe Punkt Prozesse) und Daten, die zusätzlich im Applikationskontext definiert wurden: Rechte, Objektklassen, Task-Funktionen, Tasks, Formulare und Rollen. Zugriffsrechte, die für Rollen der Applikation definiert wurden, Prozess-Interfaces (*Organisation* → *Interface*) und Default URLs für Rollen werden weiters auch mitexportiert. Die Zugriffsrechte werden am Zielsystem nur importiert, wenn die betroffenen Objekte bereits vorhanden sind.
- **Prozesse** Export eines oder mehrerer Prozesse mit den darin verwendeten Benutzern, Organisationseinheiten, Tasks, Steps, Formularen, Rollen, Prozess-Interfaces und Zugriffsrechten auf exportierte Objekte, usw. Die Zugriffsrechte werden am Zielsystem nur importiert wenn die benötigten Benutzer, Rollen und Organisationseinheiten bereits existieren.
- **Formulartypen** Export von einem oder mehreren Formulartypen. Es können alle in @enterprise vorhandene Formulare exportiert werden. Bevor ein Formular importiert werden kann, muss sichergestellt sein, dass das entsprechende Formular-Template am Zielsystem vorhanden ist. Weiters ist beim Export und anschließendem Import zu beachten, dass Formulare auf andere Formulare referenzieren können (z.B. es gibt ein Kundenformular und dieses referenziert auf ein Länderformular). Um die Referenzen zwischen den Formularen auf dem Zielsystem aufrecht zu erhalten, können die beteiligten Formulare beim Export ausgewählt und in **einem** Form-Cluster exportiert werden. Wenn die beteiligten Formulare einzeln (d.h. jedes Formular wird in einem eigenen Form-Cluster exportiert) exportiert werden und Referenzen bestehen, werden alle beteiligten Formulartypen automatisch mitexportiert.
- Organisationseinheiten Export aller Organisationseinheiten.
- **OE-Bäume** Export von OE-Bäume und dessen Organisationseinheiten.
- **Benutzer** Export aller Benutzer. Optional können Rollenzuordnungen, Rechtezuordnungen, Benutzereinstellungen und die Dashboard Elemente der Benutzer mit exportiert werden.

Achtung: Benutzereinstellungen enthalten eine Startseite. Sowohl in dieser Startseite als auch in den Dashboard Elementen können OIDs von beliebigen Objekten vorkommen. Diese werden beim Import nicht geändert, wodurch Startseiten und Dashboard Elemente möglicherweise am Importserver nicht mehr funktionieren könnten.

- Berechtigungslisten Export aller Berechtigungslisten im System.
- **Gespeicherte Anfragen** Export einer oder mehrerer gespeicherter Anfragen. Die zu exportierenden Objekte können in einem zweiten Schritt ausgewählt werden. Optional können Zugriffsrechte auf die gespeicherten Anfragen mit exportiert werden. Referenzierte Objekte werden nicht mit exportiert (in gespeicherten Anfragen können unterschiedlichste Objekte vorkommen z.B. Prozessdefinitionen, Tasks, etc.). Am Zielsystem müssen diese Objekte vorhanden sein, damit die gespeicherten Anfragen importiert werden.
- **Timer** Export eines oder mehrerer Timer. Die Timer können in einem zweiten Schritt ausgewählt werden.
- **LDAP Einstellungen** Export aller LDAP Einträge. Exportiert die LDAP Einträge, welche unter *Kommunikation* \rightarrow *LDAP* definiert sind.
- **Mail Einstellungen** Export aller Mailboxen. Exportiert die Mailboxen, welche unter *Kommunikation* \rightarrow *Mailboxen* eingetragen sind.
- **Dashboard (Default Elemente)** Export von Default Dashboard Elementen. Administratoren können Default Dashboard Elemente erstellen, welche mit diesem Export exportiert werden können. Dashboard Elemente von Benutzern werden hierbei nicht mit exportiert.
- **Prozessinstanzen** Export von Prozessinstanzen (Laufzeitdaten) eines oder mehrerer Prozesse. Hierbei werden alle Prozessinstanz-Schritte, Formularinstanzen, Adhoc Schritte, etc. exportiert. Zugriffsrechte auf exportierte Objekte können optional mit exportiert werden. Die Menge der exportierten Instanzen kann durch eine Einschränkung der Prozessstart-Zeit begrenzt werden (z.B. nur Instanzen exportieren, die nach einem bestimmten Datum gestartet wurden). Die Prozesse, von denen Instanzen exportiert werden sollen, können in einem zweiten Schritt ausgewählt werden.

Stammdaten (z.B. Prozessdefinitionen, Benutzer, Rollen) werden beim Exportieren von Prozessinstanzen nicht mit exportiert. Prozessinstanzen werden am Zielsystem nur importiert, wenn die entsprechenden Objekte vorhanden sind. Möchten Sie also Prozessinstanzen von einem Server auf einen anderen bringen, stellen Sie zuvor sicher, dass die Stammdaten am Zielsystem verfügbar sind.

DMS Ordner Export eines DMS Ordners mit Inhalt (Laufzeitdaten). Dokumente, Formulare, Notizen, Weblinks und Unterordner werden mit exportiert (rekursiv). Verknüpfungen zu anderen DMS Objekten werden nicht exportiert. Zugriffsrechte auf die exportierten Daten können optional mit exportiert werden. Darin vorkommende Benutzer, Rollen und Organisationseinheiten werden allerdings nicht exportiert - diese müssen am Zielsystem verfügbar sein, damit die Zugriffsrechte auch dort angelegt werden.

Sie können den "Allgemein" Ordner oder beliebige Benutzer-Ordner exportieren (oder Unterordner dieser Ordner). Wenn Sie einen Benutzer-Ordner exportieren möchten, wählen Sie bitte zuerst den Benutzer aus und danach den entsprechenden Ordner.

- **Formulare** Export von jeweils einer Formularinstanz. Es können alle in @enterprise vorhandenen Formularinstanzen exportiert werden. Bevor eine Formularinstanz importiert werden kann, muss sichergestellt sein, dass die Formularklasse und alle referenzierten Objekte (z.B. Prozessdefinition, da Formularinstanz Prozessformular ist) am Zielsystem vorhanden sind, d.h. die entsprechende Applikation mitsamt Formulartyp muss vorhanden sein.
- **Dauerstatistik** Export von jeweils einer Dauerstatistik. Es können alle in @enterprise vorhandenen Dauerstatistiken exportiert werden. Bevor eine Dauerstatistik am Zielsystem importiert werden kann, muss sichergestellt sein, dass alle referenzierten Objekte am Zielsystem vorhanden sind, d.h. die entsprechende Prozessdefinition muss

vorhanden sein. Am Zielsystem muss der Zeitgraph neu generiert werden (siehe Abschnitt 9.1.15).

Der Server exportiert die XML Datei zuerst in sein temporäres Verzeichnis. Nachdem der Export abgeschlossen ist, werden Sie gefragt, ob Sie die Datei herunterladen möchten. Speichern Sie die Datei, um sie später am Zielsystem importieren zu können.

Import

Der Import passiert in drei Schritten:

- 1. Zuerst wird die XML Datei auf den Server hochgeladen.
- 2. Danach zeigt der Browser Informationen über den Inhalt der Datei an.
- Dann wird die Datei importiert und Informationen über die importierten Objekte erscheinen im Browser.

Importiert wird dabei immer nach dem Prinzip, dass bei Auftreten eines Fehlers der Import abgebrochen und der Fehler angezeigt wird. Die bereits importierten Objekte werden nicht wieder gelöscht!

Beim Import von Export-Files aus früheren @enterprise Versionen (z.B. @ep7.0) wird der Benutzer zuerst darauf hingewiesen, dass der Export mit einer früheren @enterprise Version durchgeführt wurde und weiters aufgefordert, dass eine Applikation für die Standardobjekte ausgewählt werden muss, damit die Zuordnung von Applikationsobjekten (z.B. Prozesse) korrekt durchgeführt werden kann. Beim Import von z.B. Mail-Einstellungen wird diese Auswahl ignoriert, da sie in diesem Fall nicht relevant ist.

Import/Export Abhängigkeiten

Wenn Sie Daten von einem Server auf einen anderen Server bringen möchten, ist es nötig, die Dateien in der richtigen Reihenfolge zu importieren. Beim Export ist die Reihenfolge hingegen egal. Laufzeitdaten setzen voraus, dass benötigte Stammdaten bereits am Importsystem vorhanden sind, andernfalls können die Laufzeitdaten nicht importiert werden. Das gleiche gilt auch für bestimmte Objekte wie z.B. Zugriffsrechte und gespeicherte Anfragen. Es wird empfohlen, Imports in folgender Reihenfolge durchzuführen:

- 1. Benutzer (ohne Rechte)
- 2. Organisationseinheiten
- 3. Applikationen, Prozesse, Benutzer (inkl. Rechte), Berechtigungslisten
- 4. Prozessinstanzen, DMS Ordner, gespeicherte Anfragen, Timer, LDAP Einstellungen, Mail Einstellungen, Dashboard Elemente

9.3.2 Prozesse archivieren

Wenn Sie in der Liste der Administrationstasks die Funktion *Prozesse archivieren* aktivieren, dann wird eine neue HTML-Seite angezeigt, die es Ihnen ermöglicht, beendete Prozessinstanzen zu löschen (siehe Abbildung 9.12). Die Prozessinstanzen werden aus @enterprise-Tabellen gelöscht. Ist eine Archivierungsklasse eingetragen, werden die Prozesse vorher in ein Archivsystem übernommen.

Um beendete Prozesse zu löschen, führen Sie folgende Schritte durch:

- 1. Wählen Sie den zu löschenden Prozesstyp oder eine Applikation aus (alle Prozesse der Applikation).
- 2. Wählen Sie ein Datum, bis zu welchem sämtliche Instanzen des ausgewählten Typs gelöscht werden sollen. Prozesse, die am Tag des eingegebenen Tags beendet wurden, werden nicht gelöscht.
- 3. Wenn noch laufende Prozessinstanzen ebenfalls gelöscht werden sollen, markieren Sie die entsprechende Checkbox.
- 4. Archivieren Sie die Prozesse, indem Sie auf die Schaltfläche Archivieren klicken.

Prozessinstanzen archivieren			
Applikation:			
Prozess:	Incident Management (1)	✓	
Beendet vor:	01-01-2011 00:00 🛗		
Auch laufende Prozesse archivieren:			
Archivierungsschnittstelle: com.groiss.itsm.ITSMArchiver			
Archivieren			

Abbildung 9.12: Prozessinstanzen archivieren

9.3.3 Applikation installieren

Wenn ein .*zip* oder .*jar* File existiert, das einen Applikationsbaum enthält (siehe Programmierhandbuch unter *Structure of Applications in @enterprise*), kann die Applikation ganz einfach geladen werden. Geben Sie dazu im Feld *Filename* den Namen des .*zip* oder .*jar* Files ein. Dann geben sie im Feld *Zielverzeichnis* das Verzeichnis ein, in dem die Applikation installiert werden soll, und aktivieren anschließend die Schaltfläche *Installieren*. Damit ist die Installation der Applikation abgeschlossen.

9.3.4 Dateiimport

Der Dateiimport ermöglicht den Import von Daten aus Textdateien ohne aufwendige Programmierung. Um dies durchzuführen, müssen folgende Einstellungen getätigt werden:

• Import Definition: Um die Funktion Dateiimport nutzen zu können, muss eine Import Definition (import.xml) erstellt werden, die im classes-Vezeichnis von @enterprise oder in einem Applikationsverzeichnis gespeichert werden muss (siehe Programmierhandbuch - Kapitel Organization of Files). Nachstehend ein Beispiel für eine Import Definition: <?xml version="1.0" encoding="iso-8859-15" standalone="yes"?> <importDeclarations>

```
<import name="resources">
<targetClass>com.groiss.calendar.pers.Resource</targetClass>
<columns>
<column name="name"/>
<column name="description"/>
</columns>
<keyField>name</keyField>
<delimiter>;</delimiter>
</import>
```

</importDeclarations>

Die einzelnen Schlüsselwörter der Import Definition sind im nachfolgenden Unterkapitel *Schlüsselwörter der Import Definition* beschrieben.

- *Datei:* Hier können Sie den Ort auswählen, von wo die Datei hochgeladen werden soll:
 - Hochladen: Wenn diese Funktion gewählt wurde, haben Sie die Möglichkeit einen Pfad anzugeben, welche Datei hochgeladen werden soll
 - lokal: Diese Funktion erlaubt es, alle Dateien, die im @enterprise-Verzeichnis liegen, hochzuladen. Die Wurzel ist dabei das @enterprise-Verzeichnis selbst.
 - Klassenpfad: Bei Auswahl dieser Funktion können nur Dateien hochgeladen werden, die im Klassenpfad vorhanden sind
 - Gemäß Definition: Es wird die Datei hochgeladen, die in der Import Definition (Datei import.xml) mitsamt Pfadangabe eingetragen ist
- *Mode:* In dieser Dropdown-Liste können Sie zwischen 3 verschiedenen Upload-Modi auswählen:
 - Nur Parsen: Die Datei wird nur nur geparst und keine Objekte im System angelegt
 - Ohne Datenbankoperation: Die Datei wird mit bestehenden Objekten im System verglichen
 - Import: Die Datei wird hochgeladen und Objekte werden im System (Datenbank) erzeugt
- Laden: Mit Hilfe dieser Funktion wird die ausgewählte Datei hochgeladen

Schlüsselwörter der Import Definition

• <import>: Die Importbeschreibung, die dem Format <*import name*="name"> entsprechen muss. Dabei können folgende Attribute definiert werden:

9.3. IMPORT/EXPORT

Dateiimport	
Import Definition	Ressourcen
Datei	Klassenpfad 💌
Mode	Ohne Datenbankoperationen 💌
Laden	



- ignoreHeader: Wenn auf true gesetzt, wird 1. Zeile ignoriert.
- useOrgData: Wenn auf *true* gesetzt, werden die OrgData-Methode von @enterprise verwendet anstatt der Store-Methoden.
- <targetClass>: Symbolisiert den zu importierenden Typ (=Zielklasse).
- <targetCondition>: Einschränkung der Elemente der *targetClass*. Nur diese Elemente werden mit den importierten verglichen, eventuell nicht mehr vorhandene gelöscht.
- <keyField>: Feld der Zielklasse, das den Schlüssel enthält.
- <importHandler>: Wenn kein *keyField* angegeben wurde, muss ein Import-Handler angegeben werden, der das Interface *com.groiss.fileimport.ImportHandler* implementiert.
- <constants>: Beinhaltet eine Menge von Konstanten (<*constant name="name" value="wert"/>*). Diese werden zu der Menge der Werte jeder Zeile hinzugefügt.
- *<extensionClass>:* Name der Klasse für zusätzliche Daten von Stammdatenobjekten (Benutzer, OEs)
- <delimiter>: Feldtrenner, z.B.;
- <escapeMode>: Gibt an, was zu tun ist, wenn ein zu escapendes Zeichen auftritt. Backslash oder Duplicate, z.B. Sonderzeichen ist quote: dÁrtangnon bzw. d"artangnon
- <commentchar>: Jene Zeilen werden ignoriert, die mit diesem Zeichen beginnen.
- <charset>: Alle gültigen Java Charsets (default: StringUtil.getCharset());
- <file>: Pfad zur Datei.
- <columns>: Kann eine Menge von Spalten beinhalten, die zum Importieren sind:
 <column name= "name" startcol= "1" endcol= "10" length= "100" [format= "dateformat"]
 [mapping="mappingName"] />
- <dateformats>: Eine Menge von Datumsformate kann angegeben werden:
 <dateformat name="name" timezone="timezone" locale="locale"/> Beispiel:

<dateformats> <dateformat name="date">ddMMyyyy</dateformat> </dateformats>

<mappings>: Definition von Mappings möglich im Format <mappings name="name"><mapping*><keys><key>M</key></keys><value>1</value></mapping></mappings>.
 Beispiel:

```
<mappings>
<mapping name="lang">
<keys><key>DE</key></keys>
<value>de_DE</value>
</mapping>
</mappings>
```

9.4 Reorganisation

In diesem Bereich der Admin-Tasks sind Administrationsfunktionen der Reorganisation zusammengefasst. Zu diesen gehören:

- Rollenzuordnungen ändern
- Prozessinstanzen analysieren
- OE-Historie

9.4.1 Rollenzuordnungen verändern

Wählen Sie im Feld *Von alter Organisationseinheit* jene Organisationseinheit aus, deren Rollen Sie an die Organisationseinheit im Feld *Zu neuer Organisationseinheit* verschieben oder kopieren möchten (siehe Abbildung 9.14). Nachdem Sie die Schaltfläche **Weiter** aktiviert haben, wird Ihnen eine weitere HTML-Seite angezeigt (siehe Abbildung 9.15), in der Sie nun explizit die Rollenzuordnungen von bestimmten Benutzern ändern können.

Rollenzuordnungen ändern		
Von alter Organisationseinheit:	Groiss Informatics 🔹 👻	
Zu neuer Organisationseinheit:	GI	
Weiter		

Abbildung 9.14: Rollenzuordnungen verändern (1)

Sie können nun festlegen, ob:

• die Rolle bei der alten Organisationseinheit bleibt oder

9.4. REORGANISATION

Rollenzuordnungen ändern					
Benutzer	Rolle	Rolle belassen	Auf die neue OE verlegen	Kopieren	
Bush	Home	0	0	0	^
Roland Eisenberg	Home	0	0	0	
Hugenotte	Home	0	0	0	
Benutzer mit allen Rechten	Home	0	0	0	
Benutzer mit allen Rechten	OE	\bigcirc	0	\bigcirc	
sepp	Home	0	0	0	
Benutzer mit allen Rechten	hugo	\bigcirc	0	\bigcirc	≡
Jean Pierre	Home	0	0	0	
Markus Irrasch	Home	\bigcirc	0	\bigcirc	
Markus Irrasch	hugo	0	0	0	
Markus Irrasch	Zuweiser	\bigcirc	0	\bigcirc	
Markus Irrasch	Supporter	0	0	0	
Markus Irrasch	Release Manager	\bigcirc	0	\bigcirc	
Markus Irrasch	Tester	0	0	0	
Roland Eisenberg	Tester	\bigcirc	0	\bigcirc	
Roland Eisenberg	Zuweiser	0	0	0	
Roland Eisenberg	Release Manager	\bigcirc	0	\bigcirc	
Roland Eisenberg	Supporter	0	0	0	
Markus Irrasch	testrolle	\bigcirc	0	\bigcirc	
Mobi	hugo	0	0	0	
Mobi	Abteilungsleiter	\bigcirc	0	\bigcirc	
Mobi	testrolle	0	0	0	
Persian مرائب سازماني	Home	0	0	0	
Giovanni	Home	0	0	\circ	~
Anzahl der Einträge: 46 Übernehmen Zurüc	*				

Abbildung 9.15: Rollenzuordnungen verändern (2)

- die Rolle zur neuen Organisationseinheit verschoben werden soll oder
- die Rolle durch Kopieren auch der neuen Organisationseinheit zugeordnet werden soll.

Durch Aktivieren der Schaltfläche Durchführen wird die Rollenzuordnung verändert.

9.4.2 Prozessinstanzen analysieren

Hier wird Ihnen eine Liste aller Prozessinstanzen präsentiert, die derzeit keinen gültigen Akteur haben. Entweder wurde der Akteur oder die Organisationseinheit deaktiviert oder es gibt keine passende Rollenzuordnung usw.

Beispiel: Ein existierender Prozess wird weitergeleitet, der jedoch keinen gültigen Folgeakteur besitzt. Die Prozessinstanz taucht in der Tabelle mit dem Problemstatus *Weiterleiten nicht abgeschlossen* auf. Nun gibt es die Möglichkeit die Detailansicht der Instanz zu öffnen (Klick auf Prozessinstanz-ID) und den Prozess einem anderen Akteur zuzuweisen, der den Prozess korrekt abschließen kann. Dazu muss in der Prozesshistorie auf den letzten aktiven Akteur geklickt und über die Maske *Akteur ändern* die Prozessinstanz einem anderen Akteur zugewiesen werden. Die 3 Fragezeichen (???) in der Historie symbolisieren, dass die Instanz noch keinem Akteur zugewiesen werden konnte; sollte auch nicht über die Funktion *Prozessinstanzen analysieren* durchgeführt werden, da sonst eine weitere Prozessinstanz erzeugt wird.

9.4.3 OE–Historie

Hier wird Ihnen die Möglichkeit angeboten, die Änderungsgeschichte von OEs manuell festzuhalten. Dies kann im Rahmen von Umstrukturierungen der Organisationsstruktur erforderlich sein, wenn Sie wissen wollen, welche OE im Rahmen der Umstrukturierung aus welcher anderen OE hervorgegangen ist.

9.5 Kommunikation

In diesem Bereich der Admin-Tasks sind alle Administrationsfunktionen, die mit der Kommunikation in Zusammenhang stehen, zusammengefasst. Zu diesen gehören:

- Mailboxen
- LDAP
- BatchJobs
- WfXML

9.5.1 Mailboxen

Hier können Sie Mailboxen definieren. Es handelt sich dabei um Mailboxen auf einem Mail-Server auf die mit dem Protokoll IMAP4 oder POP3 zugegriffen wird. Eingehende Mail in solchen Mailboxen können auf unterschiedlichste Art verarbeitet werden. Häufig werden Workflows mit Daten aus der E-Mail gestartet.

Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Server: Mail-Server
- Benutzer: Benutzer der Mailbox
- Passwort: Das Passwort für die Mailbox
- Ordner: Ordner, auf den zugegriffen wird
- Mail Protokoll: IMAP4 oder POP3
- Art der Kommunikation:
 - plain: Die Kommunikation ist unverschlüsselt.
 - encrypted: Die Kommunikation ist verschlüsselt.

9.5. KOMMUNIKATION

🥹 Mailboxen: imiptest	(wm.groiss.com) - @enterprise - Mozilla Firefox		
🔝 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.mailbox&func=edi 🏫 🥪			
Allgemein Mailbox	anzeigen		
ld:	Mailbox		
Server:	wm.groiss.com		
Benutzer:	imiptest		
Passwort:	•••••		
Ordner:			
Mail Protokoll:			
Art der Kommunikati	on: unverschlüsselt 💌		
Aktion			
 Als WfXML Prozess sta Prozess: 	Nachricht interpretieren arten		
 Andere Akt Javaklasse: 	ion com.groiss.calendar.iCalendar.IMIPHandler		
Durch Timer ausführe Beschreibung:	n: ▼ iMIP-mailbox		
	Mails downloaden		
Löschen	Ok Abbrechen Übernehmen		

Abbildung 9.16: **Objekt-Detailansicht: Mailbox**

- trusted: Die Kommunikation ist SSL verschlüsselt mit MailServerauthentifizierung. Hierzu muss das Zertifikat des Mail Server als vertrauenswürdiges Zertifikat in den Keystore importiert werden. (Siehe Kapitel 9.1.11)
- Aktion: Eine der folgenden Aktionen wird auf die eingehenden Mails angewendet:
 - Als WfXML Nachricht interpretieren
 - Prozess starten
 - Andere Aktion: Eine Java Klasse, die das Interface com.groiss.mail.MailHandler implementiert, siehe entsprechendes Beispiel im Programmierhandbuch.
- **Durch Timer ausführen:** MailTimer liest diese Mailbox und führt Aktion automatisch durch.
- Beschreibung: Freier Text.

9.5. KOMMUNIKATION

Wenn Sie die Schaltfläche **Mails downloaden** aktivieren, wird die definierte Aktion auf alle eingehenden E-Mails sofort ausgeführt.

Registerkarte: Mailbox anzeigen

Beim Wechseln auf diese Registerkarte wird der Inhalt der Mailbox angezeigt.

9.5.2 LDAP

Hier können Sie LDAP (Lightweigth Directory Access Protocol) Server definieren. Diese dienen dem Abgleich der @enterprise Organisationsdaten mit bestehenden Verzeichnisdiensten. Neben einem vordefinierten Abbildungsmechanismus können auch eigene Synchronisationssemantiken für spezifische LDAP-Schemata realisiert werden. Näheres dazu findet man im Programmierhandbuch.

Registerkarte: Allgemein

Informationen der Registerkarte Allgemein (Mussfelder sind fett dargestellt):

- Name: Name des Servers.
- Server: Hostname
- Port: Port (default Port 389 wenn nicht eingetragen)
- Richtung: Richtung der Synchronisation: entweder
 - nach LDAP oder
 - nach @enterprise
- Suchpfad: LDAP Suchstring, z.B. *dc=groiss,dc=com*
- Benutzer: LDAP-Benutzer, z.B. cn=LDAPAdmin,dc=groiss,dc=com
- Passwort: Passwort des Benutzers
- Filter: LDAP Filter: erlaubt die Einschränkung auf bestimmte LDAP-Einträge z.B.: (objectClass=*)
- Klassenname: durch Angabe einer Klasse, die com.groiss.ldap.DirectorySyncer implementiert, können eigene Abbildungen proprietärer LDAP-Schemata realisiert werden.
- Beschreibung: Freier Text. In diesem Text können Sie zudem einen Parameter *_page-size* angeben, falls es beim Auslesen eines Active Directory Probleme mit der Größe des Resultats gibt (z.B. Suchpfad nicht tief genug). Mit diesem Parameter wird das Ergebnis gepaged ausgelesen, abhängig von der Anzahl, z.B. _pagesize=500.
- Durch Timer ausführen: Wenn diese Checkbox aktiviert ist, führt der LDAPDirSyncTask-Timer die Aktion automatisch durch.

9.5. KOMMUNIKATION

🕹 LDAP: com.gr	oiss.Idap.DirectoryServer@12 - @enterprise - Mozilla Firefox 📃 🗖 🔀			
ttp://localhos	👔 http://localhost:8380/wf/servlet.method/com.groiss.storegui.TabbedWindow.showDialog?node=admin.ldap&func=ec 🏫 🥪			
Allgemein	nhalt auflisten			
Name:	groiss			
Server:	10.205.112.33			
Port:	389			
Richtung:	O nach LDAP			
	💿 nach @enterprise			
Suchpfad:	dc=groiss,dc=com			
Benutzer:	cn=Manager,dc=groiss,dc=com			
Passwort:	•••••			
Filter:	objectClass=*			
Klassenname:	\odot			
Beschreibung:				
Durch Timer a	ısführen:			
Ora einheiten:				
OE-Bäume:				
Rechte:				
Rollen:				
Benutzer:				
	Synchronisieren			
Löschen	Ok Abbrechen Übernehmen			

Abbildung 9.	17: Ob	jekt-Detailan	sicht:	LDAP
--------------	---------------	---------------	--------	------

- Organisationseinheiten: Ist diese Checkbox angekreuzt, wird diese Objektklasse bei der Synchronisation berücksichtigt.
- OE–Baum: Ist diese Checkbox angekreuzt, wird diese Objektklasse bei der Synchronisation berücksichtigt.
- Rechte: Ist diese Checkbox angekreuzt, wird diese Objektklasse bei der Synchronisation berücksichtigt.
- Rollen: Ist diese Checkbox angekreuzt, wird diese Objektklasse bei der Synchronisation berücksichtigt.

• Benutzer: Ist diese Checkbox angekreuzt, wird diese Objektklasse bei der Synchronisation berücksichtigt.

Wenn Sie die Schaltfläche **Synchronisieren** aktivieren, wird die Synchronisation in die angegebene Richtung durchgeführt.

Registerkarte: Inhalt auflisten

Wenn Sie auf diese Registerkarte wechseln, wird der Inhalt des LDAP-Servers aufgelistet.

9.5.3 BatchJobs

Mit dieser Funktion ist es möglich nach BatchJobs zu suchen. Als Suchkriterien können sowohl die Prozess–Id als auch der Status eines BatchJobs verwendet werden. Weiters kann nach einer gewissen Zeitspanne eingegrenzt werden, wann der Batchschritt gestartet wurde. Nach Aktivieren der Schaltfläche *Suchen* wird eine Tabelle mit allen Batchschritten aufgrund der angegebenen Bedingungen angezeigt. Durch einen Doppelklick auf einen Eintrag können die Details zu diesem Batchschritt angezeigt und anschließend verändert werden. Durch das Aktivieren der Schaltfläche *Abbrechen und Zurückgehen* wird der Batchschritt abgebrochen und zum letzten inaktiven Schritt im Prozess zurückgegangen.

Nähere Informationen zu BatchJobs finden Sie im *Programmierhandbuch* im Abschnitt *Batch Processing*.

9.5.4 WfXML

WfXML ist ein Protokoll für Workflow Engines, um diese miteinander zu verbinden. WfXML 2.0 ist dabei eine aktualisierte Version dieses Protokolls, aufbauend auf das ASAP-Protokoll (Asynchronous Service Access), das wiederum auf das SOA-Protokoll aufbaut (Simple Object Access).

@enterprise bietet eine Implementierung dieses Standards an. Dieser umfasst das Empfangen von WfXML-Nachrichten zum Starten von Prozessen, das Setzen und Auslesen des Prozessstatus und dem Senden jeglicher Art von Nachrichten.

Nähere Informationen zu WfXML und dessen Verwendung in @enterprise finden Sie im *Programmierhandbuch*.

9.5.5 Web Dienste

Die Funktion *Web Dienste* an dieser Stelle umfasst die Funktion *Lokale Dienste*, die eine Tabelle aller gefundenen Web Dienste in der @enterprise-Umgebung anzeigt. Hier können Sie neue Web Dienste hinzufügen, bestehende löschen oder aktiveren/deaktivieren (Deployment). Die Erstellung der Objekte *Web Dienste Client/Server* ist pro Applikation möglich und wird bei jeder Applikation mit entsprechenden Funktionen angeboten (siehe Kapitel 6.10).

Nähere Informationen über Web Dienste finden Sie im Programmierhandbuch.

10 Konfiguration

Dieses Kapitel beschreibt die Konfiguration des @enterprise-Servers. Für nähere Informationen über die Menüpunkte

- Lizenz
- HTTP-Server
- Datenbank
- Verzeichnisse
- Logging
- Klassen
- Lokalisierung
- Kommunikation
- Cluster
- Workflow
- DMS
- Suche
- Tuning
- Sicherheit
- Passwort Policy
- Kalender
- Zeitverwaltung
- Administratorpasswort ändern
- DB–Schema initialisieren

sehen Sie bitte im Installationshandbuch - Kapitel Configuration nach.

11 Dashboard

In der Systemadministration von @enterprise besteht die Möglichkeit sich ein Dashboard anzulegen, das auf die Bedürfnisse des Systemadministrators bzw. den Benutzer abgestimmt werden kann. Nach dem Anklicken des Links *Dashboard* unter *Admin-Tasks* kann in die leere Seite geklickt werden, wodurch ein Popup–Fenster angezeigt wird, in dem folgende Funktionen zur Verfügung stehen:

11.1 Neu

Durch das Aktivieren der Schaltfläche *Neu* wird eine HTML–Seite angezeigt, mit der Sie neue Fenster zum Dashboard hinzufügen können. Dazu stehen Ihnen mehrere Möglichkeiten zur Verfügung:

- URL: Geben Sie hier die URL einer HTML–Seite ein, die Sie zukünftig in einem Fenster auf Ihrem Dashboard sehen möchten, und bestätigen Sie Ihre Eingabe mit der *Return*–Taste.
- Gespeicherte Anfragen: Durch das Anklicken dieses Links werden die gespeicherten Anfragen in einem Fenster auf Ihrem Dashboard angezeigt.
- Administration: Durch das Anklicken dieses Links werden die Links der Admin.– Tasks in einem Fenster auf Ihrem Dashboard angezeigt.
- Kalender: Durch das Anklicken dieses Links wird ein Kalender in einem Fenster auf Ihrem Dashboard angezeigt.
- Arbeitskorbübersicht: Durch das Anklicken dieses Links wird eine Übersicht über die Anzahl der Einträge im Arbeitskorb in einem Fenster auf Ihrem Dashboard angezeigt.
- Neuigkeiten: Durch das Anklicken dieses Links werden die Neuigkeiten in einem Fenster auf Ihrem Dashboard angezeigt. Dafür muss im DMS ein Ordner mit dem Namen *News* unter *Allgemein* angelegt werden, wo dann Nachrichten hinterlegt werden können (z.B. eine Notiz).
- Termine: Durch das Anklicken dieses Links werden die Termine des heutigen Tages in einem Fenster auf Ihrem Dashboard angezeigt.



Hinweis: Jedes Fenster am Dashboard kann (wie unter Windows üblich) an eine beliebige Stelle innerhalb des Dashboards verschoben bzw. in seiner Größe verändert werden.

11.2 Öffnen

Durch das Aktivieren der Schaltfläche Öffnen kann ein bestehendes Dashboard-Profil geladen werden. Dies muss zuvor mittels der Funktion *Speichern unter* persistent gemacht worden sein. Es wird dabei zwischen persönlichen Dashboard und anderen Dashboards unterschieden. Persönliche Dashboards sind jene, die der aktuelle Benutzer abgespeichert hat. Andere Dashboards sind jene, die von anderen Benutzer über den Reiter *Zugriff* des zu speichernden Dashboards freigegeben werden (*Freigabe*-Recht).

11.3 Speichern

Durch das Aktivieren der Schaltfläche *Speichern* werden die aktuellen Einstellungen für das geöffnete Dashboard gespeichert. Dabei wird ein neuer Dialog mit folgenden Attributen geöffnet:

- Name: Der eindeutige Name des Dashboards muss hier eingegeben werden.
- Beschreibung: Frei wählbarer Text
- Default: Hier kann aus 2 Einträgen ausgewählt werden:
 - Dashboard user/admin für <User>: Dashboard-Einstellungen werden für den aktuellen Benutzer als Default gespeichert. Dabei gibt es die Unterscheidung, ob der Benutzer das Dashboard in seinem Arbeitskorb (user) oder in der Administration (admin) erstellt hat. Dieses Dashboard wird für den Benutzer standardmäßig angezeigt, falls dieser noch kein anderes besitzt.
 - Dashboard user/admin f
 ür alle Benutzer: Dashboard-Einstellungen werden f
 ür alle Benutzer als Default gespeichert. Wenn ein Benutzer kein eigenes Dashboard besitzt, wird dieses standardm
 äßig angezeigt. Auch hier gibt es die Unterscheidung user und admin.
- *Erzeugt von:* Dieses Feld kann nicht verändert werden und zeigt den Ersteller des Dashboards an.
- *Dashboard-Id:* Dieses Feld kann ebenso nicht verändert werden und zeigt an, ob das Dashboard im Arbeitskorb (user) oder in der Administration (admin) gespeichert wird/wurde.

Weiters besteht die Möglichkeit über den Reiter Zugriff das Freigabe-Recht für andere Benutzer zu vergeben. Diese können dann bei positiver Vergabe des Rechts das aktuelle Dashboard öffnen (siehe Funktion Öffnen).

Hinweis: Die Funktion *Speichern* ist nur für Benutzer sichtbar, die Erzeuger des Dashboards sind oder die Rolle *SYS* zugeordnet bekommen haben.

11.4 Speichern unter

Nachdem die Schaltfläche *Speichern unter* aktiviert wurde, öffnet sich ein neues Fenster (analog zu Funktion *Speichern*). Hier können Sie nun das aktuelle Dashboard unter einem neuen Namen speichern.

Ändert ein Benutzer durch das Aktivieren der Schaltfläche *Neu* und *Speichern unter* das Dashboard, wird es zu seinem persönlichen Dashboard und das Default–Dashboard bleibt unverändert. Der Benutzer kann aber auch ein bereits bestehendes Dashboard-Profil mittels der Funktion *Öffnen* laden. Die Identifikation erfolgt über den URL Parameter *id*.

11.5 Löschen

Mit Hilfe dieser Funktion werden die aktuellen Dashboard-Einstellungen bzw. das Dashboard-Profil (aus der Datenbank) gelöscht.

12 Administration via Kommandozeile

Dieses Kapitel beschreibt die Administration von @**enterprise** über die Kommandozeile. Dadurch ergeben sich folgende Vorteile:

- Administrationstätigkeiten können in einem Skript zusammengefasst und auf verschiedenen Servern ausgeführt werden
- Synchronisierung zwischen Entwicklungs- und Produktionsumgebung
- Übermitteln eines Skripts an den Systemoperator
- Aktionen können dokumentiert werden

12.1 Architektur und Anwendung

Das Kommandozeilen-Interface hat eine Client- und eine Serverkomponente. Die Serverkomponente ist im @enterprise Server integriert. Die Clientkomponente ist in einer separaten Datei *adminshell.jar* integriert, die im *bin*-Verzeichnis von @enterprise liegt. Der Client kann sich über HTTP oder HTTPS mit dem Server verbinden. Die Konfiguration dazu ist auf der Konfigurationsmaske *Kommunikation* zu finden. Aktiviert wird das Kommandozeilen-Interface mit dem versteckten Parameter *ep.adminshell.enable*. Nähere Details dazu finden Sie im *Installations- und Konfigurationshandbuch*.

Hinweis: Der ausführende Benutzer muss das Recht *Ausführen* auf alle Objekte besitzen! Weiters benötigt er die entsprechenden Rechte, um Server Kommandos korrekt auszuführen.

Um die Kommandozeile zu aktivieren, müssen Sie folgenden Befehl absetzen:

java -jar adminshell.jar url user [password] [-log logfile | -append logfile] [-passwdfile file] [-execute scriptfile]

Parameter:

- **url:** Die URL zum Server, z.B. *http://localhost:8083/wf/*. Wenn kein Context-Root angegeben wurde, wird automatisch *wf* als Standard verwendet.
- user: Der Benutzername des ausführenden Benutzers

• **password:** Das Passwort des Benutzers (falls vorhanden). Wenn Sie kein Passwort angeben, müssen Sie die Option *-passwdfile* verwenden oder Sie werden beim Login zur Eingabe des Passworts aufgefordert.

Optionen:

- -log logfile: Das Logfile, wo alle Interaktionen der Kommandozeile hingeloggt werden.
- **-append logfile:** Analog zu *-log*, jedoch werden die Interaktionen dem bestehendem Logfile angehängt.
- **-passwdfile file:** Diese Datei beinhaltet das Passwort (plain) für den ausführenden Benutzer ohne führende bzw. nachstehende Zeichen.
- -execute scriptfile: Führt das Skript in der angegebenen Datei aus.

12.2 Kommandos

Bei den Kommandos wird in 2 Gruppen unterschieden:

- 1. **Client Kommandos** werden auf dem Client ausgeführt und definieren das Verhalten des Skript-Clients.
- 2. Server Kommandos werden auf dem Server ausgeführt und beinhalten Funktionen zur Administration von @enterprise.

12.2.1 Client Kommandos

Folgende Client Kommandos sind verfügbar:

- exit: Beendet den Client.
- help oder ?: Gibt eine Übersicht aller Kommandos aus.
- log <file>: Schreibt Kommandos in das angegebene Logfile.
- **log off**: Kommandos werden ab diesem Zeitpunkt nicht mehr im Logfile mitprotokolliert.
- **append <file>**: Schreibt Kommandos in das angegebene Logfile. Wenn die Datei bereits existiert, werden die Kommandos der Datei hinten angehängt.
- execute <file>: Führt das angegebene Skript aus der Datei aus.

Alle Kommandos, die nicht in der oben genannten Liste vorhanden sind, werden als Serverkommandos interpretiert.

12.2.2 Server Kommandos

Diese Kommandos werden als Groovy Expressions interpretiert. Groovy ist eine Skriptsprache basierend auf Java. Kommentare haben die selbe Syntax wie in Java (Inline- und Blockkommentare). Ein Kommando wird nur durch eine Zeile mit einem Punkt-Zeichen abgeschlossen und auf Loglevel 1 oder höher im Serverlog mitprotokolliert.

Die folgenden Variablen sind im initialen Kontext enthalten (varname und Instanz von):

- admin: com.groiss.server.Admin
- **store**: com.groiss.store.Store
- engine: com.groiss.wf.WfEngine
- dms: com.groiss.dms.DMS
- orgdata: com.groiss.org.OrgData
- config: com.groiss.component.Configuration (die Systemkonfiguration)
- user: com.groiss.org.User (der aktuelle Benutzer)
- session: javax.servlet.http.HTTPSession (die HTTPSession)

Nähere Informationen über die einzelnen Objektmethoden sind im API vorhanden. Jedes Kommando wird in einer eigenen Transaktion ausgeführt. Nachdem ein Kommando ausgeführt wurde, wird entweder ein Commit oder - falls ein Fehler auftritt - ein Rollback durchgeführt.

Wenn Sie (eigene) Variablen im Skript verwenden wollen, können Sie diese mit folgendem Kommando definieren:

set(varname,value);

Um den Wert wieder auszulesen, müssen Sie folgendes Kommando absetzen:

get(varname);

Selbst deklarierte Variablen haben den Vorteil, dass sie über eine Transaktion hinaus verfügbar sind, da sie in die Session geschrieben werden.

12.3 Beispiele

12.3.1 Einen Konfigurationsparameter setzen

config.setProperty("database.connections",5); config.store();

Alternative Version mit eigenen Variablen:

```
set("connections",5);
config.setProperty("database.connections",get("connections"));
config.store();
```

12.3.2 Serverneustart

admin.restartServer(); //restarts the server - no login necessary for current user

12.3.3 Rolle einem Benutzer hinzufügen bzw. entziehen

```
u = orgdata.getById(com.groiss.org.User.class,'my_user'); //replace by existing user
role = orgdata.getById(com.groiss.org.Role.class,'sys'); //get SYS role
checkuserrole = store.get(com.groiss.org.UserRole.class,"role = ? AND userid = ?",
 role.getOid(), u.getOid()); //with prepared statements - new Object[] {args}
//If User has no sys-role, add it
if(checkuserrole == null) {
userrole = orgdata.createUserRole();
userrole.setRole(role);
userrole.setUser(u);
userrole.setActive(true);
orgdata.insert(userrole);
}
//If User has sys-role, remove it
else {
orgdata.delete(checkuserrole);
}
```

12.3.4 Zeitintervall eines Timers setzen

```
t = orgdata.getById(com.groiss.timer.TimerEntry.class,'Suspension');
t.setPattern("360");
store.update(t);
```

12.3.5 Arbeitskorb Handling

Dieses Beispiel durchsucht den Arbeitskorb der Applikation *default* und beendet bereits abgelaufene Tasks, sofern diese beendet werden können:

```
appl = orgdata.getById(com.groiss.org.Application.class, "default");
worklist = engine.getWorklist(appl,true);
for(com.groiss.wf.ActivityInstance ai:worklist) {
    duedate = ai.getDuedate();
    //if ai's duedate is expired, finish task
    if(duedate != null && duedate.getTime() < new java.util.Date().getTime()) {</pre>
```

```
try {
  engine.finish(ai);
  }
  catch(ex) {/*Do nothing, but continue with finishing other ai's*/};
  }
}.
```

12.3.6 Session Handling

Dieses Beispiel prüft die derzeitige Session und invalidiert sie, falls der letzte Zugriff (lastAccessed) nicht innerhalb der Toleranzzeit liegt. Die Informationen der Session werden ins Serverlog von @enterprise ab Loglevel 2 geschrieben:

```
attrbnames = session.getAttributeNames();
invalidate = false;
\log = " \setminus n";
\log = \log + "Session-Parameter:\n";
for(String attrname:attrbnames) {
attrvalue = session.getAttribute(attrname);
if(attrname.equalsIgnoreCase("lastAccessed")) {
 if(attrvalue instanceof java.util.Date) {
 onehour = 60*60*1000; //tolerance time
 //invalid, if not in tolerance time
 if((attrvalue.getTime()+onehour) >= new java.util.Date().getTime() ||
    (attrvalue.getTime()-onehour) <= new java.util.Date().getTime()) {
  invalidate = true;
 }
 }
}
log = log + "Attribute-Name: " + attrname + "/Attribute-Value: " + attrvalue + "\n";
}
\log = \log + "";
com.groiss.util.Settings.log(log,2); //write all session parameter to Server-Log on Level 2
if(invalidate == true) {
session.invalidate();
}
```

13 Prozess–Cockpit

Das Prozess–Cockpit gibt eine Übersicht über die Prozesse der Organisation. Es stellt Informationen sowohl über die Definition als auch über die Instanzen der Prozesse zur Verfügung. Im Standard-GUI befindet sich der Link zum Cockpit in der Gruppe *Extras*.

13.0.1 Konfiguration

Die Prozesse werden in einer Hierarchie dargestellt, die im DMS von @enterprise erstellt wird. Zu diesem Zwecke müssen Sie eine Struktur von Ordnern abbilden, die den Prozessen Ihrer Organisation entspricht, z.B:

- 1. Operating processes
 - (a) Manufacture
 - (b) Marketing
 - i. Manage sales plans

In der Konfiguration von @enterprise muss dann der Pfad auf die Wurzel dieser Struktur festgelegt werden (siehe *Installationshandbuch* - Kapitel *Process Cockpit*).

Die einzelnen Knoten des Baums sollten mit dem Prozess-Cockpit Ordner-Formular erstellt werden, um die Cockpit-Funktionen vollständig nutzen zu können (Zwischenknoten können aber auch normale Ordner sein).

Im Ordnerformular kann zuerst der Typ ausgewählt werden (siehe Abbildung 13.1):

- **Prozess mit Definition:** Hier können Sie genau 1 Prozessdefinition aus der Liste auswählen. Mit der Checkbox *Instanzen anzeigen* können Sie festlegen, ob Instanzen im Reiter *Laufzeit* angezeigt werden sollen oder nicht.
- **Prozess ohne Definition:** Für Prozesse, zu denen es keinen eindeutigen @enterprise Prozess gibt. Es kann ein Name und eine verbale Beschreibung des Prozesses erfasst werden. Für diese Art der Prozesse erfolgt die Zuordnung zu Instanzen folgendermaßen:

Es gibt Prozesse mit Prozessformularen, die ein Formularfeld mit dem Namen *area* enthalten. Der Wertebereich des Feld *area* sind die Knoten aus dem Prozess-Cockpit. Die in Frage kommenden Prozesse werden in der Konfiguration definiert beim Parameter *Allgemeine Prozesse* (siehe *Installationshandbuch* - Kapitel *Process Cockpit*). Ein konkretes Beispiel dazu ist im *Benutzerhandbuch* im Abschnitt *Details für Prozesse* vorhanden. • **Prozessgruppe:** Zwischenknoten im Prozessbaum, der nur einen Namen und eine Beschreibung benötigt. In diesem Fall können zusätzlich Dokumente im Ordner hinzugefügt werden, die dann auf der Prozessdetailseite explizit als Links in der Tabelle *Dokumente* angezeigt werden.

Weitere Felder im Cockpit Formular sind je nach Typ sichtbar:

- Verantwortlich: Die Person, die verantwortlich ist für den Prozess bzw. die Prozessgruppe.
- Verfügbare Reports: Eine Liste von Reports, die den Prozess betreffen und auf der Prozessdetailseite (z.B. im Reiter *Laufzeit*) als Link angezeigt werden.
- Direkt ausgeführte Reports: Eine Liste von Reports, die sofort auf der Prozessdetailseite (z.B. im Reiter *Laufzeit*) ausgeführt werden.
- **Funktionen:** Eine Liste von Funktionen, die auf der Prozessdetailseite (z.B. im Reiter *Laufzeit*) als Link angezeigt werden.
- Links: Eine Liste von Links bestehend aus URL und einem Text, die auf der Prozessdetailseite (z.B. im Reiter *Laufzeit*) angezeigt werden.

Einen Link zu den Ordnerformularen gibt es auch im Toolbar der Detailansicht des Prozess-Cockpits. Dort kann jeweils die Darstellung des gerade angezeigten Prozesses (oder die Prozessgruppe) konfiguriert werden.

13.0.2 Berechtigungen

Jeder, der das Recht *Objekte bearbeiten* auf die Ordner des Prozess-Cockpits hat, darf dieses manipulieren. Jeder, der das Recht *Objekte ansehen* auf einen Ordner hat, sieht den Link und die zugehörige Detailseite im Cockpit. Die Reports zu den Instanzen (laufend, beendet, diese Woche, dieses Monat, überfällig) werden ebenfalls ohne weitere Rechteprüfung ausgeführt. Bei den weiteren Reports wird wie üblich das Recht *Objekt ausführen* geprüft.

🥙 Cockpit-Ordner - Mozilla Firefox		_ 🗆 🔀
http://localhost:8380/wf/servlet.method/com.groiss.st	oregui.FormWrapper.showForm?date=1310107693164&object=com.groiss.cockpit.Cock	pitFo 🏫 🛞
●Prozess mit Definition ○Prozess o	hne Definition OProzessgruppe	
Prozessdefinition Candidature Process (1)	▼	
Instanzen 🔽 anzeigen		
Verantwortlich Irrasch Markus markus	▼	
Verfügbare Reports	Direkt ausgeführte Reports	
Arbeitskorb eines Benutzers Anzahl Prozesse pro Prozesstyp und OE Anzahl offener Probleme pro Akteur Anzahl offener Incidents pro Akteur	Arbeitskorb eines Benutzers	▲ ± ★
Funktionen		
Links URL	Text	
http://www.extern.com	EXTERN	
http://intern:8123	INTERN	
Speichern Speichern und Schließen	Abbrechen	

Abbildung 13.1: Prozess–Cockpit Formular