deegree User Rights, Roles and Resources (U3R) v2.5

lat/lon GmbH
Aennchenstr. 19
53177 Bonn
Germany
Tel ++49 - 228 - 184 96-0
Fax ++49 - 228 - 184 96-29
info@lat-lon.de
www.lat-lon.de

Geographisches Institut
Universität Bonn
Meckenheimer Allee 166
53115 Bonn

Tel. ++49 228 732098

Change log

Datum	***Beschreibung***	***Author***
2006-10-17	Verallgemeinerung des Dokumentes aus projektspezifischer Dokumenation	Markus Müller
2007-01-08	Abbildungen aktualisiert; Definition von Transaktionsrechten beschrieben	Andreas Poth
2007-01-16	Zusammenfassung der drei Dokumentationen zu U3R in ein Dokument; Harmonisierung mit Standard-deegree Dokumentationsstruktur	Markus Müller
2007-03-02	Präzisierung unklarer Stellen zur Verwaltung von FeatureTypes, dem Ändern des SEC_ADMIN Passwortes, zur Benutzung des Kommandozeilentools und einer Reihe weiterer Details.	Markus Müller, Andreas Poth
2007-04-23	Bug Fixes	Andreas Poth
2007-06-01	Erweiterung um context chooser für iGeoPortal	Judit Mays
2007-06-22	Ergänzungen für context chooser	Judit Mays
2007-06-22	Kleine Verbesserungen	Markus Müller
2009-07-27	Version geändert	Andreas Poth
	Aktualisierung auf Version 2.3 (to be done)	

Inhalt

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

deegree ist ein Java Framework, das die Bausteine für Geodateninfrastrukturen (GDI) zur Verfügung stellt. Seine ganze Architektur wurde auf Basis von Standards des Open Geospatial Consortium (OGC) und des ISO Technical Committee 211 – Geographic information / Geoinformatics (ISO/TC 211) entwickelt. a Java Framework offering the main building blocks for Spatial Data Infrastructures (SDIs). deegree umfasst außer OGC Web Services auch Clients. deegree ist Freie Software, wird durch die GNU Lesser General Public Licens (GNU LGPL) geschützt und steht über http://www.deegree.org zur Verfügung.

deegree2 ist das neue Release von deegree, das eine Anzahl von Funktionalitäten zur Verfügung stellt, zu denen deegree1 nicht in der Lage war. Diese Dokumentation beschreibt die Installation und Konfiguration von deegree User Rights, Roles and Resources (U3R), der Benutzer- und Rechteverwaltung von deegree. U3R ist ein Teil von deegree iGeoSecurity und kann zusammen mit deegree owsProxy und dem WASS eingesetzt werden.

Dieses Dokument wendet sich an die Administratoren von deegree U3R d. h. diejenigen die Benutzer, Layer und FeatureTypes bei ihr registrieren und die Zugriffsrechte auf letztere vergeben. Es beschreibt die Installation und Konfiguration der deegree Benutzer- und Rechteverwaltung. Die Beschreibung der Funktionsweise und der Möglichkeiten des Systems wird ebenfalls vorgenommen.

Die mit der deegree Benutzer- und Rechteverwaltung zur Verfügung stehenden Möglichkeiten zur Definition von Benutzern, Rechten und ihrer Verknüpfung sind äußerst vielfältig. Selbst extrem komplexe Baum- und Netzstrukturen mit entsprechenden Vererbungen regelbasierter Zugriffsrechte sind abbildbar. Damit das System für den Anwender (Administrator) wartbar bleibt, empfehlen wir dringend, vor der Nutzung ein formales und verbindliches Rechtekonzept zu entwickeln, das dann mit der deegree Benutzer- und Rechteverwaltung umgesetzt wird.

Außer U3R umfasst deegree eine Anzahl von Diensten und Clients. Eine komplette Liste der deegree Komponenten ist einsehbar unter:

`http://www.lat-lon.de` → Produkte

Installationspakete bestimmter Komponenten können unter der folgenden URL gefunden werden:

`http://www.deegree.org` → Download

Die Web Services und Benutzeroberflächen von deegree sind als Javamodule implementiert, die von einem zentralen Servlet (dem “dispatcher”) gesteuert werden. Dieses Servlet muss in einem entsprechenden Web Server / Servlet Engine veröffentlicht werden. Die meisten der verbreiteten Web Server unterstützen Servlettechnologie, womit deegree universell einsetzbar ist. Der Einsatz der Apache Tomat 5.5 Servlet Engine wird empfohlen aufgrund seiner großen Verbreitung und seines Status als Open Source-Produkt.

2 Download / Installation

2.1 Voraussetzungen

Zur Installation von deegree U3R werden benötigt:

- Java (JRE or JSDK) version 1.5.x
- Tomcat 5.5.x

Zur Installation dieser Komponenten sollte die entsprechende Dokumentation auf java.sun.com und tomcat.apache.org herangezogen werden.

2.2 deegree U3R Release

Bislang steht kein Installationspaket für deegree U3R zur Verfügung. Aus diesem Grund müssen die entsprechenden Dateien aus dem deegree SVN bezogen werden.

Um U3R zu installieren werden die folgenden Komponenten benötigt:

- Datenbankschema zur persistenten Ablage aller Benutzer und Rechte.
- Java Server Pages (JSPs), realisieren die Administrationsmasken für die Rechteverwaltung
- Java-Klassen, enthalten ein Application Programming Interface (API) und Hilfsklassen zur Rechteverwaltung. Sie sind Bestandteil von deegree und im Archiv deegree2.jar enthalten.

2.3 Einrichten des Datenbankschemas

Zur Inbetriebnahme der Rechteverwaltung muss zunächst das benötigte Datenbankschema eingerichtet werden. Zur Zeit ist die Verwendung von Postgres und Oracle als Datenbanken möglich. Die Installationsdateien der deegree Benutzer- und Rechteverwaltung umfassen zwei SQL-Skripte (eines für jede der genannten Datenbanken) mit deren Hilfe alle benötigten Tabellen angelegt und initialisiert werden können.

Um sich später bei der Rechteverwaltung anmelden und neue Nutzer einrichten zu können, legt das Initialisierungsskript einen Administrator an. Dieser wird mit dem Namen 'SEC_ADMIN' und dem Passwort 'JOSE67' eingerichtet. Während der Name des Administrators fix ist, sollte das Passwort auf jeden Fall geändert werden. Zu diesem Zweck muss die folgende Zeile des Initialisierungsskripts angepasst werden:

```
INSERT INTO SEC_USERS ("ID", "PASSWORD", "FIRSTNAME", "LASTNAME", "EMAIL") VALUES (1, 'JOSE67', 'SEC_ADMIN', 'SEC_ADMIN', 'admin@myhost.de');
```

An dieser Stelle sollte auch die Email-Adresse eingetragen werden, an die Systemmeldungen verschickt werden sollen. Alternativ können beide Angaben auch später über Datenbankmechanismen angepasst werden.

Nach Anpassen des Skripts muss dieses mit einem geeigneten Datenbank-Client oder per Kommandozeile ausgeführt werden. Läuft das Skript ohne Probleme durch, sollten sich in der Datenbank die folgenden Tabellen befinden:

SEC_JT_GROUPS_GROUPS

SEC_JT_GROUPS_ROLES

SEC_JT_ROLES_PRIVILEGES

SEC_JT_ROLES_SECOBJECTS

SEC_JT_USERS_GROUPS

SEC_JT_USERS_ROLES

SEC_PRIVILEGES

SEC_RIGHTS

SEC_ROLES

SEC_USERS

SEC_GROUPS

SEC_SECURED_OBJECT_TYPES

SEC_SECURED_OBJECTS

SEC_SECURABLE_OBJECTS

(siehe Abb. 9)

2.4 Installieren des Web-Frontend

Nach dem erfolgreichen Einrichten des Datenbankschemas kann die Administrationsoberfläche eingerichtet werden. Diese ist als eine Reihe von JSP-Seiten angelegt, deren Aufruf durch ein zentrales Java-Servlet gesteuert wird. Daher muss sie im Web Context eines Servlet Containers (z. B. Apache Tomcat) eingebunden werden.

[Auszug aus `$TOMCAT_HOME$/conf/server.xml`]

```
...
<Context path="/drm-admin" docBase="D:/java/webapps/drm-admin" debug="0"
 reloadable="true">
   <Logger className="org.apache.catalina.logger.FileLogger" directory="logs"
     prefix="log_drm-admin." suffix=".txt" timestamp="true"/>
</Context>
...
```

Alle deegree-spezifischen Klassen sind im Archiv deegree2.jar enthalten. Zusätzlich werden die folgenden Archive benötigt[1]:

- commons-codec-1.3.jar
- commons-httpclient-2.0.2-deegreeversion.jar
- commons-discovery-0.2.jar
- commons-logging.jar
- jaxen-1.1-beta-8.jar
- j3dcore.jar
- j3dutils.jar
- vecmath.jar
- jts-1.8.jar
- acme.jar
- jai_codec.jar
- jai_core.jar
- mlibwrapper_jai.jar
- vecmath.jar
- log4j-1.2.9.jar
- mail.jar
- xerces_2_5_0.jar
- xml-apis.jar
- postgresql-8.0-311.jdbc3.jar bzw. ojdbc14_10g.jar

Bei der Registrierung müssen dem Servlet im Deployment Descriptor (web.xml) mehrere Initialisierungsparameter mitgegeben werden. Im folgenden ist ein Beispiel eines Deployment Descriptors wiedergegeben, die Initialisierungsparameter werden anschließend im Detail erklärt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
   <servlet>
      <servlet-name>SecurityRequestDispatcher</servlet-name>
      <servlet-class>
         org.deegree.portal.standard.security.control.SecurityRequestDispatcher
      </servlet-class>

      <init-param>
```

[1]Im Fall des Apache Tomcat reicht es aus, die Archive in das lib-Verzeichnis des entsprechenden Kontextes zu kopieren.

```
        <param-name>Handler.configFile</param-name>
        <param-value>
           WEB-INF/conf/security_controller.xml
        </param-value>
     </init-param>
     <init-param>
        <param-name>Security.configFile</param-name>
        <param-value>
           WEB-INF/conf/security.xml
        </param-value>
     </init-param>
  </servlet>
  <servlet-mapping>
     <servlet-name>SecurityRequestDispatcher</servlet-name>
     <url-pattern>/SecurityRequestDispatcher</url-pattern>
  </servlet-mapping>
</web-app>
```

Über `Handler.configFile` wird dem Servlet mitgeteilt, wo sich die Konfigurationsdatei zur Verteilung von Anfragen an interne Handler-Klassen und JSP-Seiten befindet. Diese Datei muss/sollte nicht modifiziert werden, da ansonsten die korrekte Funktion der Anwendung nicht mehr garantiert werden kann.

Der letzte Parameter, `Security.configFile`, definiert eine Referenz auf eine XML-Datei, die die Zugriffsparameter auf die anzubindende Datenbank enthält.

```
<security>
   <registryClass>org.deegree.security.drm.SQLRegistry</registryClass>
   <readWriteTimeout>300</readWriteTimeout>
   <registryConfig>
      <jdbc:JDBCConnection xmlns:jdbc="http://www.deegree.org/jdbc">
         <jdbc:Driver>oracle.jdbc.OracleDriver</jdbc:Driver>
         <jdbc:Url>jdbc:oracle:thin:@localhost:1521:latlon</jdbc:Url>
         <jdbc:User>security</jdbc:User>
         <jdbc:Password>security</jdbc:Password>
         <jdbc:SecurityConstraints/>
         <jdbc:Encoding>iso-8859-1</jdbc:Encoding>
      </jdbc:JDBCConnection>
   </registryConfig>
</security>
```

- driver: Java Treiberklasse für den JDBC-Zugang (abhängig von der verwendeten Datenbank; im Beispiel ist eine ODBC-Datenbank konfiguriert)
- url: Adresse und Name unter der die entsprechende Datenbank über den JDBC-Treiber verfügbar ist
- user: Benutzername (optional)
- password: (optional)
- SecurityConstraints und Encoding müssen angegeben werden, dienen im Augenblick aber lediglich als Platzhalter und werden nicht ausgewertet.

2.5 Ändern des Passwortes von SEC_ADMIN

Bei der Erstinstallation von U3R ist ein default-Passwort eingerichtet, das unbedingt geändert werden sollte. Um nach Änderung des Administrator-passworts das Kommandozeilen-Tool DRMAccess (siehe Kapitel 5) benutzen zu können, muss `org/deegree/tools/security/sec.properties` entsprechend angepasst werden.

3 Grundlagen

3.1 Das Konzept von U3R

Das Rechtemanagement von deegree dient grundsätzlich dazu, festzustellen, auf welche Objekte ein Benutzer wie zugreifen darf. Dazu werden die folgenden zentralen Konzepte verwendet:

Benutzer:

- Die Person, die auf eine Ressource zugreift.
- Zur Identifikation wird der Benutzername und ein Passwort verwendet[2].
- Die Benutzerdaten können von der Anwendung, die die Rechteverwaltung nutzt, **nicht** editiert werden, dies ist nur über die Administrationsoberfläche möglich.

Gruppen:

- Dienen der Zusammenfassung von Benutzern zu Gruppen.
- Gruppen können zu Supergruppen zusammengefasst werden, d. h. sie können Mitglied der Supergruppe sein.
- Es können netzartige Beziehungen zwischen beliebigen Gruppen definiert werden.

Rollen:

- Sind Sammlungen von Rechten bzgl. der Datensätze (d. h. eine Rolle besitzt für jeden einzelnen Datensatz das Zugriffsrecht oder nicht).
- Zugriffsrechte können parametrisiert werden; so ist es z. B. möglich das Zugriffsrecht auf einen Layer zu vergeben, das auf einen bestimmten Raumbereich eingeschränkt wird.
- Die zugewiesenen Rechte können jeder Zeit editiert werden.
- Rollen werden mit Gruppen assoziiert. Eine Rolle kann dabei beliebig vielen Gruppen zugewiesen werden. Außerdem kann eine Gruppe auch mehrere assoziierte Rollen besitzen. Die Mitglieder dieser Gruppe besitzen dann die akkumulierten Rechte aller Rollen, die mit der Gruppe verbunden sind.

3.2 Beispiel für die Akkumulation von Benutzerrechten

Ein sehr einfaches Beispiel für eine mögliche Rollen-/Rechtevergabe:

[2]Alternative Konzepte wie die Übernahme der Netzwerkkennung eines Benutzers wären auch realisierbar.

	Datensatz DS1	***Datensatz DS2***	***Datensatz DS3***	***Erläuterung***
Rolle R1	x	x	-	R1 hat Zugriff auf DS1 und DS2
Rolle R2	-	x	-	R2 hat Zugriff auf DS2
Rolle R3	x	-	-	R3 hat Zugriff auf DS1
Rolle R4	-	-	x	R4 hat Zugriff auf DS3

Tabelle 1: Zuordnung von Rollen und Datensätzen

	Rolle R1	***Rolle R2***	***Rolle R3***	***Rolle R4***	***Erläuterung***
Gruppe G1 (Altlasten)	x	x	-	-	G1 ist assoziiert mit R1 und R2
Gruppe G2 (Chemie)	-	x	x	-	G2 ist assoziiert mit R2 und R3
Gruppe G3 (Test)	-	-	-	x	G3 ist assoziiert mit R4

Tabelle 2: Zuordnung von Gruppen und Rollen

Benutzer **Mustermann** sei Mitglied der Gruppen **Altlasten** und **Chemie**. Wie bestimmt das System, auf welche Datensätze **Mustermann** zugreifen darf?

1. Es wird ermittelt, welchen Gruppen **Mustermann** angehört, in diesem Beispiel also **Altlasten** und **Chemie**.
2. Die Gruppe **Altlasten** ist mit den Rollen **R1** und **R2** assoziiert, die Gruppe **Chemie** mit **R2** und **R3**.
3. Folglich ist der Benutzer mit den Rollen **R1**, **R2** und **R3** assoziiert (die doppelte Zuordnung von R1 ist irrelevant).
4. Für die Rolle **R1** ergeben sich Rechte bzgl. **DS1** und **DS2**. **R2**: **DS2**, **R3**: **DS1**.
5. Der Benutzer besitzt also Zugriffsrechte auf die Datensätze **DS1** und **DS2** (die mehrfachen Zuordnungen von Rechten sind wiederum irrelevant).

3.3 Administratoren

Für die Administration und die Rechteverwaltung des Systems werden zusätzlich zu den erwähnten Benutzern, Gruppen und Rollen noch folgende Konzepte verwendet:

- Administratoren sind alle Benutzer, die mit der Rolle „SEC_ADMIN“ verknüpft sind. „SEC_ADMIN“ ist eine besondere Rolle, die vom System intern verwendet wird und nicht gelöscht werden kann.
- Administratoren haben (als einzige) Zugriff auf die Administrationsmasken, sie dürfen:
- Benutzer anlegen, löschen und editieren
- Gruppen anlegen und löschen
- Gruppenzugehörigkeiten editieren
- Rollen anlegen, löschen und editieren
- Rollen-Gruppen Zugehörigkeiten ändern
- Rechte mit Rollen assoziieren (und Rechte-Constraints definieren)

U3R kann entweder mit einer graphischen Web-Oberfläche oder einem Kommandozeilenwerkzeug verwaltet werden. Diese beiden Möglichkeiten werden in den beiden nächsten Kapiteln erläutert.

4 Die Oberfläche

Die Administrationsmasken sind mittels dynamischer Webseiten (JSPs) und JavaScript realisiert, um ein möglichst komfortables und zügiges Arbeiten zu gewährleisten. Allerdings bedeutet das auch, dass nicht jede Veränderung unmittelbar auf dem Server abgespeichert wird. Grundsätzlich gilt deshalb für alle Masken:

- **„übernehmen**“ sendet die vorgenommenen Änderungen an den Server und speichert sie. Die Durchführung der Änderungen wird auf der folgenden Seite im Browser bestätigt.
- **„abbrechen“** verwirft alle vorgenommen Änderungen und stellt im Browser den Ausgangszustand wieder her, der unmittelbar nach dem Laden der Maske bestand.

An sinnvollen Stellen ist es möglich, mehrere Einträge einer Liste zu selektieren und diese auf einmal zu verarbeiten (z. B. mehrere Gruppen einer Rolle zuzuordnen). Mehrfachauswahl erfolgt durch Gedrückthalten der Umschalttaste (SHIFT) und gleichzeitiges Klicken auf die gewünschten Einträge.

Nach dem Aufruf der Anwendung (`http://myhost[:port]/[path]` -> z. B. `http://127.0.0.1:8080/drm-admin`) erscheint die initiale Login-Seite (Abb. 1Fehler: Referenz nicht gefunden). Hier meldet sich der Benutzer mit einem gültigen Nutzernamen und Passwort an. Unmittelbar nach der Installation der Anwendung steht lediglich der Systemadministrator (SEC_ADMIN) als Nutzer zur Verfügung. Seine Aufgabe ist es u. a. neue Nutzer in der Rechteverwaltung zu registrieren.

Abbildung 1: Initiale Login-Seite

Nach erfolgreicher Anmeldung wird dem Nutzer die zentrale Seite zur Navigation in der Benutzer- und Rechteverwaltung angezeigt (Abb. 2). Von hier kann zu den verschiedenen Unterseiten zum Registrieren und Löschen von Ressourcen (*Layers/FeatureTypes*), zum Anlegen, Löschen und Editieren von Nutzern (*Benutzer*), zum Anlegen, Löschen und Editieren von Gruppen (*Gruppen*) und zum Anlegen, Löschen und Editieren von Rollen (*Rollen*) navigiert werden. Ferner kann sich der aktuelle Benutzer beim System wieder abmelden (*logout*).

Die obere Menüleiste bleibt auf allen Seiten sichtbar, so dass jederzeit von einem Menüpunkt in einen anderen gewechselt werden kann.

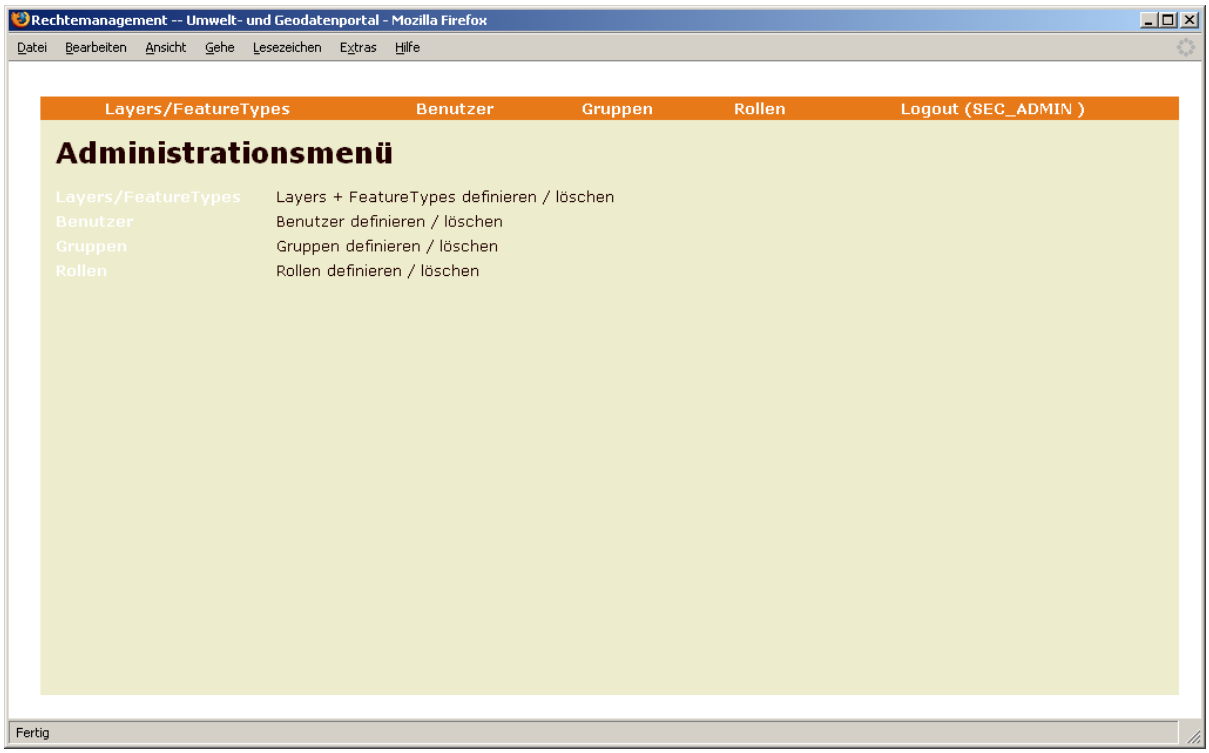

Abbildung 2: Zentrale Seite zur Navigation

4.1 Layers/FeatureTypes

Jedes über die Rechteverwaltung zu schützende Objekt muss zunächst bei dieser registriert werden. Die Art dieser Objekte ist nicht begrenzt. Um die Eingabe für den Nutzer möglichst einfach zu halten, beschränkt sich die vorliegende Administrationsoberfläche auf Optionen zur Eingabe von (WMS)Layern und (WFS/WFS-G)FeatureTypes (Abb. 3).

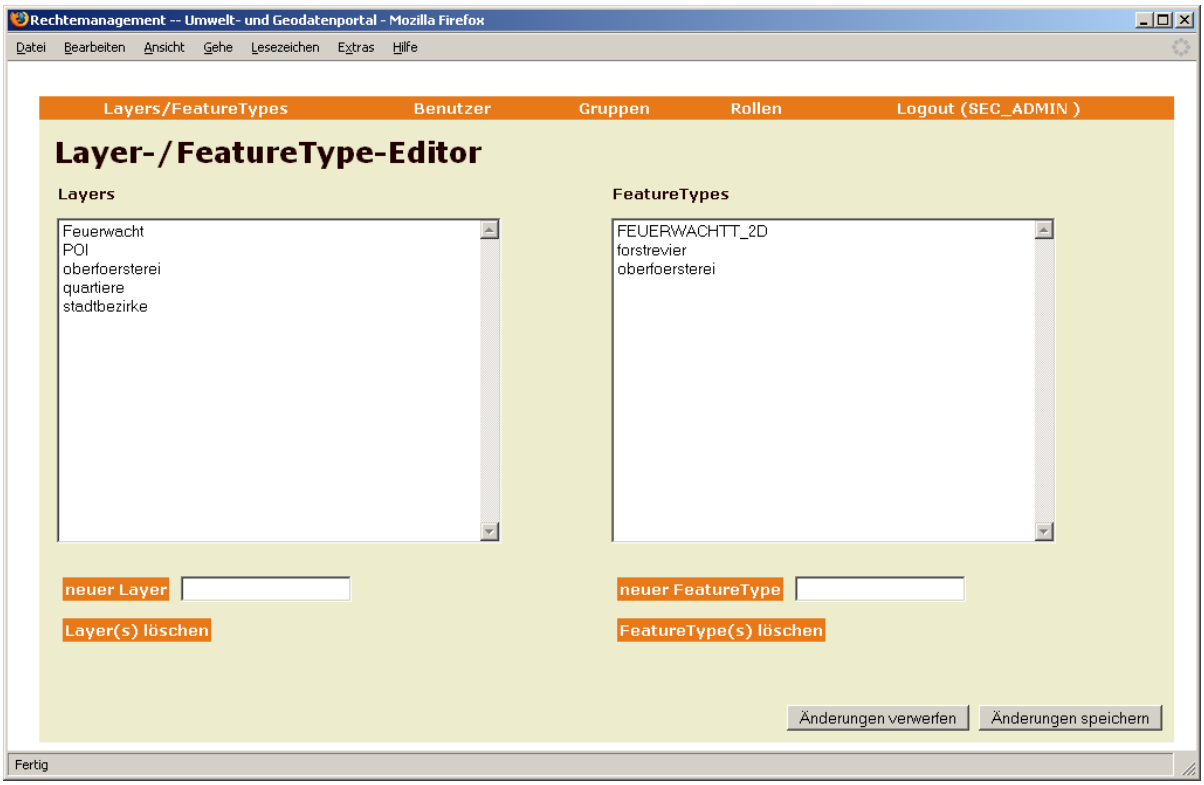

Abbildung 3: Formular zur Eingabe neuer Layer und FeatureTypes sowie zum Löschen bereits bestehender Layer und FeatureTypes

Die Bedienung des Formulars ist weitestgehend selbsterklärend. Durch Eingabe eines neuen Layers bzw. FeatureTypes in die Felder rechts unterhalb der Listen der bereits bekannten Layer/FeatureTypes und Mausklick auf die jeweils sich links davon befindende Schaltfläche werden diese in die Listen übernommen. **Hinweis:** FeatureTypes müssen im Falle der Verwendung eines WFS der Version 1.1.0 mitsamt ihres namespaces angegeben werden. Die Syntax hierfür ist folgendermaßen:

```
{http://www.deegree.org/app}:FT1
```

Der namespace muss also innerhalb geschweifter Klammern dem Namen des FeatureType vorangestellt werden.

Durch Markieren von Einträgen in den Listen und einen Mausklick auf die Schaltfläche 'selektierte Layer löschen' bzw. 'selektierte FeatureTypes löschen' werden diese aus den Listen entfernt. Zu beachten ist, dass, wie oben dargestellt, alle gemachten Änderungen erst dann in die Rechteverwaltung übernommen werden, wenn die Schaltfläche 'übernehmen' mit der linken Maustaste angeklickt und der sich anschließend öffnende Dialog bestätigt wird.

4.2 Benutzer-Editor

Über den Benutzer-Editor besteht die Möglichkeit neue Benutzer bei der Benutzerverwaltung anzumelden sowie bestehende Benutzer aus ihr zu entfernen bzw. ihre Stammdaten zu ändern (Abb. 4). Hinsichtlich der einem Nutzer zugeordneten Stammdaten ist das System weitestgehend offen. Bei entsprechender Anpassung des Datenbankschemas können beliebige Attribute einem Nutzer zugeordnet werden. Verbindlich sind lediglich:

- Benutzername (Name mit dem sich ein Benutzer beim System anmeldet)
- Passwort
- Email-Adresse

Optional realisiert sind in der vorliegenden Benutzerverwaltung die Felder Vorname und Nachname. Sie dienen lediglich dazu, um einen Nutzer z. B. bei einer Benachrichtigung persönlich ansprechen zu können.

Nach Aufruf der Benutzerverwaltung öffnet sich das in Abbildung 4 wiedergegebene Fenster in dessen linkem Teil sich eine Liste der bereits registrierten Benutzer befindet. Unterhalb der Liste befindet sich eine Schaltfläche und ein Eingabefeld zum Anlegen neuer Nutzer und eine Schaltfläche mit der selektierte Nutzer aus der Benutzerliste gelöscht werden können. Rechts neben der Benutzerliste befinden sich sechs Eingabefelder die sowohl dazu dienen die Daten des aktuell ausgewählten Benutzers anzuzeigen als auch dazu diese zu ändern bzw. um neue Nutzer anzulegen.

Um die Daten eines bestehenden Benutzer zu modifizieren, können die Inhalte der sechs Eingabefelder einfach editiert werden. Zu beachten ist, dass die Felder 'Benutzername', 'eMail' und 'Passwort' zwingend ausgefüllt werden müssen. Durch einen Mausklick auf die Schaltfläche 'Benutzerdetails ändern' werden die gemachten Änderungen in der GUI übernommen. Um einen neuen Benutzer anzulegen, Wird in das Textfeld unter der Benutzerliste der Name des neuen Benutzers eingetragen und anschließend mit der Maus auf die Schaltfläche 'neuer Benutzer' geklickt. Wird der neu angelegte Benutzer in der Liste ausgewählt, können in die leeren Textfelder seine Daten eingetragen werden.

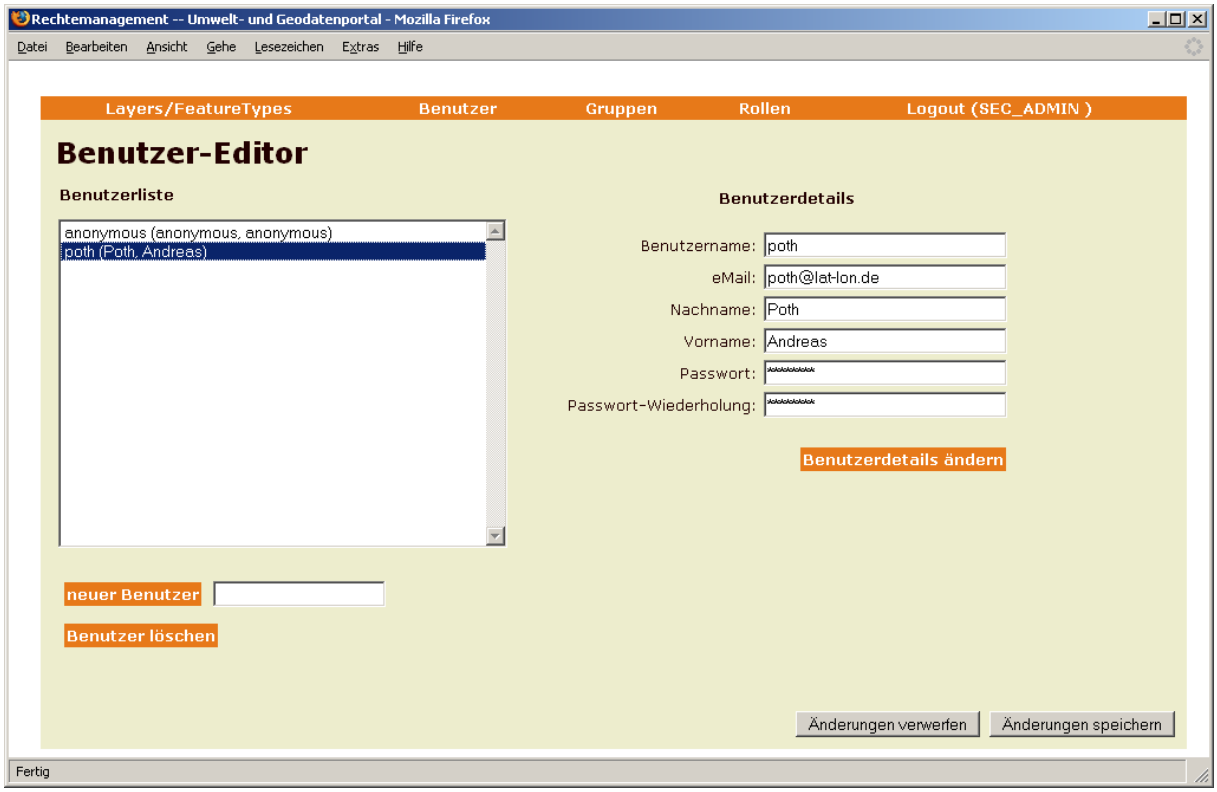

Abbildung 4: Formular zur Verwaltung von Benutzern

Wie beim Layers/FeatureTypes-Formular gilt auch für die Benutzerverwaltung, dass alle Änderungen erst nach einem Mausklick auf die Schaltfläche 'Änderungen speichern' tatsächlich in der Datenbank eingetragen werden.

4.2.1 Erweiterung des Benutzer-Editors: WebMapContext zuweisen

Die Benutzerverwaltung kann so erweitert werden, dass zusätzlich zu den oben genannten Punkten jedem Benutzer ein WebMapContext als Startkontext zugewiesen werden kann. Diese Erweiterung ist bei der Verwendung von iGeoPortal sinnvoll, aber nicht zwingend.

Zur Aktivierung dieser Erweiterung sind die folgenden Änderungen notwendig:

1. Definition des Übergabeparameter “configFile”
2. Konfiguration der “configFile”

4.2.1.1 Definition des Übergabeparameters configFile

In der Datei `security-controller.xml` (Verzeichnis `WEB-INF/conf/security/` oder `WEB-INF/conf/drm-admin/`) wird dem Event “initUserEditor” ein Übergabeparameter mitgegeben:

```
<event name="initUserEditor"
class="org.deegree.portal.standard.security.control.InitUserEditorListener"
next="usereditor.jsp">
    <!-- you might want to comment in this parameter in order to use
         the context chooser extension for iGeoPortal -->
    <!--
    <parameter>
        <name>configFile</name>
        <value>WEB-INF/conf/security/config_startcontext.xml</value>
    </parameter>
    -->
</event>
```

Der Parameter definiert, wo die Konfigurationsdatei für die WebMapContexte zu finden ist. Diese Datei wird bei der Initialisierung des Benutzer-Editors ausgewertet und verarbeitet.

4.2.1.2 Anpassung der Konfigurationsdatei (configFile)

Die Konfigurationsdatei beschreibt, welche Web Map Context Dateien einem Benutzer über die Bedienoberfläche zugewiesen werden können (`availableWMC`), bei welchem Kontext es sich um den Default-Kontext handelt(`isDefault="1"`) und wo das Benutzerverzeichnis von iGeoPortal zu finden ist (`UserDirectory`).

```
<?xml version="1.0" encoding="UTF-8"?>
<deegree:Drm xmlns:deegree="http://www.deegree.org/security">
    <deegree:availableWMC>
        <!--
        must contain an entry for each start context that shall be used by
        the context chooser. Relative paths from this file to
        $iGeoPortal_home$ probably need to be adjusted on your system
        -->
        <deegree:WMC isDefault="1">
            <deegree:Name>startcontext</deegree:Name>
            <deegree:URL>
   ../../../../deegree2_igeo_std/WEB-INF/conf/igeoportal/wmc_start_utah.xml
            </deegree:URL>
        </deegree:WMC>
        <deegree:WMC>
            <deegree:Name>saltLakeCity</deegree:Name>
            <deegree:URL>
     ../../../../deegree2_igeo_std/WEB-INF/conf/igeoportal/wmc_saltlake.xml
            </deegree:URL>
        </deegree:WMC>
        <!-- add more context files here -->
    </deegree:availableWMC>
    <deegree:UserDirectory>
        ../../../../deegree2_igeo_std/WEB-INF/conf/igeoportal/users
    </deegree:UserDirectory>
</deegree:Drm>
```

Zu jedem WMC gehören zwei Einträge: Der Eintrag `<deegree:Name>` wird zur Darstellung in der Oberfläche (Benutzer-Editor) genutzt. Es sollten deshalb möglichst kurze aber sprechende und eindeutige Namen gewählt werden. Der Eintrag `<deegree:URL>` gibt an, wo sich die WebMapContext-Datei im Verzeichnisbaum befindet.
Der Eintrag `<deegree:UserDirectory>` schließlich gibt an, wo sich der Users-Ordner der zugehörigen iGeoPortal-Instanz befindet.

Hinweis: Alle Pfade werden relativ eingetragen, ausgehend vom aktuellen Verzeichnis, hin zu den WMC-Dateien und zum Users-Verzeichnis in iGeoPortal.

4.2.1.3 Erläuterungen zur Administration

Die in Kapitel 4.2.1.1 und 4.2.1.2 beschriebenen Änderungen führen zu folgender veränderter Benutzeroberfläche:

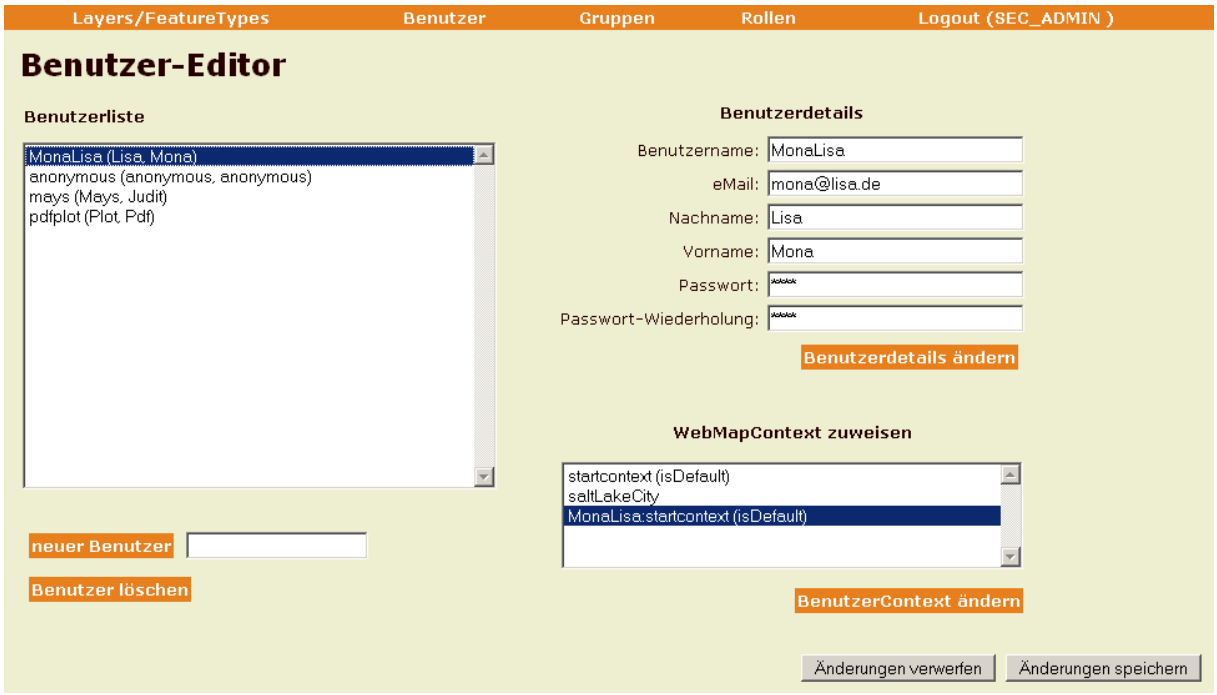

Abbildung 5: Benutzer-Editor mit Context Chooser

Der einfache Benutzer-Editor ist nun um eine Auswahlliste erweitert, die es ermöglicht, den einzelnen Benutzern bestimmte WebMapContext-Dateien zuzuweisen. Ziel ist es über diese Auswahlliste den Startkontext des Nutzers festzulegen, der bei dessen Anmeldung am Portal geladen wird.

Der Startkontext eines Nutzers wird in iGeoPortal über einen entsprechenden Eintrag (`STARTCONTEXT=wmc_datei.xml`) in der Datei `context.properties` bestimmt. Hierzu wird zunächst überprüft, ob der Nutzer in seinem eigenen Nutzer-Verzeichnis (`WEB-INF/conf/igeoportal/users/nutzername/`) diese Datei besitzt. Wenn nicht, wird der Inhalt der gleichnamigen Datei aus dem übergeordneten `users`-Verzeichnis ausgelesen.

Dieser Mechanismus wird in der Erweiterung des Benutzer-Editors verwendet, um Nutzern einen Startkontext zuzuweisen.

Die Administrations-Oberfläche:
In der Auswahlliste “WebMapContext zuweisen” sind alle Startkontexte aufgeführt, die in der Konfigurationsdatei unter `<deegree:availableWMC>` aufgeführt sind. Der Kontext, der in der Konfigurationsdatei mit dem Attribut `isDefault=”1”` versehen ist, wird in der Auswahlliste des Benutzer-Editors mit dem Zusatz `(isDefault)` gekennzeichnet.

Zusätzlich enthält die Auswahlliste einen weiteren Eintrag, der sich aus dem Namen des aktuellen Benutzers und dem ihm zugewiesenen WebMapContext zusammen setzt (Eintrag `MonaLisa:startkontext(isDefault)` in Abbildung 5). Der Zusatz (isDefault) weist darauf hin, dass der Nutzer MonaLisa keine eigene context.properties Datei in seinem Nutzer-Verzeichnis besitzt, sondern auf den Default-Wert zurück gegriffen wird.

Soll einem Nutzer ein anderer Startkontext zugewiesen werden, so wählt man diesen Kontext aus der Liste aus und klickt auf “Benutzerkontext ändern”. Daraufhin ändert sich die Darstellung in der Auswahlliste: Der Eintrag für den aktuellen Benutzer ändert sich auf `Nutzername:neuerKontext`. Gespeichert wird diese Änderung erst dann in der Datenbank, wenn man die Schaltfläche “Änderungen speichern” betätigt.

Wenn das Attribut `isDefault=”1”` in der Konfigurationsdatei einem neuen WebMapContext zugewiesen und der Tomcat neu gestartet wurde (siehe unten), erhält man beim nächsten Aufruf der Administrations-Oberfläche einen entsprechenden Hinweis.

Die Konfigurations-Datei:
Über das Attribut `isDefault="1"` wird festgelegt, welcher Web Map Context in der Datei `WEB-INF/conf/igeoportal/users/context.properties` als Startkontext gespeichert wird. Sinnvollerweise wird man dieses Attribut in der Konfigurationsdatei bei dem Kontext setzen, der den meisten Nutzern als Startkontext bereit gestellt werden soll. Möchte der Administrator den standardmäßigen Startkontext ändern, so kann man das Attribut `"isDefault"` einem anderen WebMapContext der Konfigurationsdatei zuweisen, und die Anwendung im Tomcat neu starten. Dadurch wird automatisch der Wert `STARTCONTEXT` in der oben genannten Datei neu gesetzt, also überschrieben(!).

Wenn iGeoPortal zuvor bereits ohne die drm-admin-Erweiterung konfiguriert wurde, sollte unbedingt der in der allgemeinen `context.properties` Datei eingetragene Startkontext in die Liste der verfügbaren WebMapContexte aufgenommen und mit dem Attribut `isDefault="1"` versehen werden. Nur so kann sichergestellt werden, dass sich der Startkontext der Nutzer nicht ungewollt verändert.

Aus den oben beschriebenen Mechanismen ergibt sich eine wichtige Besonderheit: Es kann vorkommen, dass einem Benutzer der Default-StartKontext als WebMapContext zugewiesen ist, ohne dass der Zusatz `(isDefault)` hinter dem Eintrag `Nutzername:KontextName` vorhanden wäre. Dies bedeutet, dass dieser Nutzer eine eigene context.properties-Datei besitzt, in der der DefaultStartKontext eingetragen ist. Warum ist das so?

Ein Beispiel: Es gibt 2 Kontext-Dateien (kontext1 und kontext2), sowie 3 Nutzer (nutzer1, nutzer2 und nutzer3). Die Zuweisung der Kontext-Dateien zu den Nutzern ist wie folgt:

```
users/context.properties: STARTCONTEXT=./kontext1
users/nutzer1/context.properties STARTCONTEXT=./../kontext1
users/nutzer2/context.properties: STARTCONTEXT=./../kontext2
users/nutzer3/    (keine eigene context.properties-Datei vorhanden)
```

Nun wird die Konfigurationsdatei erstellt und kontext1 erhält den Zusatz isDefault="1". Beim Starten des drm-admin wird erkannt, dass nutzer1 und nutzer2 bereits eigene context.properties-Dateien besitzen, sodass die darin enthaltenen Startkontexte ausgelesen und angezeigt werden. nutzer3 hat keine eigene context.properties-Datei, weshalb ihm der default-Kontext zugewiesen wird:

```
nutzer1:kontext1
nutzer2:kontext2
nutzer3:kontext1(isDefault)
```

Beachte: Obwohl der Kontext für nutzer1 und nutzer3 identisch ist, ist die Darstellung in der Oberfläche nicht gleich. Der Zusatz (isDefault) erscheint nur bei dem Nutzer, der keine eigene context.properties-Datei besitzt.

Nun verschiebt man in der Konfigurationsdatei den Eintrag `isDefault="1"` von kontext1 zu kontex2 und startet sowohl den Tomcat und als auch den drm-admin neu. Man wird beim Öffnen des Benutzer-Editors darüber informiert, dass der Default-Startkontext sich geändert hat. Gleichzeitig wird die Datei `users/context.properties` überschrieben: `STARTCONTEXT=./kontext2`. Dies hat zur Folge, dass im Benutzer-Editor für die jeweiligen Benutzer folgenden Einträge in der Auswahlliste für WebMapContexte zu sehen sind:

```
nutzer1:kontext1
nutzer2:kontext2
nutzer3:kontext2(isDefault).
```

Beachte: nutzer1 hat seinen Kontext behalten, obwohl der Default-Kontext von kontext1 zu kontext2 verändert wurde; nutzer2 hat ebenfalls seinen Kontext behalten, hat aber keinen Zusatz (isDefault) erhalten. Denn: Beide Nutzer hatten vor der Änderung eine eigene context.properties-Datei und haben diese nach der Änderung immer noch. Nur bei nutzer3 ist der Zusatz (isDefault) zu sehen, bei ihm hat sich der Kontext von kontext1 zu kontext2 verändert. Er hat immer noch keine eigene context.properties-Datei.

Wenn man erreichen möchte, dass sich der Startkontext für nutzer1 oder nutzer2 mit verändert, wenn das Attribut isDefault="1" zu einem anderen Kontext verschoben wird, dann muss man vorher dafür sorgen, dass dem entsprechenden Nutzer in der Oberfläche des Benutzer-Editors der Kontext zugewiesen wird, der den Zusatz (isDefault) trägt.

4.3 Gruppen-Editor

Wie oben dargestellt umfasst die deegree Benutzerverwaltung ein Gruppenkonzept bei dem Nutzer zu Gruppen zusammengefasst und Gruppen frei miteinander assoziiert werden können. Die hierzu benötigte Oberfläche gestaltet sich wie in Abbildung 6 dargestellt.

Am linken Rand befindet sich eine Liste aller registrierten Gruppen. Unterhalb der Liste befindet sich ein Textfeld und eine Schaltfläche zur Eingabe neuer Gruppen und eine Schaltfläche zum Löschen bestehender Gruppen. Rechts neben der Gruppenliste befinden sich weitere GUI-Elemente mit denen die Zusammensetzung der jeweils in der Gruppenliste ausgewählten Gruppe editiert werden kann.

Über die beiden oberen Listen besteht die Möglichkeit bereits bestehende Gruppen einer anderen Gruppe zuzuordnen. In Abbildung 6 wurde der Gruppe '*anonymous*' die Gruppe '*privileged*'. Damit 'erben' alle Mitglieder von '*privileged*' automatisch aller Rechte, die an '*anonymous*' vergeben wurden. Im Rahmen dieser Zuweisungen können sowohl baum- als auch netzartige Beziehungen zwischen Gruppen realisiert werden. Die Besonderheit netzartiger Strukturen besteht darin, dass über einen oder mehrere Zwischenschritte zyklische Beziehungen zwischen Gruppen entstehen können. Z.B. könnte '*privileged*' Mitglied von '*anonymous*' sein und '*anonymous*' wäre Mitglied von '*privileged*'. Ergebnis wäre ein Zyklus in dem jede Gruppe mit zwei Zwischenschritten mit sich selbst assoziiert ist.

Die Eingabe direkter oder indirekter Zyklen bei der Gruppendefinition ist grundsätzlich zulässig, doch wird der Benutzer durch eine Meldung in der GUI darauf hingewiesen. Die deegree Benutzerverwaltung besitzt einen internen Mechanismus zur Zyklenprüfung, der auch bei wesentlich komplexeren Situation in der Lage ist diese zu identifizieren. Damit wird bei der Überprüfung von Berechtigungen verhindert, dass ein infiniter Regress entsteht.

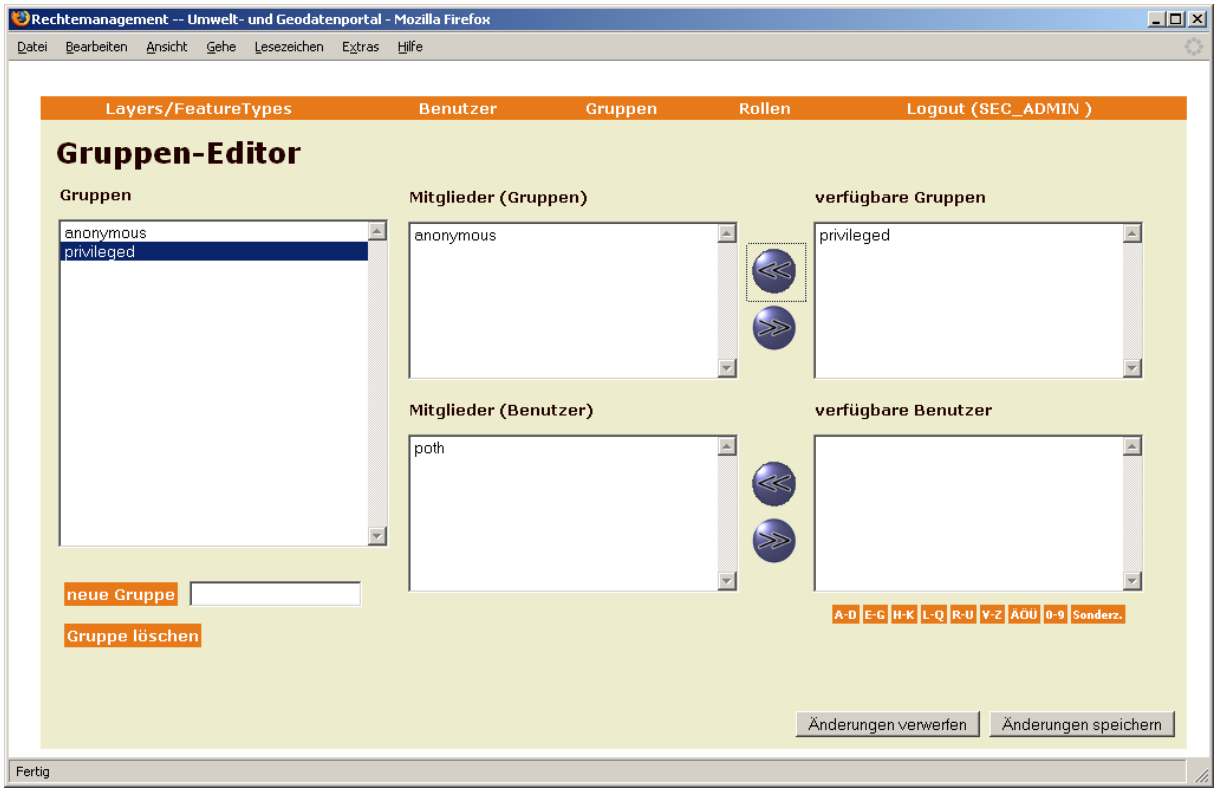

Abbildung 6: Formular zur Verwaltung von Benutzergruppen

Über die beiden unteren Listen in Abbildung 6 können einer Gruppe einzelne Nutzer zugewiesen werden, die aus der Liste der verfügbaren Benutzer ausgewählt werden. Diese ist alphabetisch in mehrere Bereich gegliedert, um das Auffinden/Auswählen bestimmter Nutzer zu erleichtern und die Bedienbarkeit der Anwendung bei einer großen Anzahl an Nutzern sicher zu stellen.

Hinsichtlich der Übernahme in die Datenbank gelten die bereits mehrfach genannten Hinweise.

4.4 Rollen-Editor

Der Rollen-Editor (Abb. 7) ermöglicht es, ausgewählte Gruppen mit Rechten an Datensätzen zu assoziieren. D. h. alle unmittelbaren und mittelbaren Mitglieder einer mit einer Rolle assoziierten Gruppe haben alle Rechte, die in dieser Rolle zusammenfasst sind (s.o.).

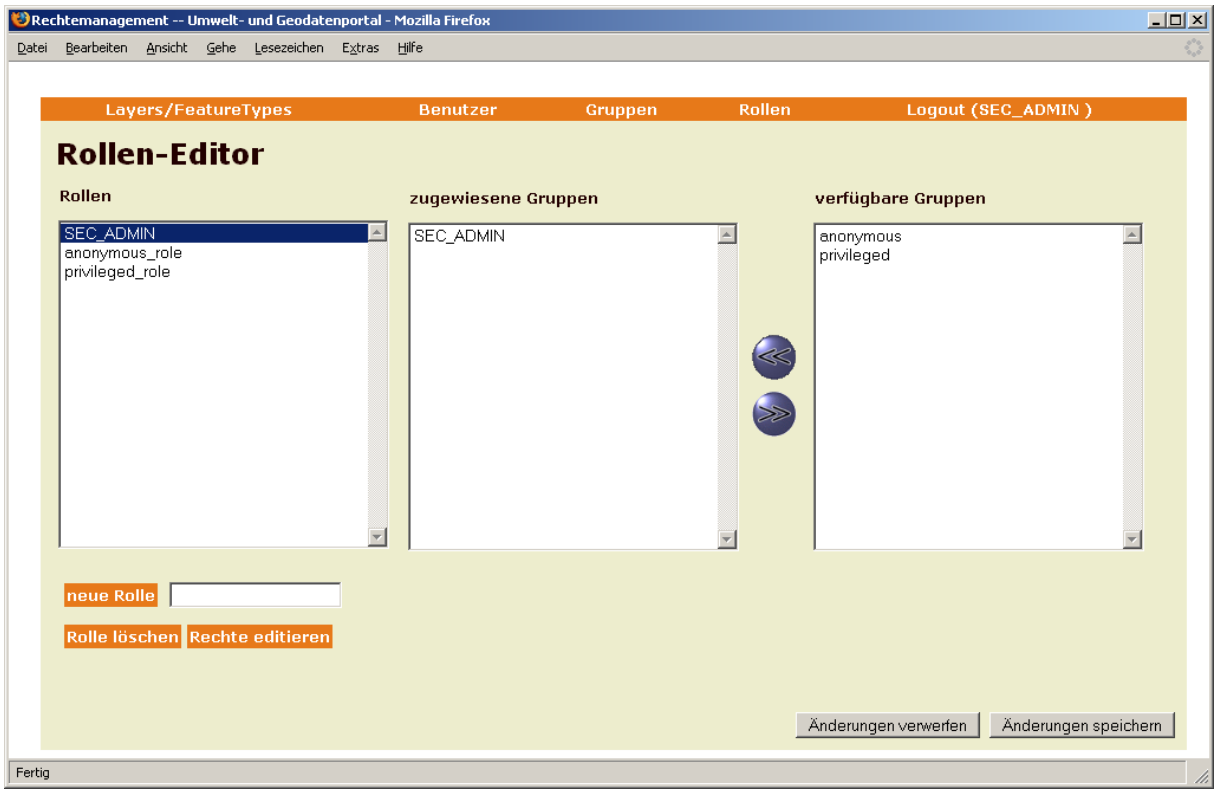

Abbildung 7: Formular zur Verwaltung von Rollen-Gruppen-Zugehörigkeiten

4.4.1 Eine Rolle anlegen

Eingabe des Rollennamens in das Feld neben der Schaltfläche 'neue Rolle anlegen'. Eingabe durch Anklicken Schaltfläche bestätigen. Die neue Rolle erscheint in der Liste 'Rollen'.

4.4.2 Eine Rolle löschen

Man wählt in der linken der drei Listen die zu löschende Rolle aus. Anschließend wählt man die Schaltfläche 'Rolle löschen'. Nach erfolgter Bestätigung wird die Rolle aus der Liste entfernt (gespeichert wird die Veränderung aber erst durch 'übernehmen').

4.4.3 Rollen-Gruppen-Zuordnungen bearbeiten

Auswahl der zu bearbeitenden Rolle in der linken Liste. In der Liste 'zugewiesene Gruppen' erscheinen nun die Gruppen, die dieser Rolle zugeordnet sind. Unter 'verfügbare Gruppen' werden alle dem System bekannten Gruppen aufgelistet, abzüglich derer, die aktuell mit der Rolle verknüpft sind. Um weitere Gruppen mit der Rolle zu verknüpfen, wählt man diese in der rechten Liste aus und klickt auf den Doppelpfeil nach links. Die selektierten Gruppen werden daraufhin in die mittlere Liste verschoben. Analog verfährt man, um die Zuordnung von Gruppen zu einer Rolle zu entfernen; man wählt die zu entfernenden Gruppen in der mittleren Liste und klickt den Doppelpfeil nach rechts.

4.4.4 Eine Rolle editieren

Um die Rechte zu editieren, über die eine Rolle verfügt, selektiert man diese in der linken Liste und wählt 'Rolle editieren'. Dies ist nur möglich, wenn keine nicht gespeicherten Änderungen in der Maske vorhanden sind; ansonsten muss erst 'übernehmen' oder 'abbrechen' gewählt werden.

4.5 Rechte-Editor

Der Rechte-Editor erlaubt die Einstellung der Rechte, über die eine einzelne Rolle verfügt (Abb. 8). Im oberen linken Teil der Maske stehen die FeatureTypes, auf welche die Rolle zugreifen darf (bzw. die mit der Rolle assoziierten Gruppen/Benutzer). Rechts oben stehen die FeatureTypes, für die der Zugriff verweigert wird. Mittels der Doppelpfeil-Schaltflächen zwischen den Listen können die Rechte-Zuordnungen verändert werden (wie bei der Rollen-Gruppen-Zuordnung).

Für jeden FeatureType können die für die aktuelle Rolle zulässigen Transaktionen definiert werden. Hierzu befinden sich unterhalb der Liste der ausgewählten FeatureTypes drei Checkboxen. Mittels eines Mausklicks muss zunächst ein FeatureType ausgewählt werden. Anschliessend können Rechte zum Anlegen (insert), Löschen (delete) und Verändern (update) des ausgewählten FeatureTypes durch (de-)aktivieren vergeben/entzogen werden.

Im unteren Teil werden die Rechte editiert, die die Rolle auf einzelnen Layern besitzt. Auf die Layer in der rechten Liste besitzt die Rolle keinerlei Zugriffsrechte, auf die Layer in der linken Liste darf zugegriffen werden.

Durch einen Mausklick auf die Schaltfläche 'Änderungen speichern' werden alle vergebenen/entzogenen Rechte persistent gemacht. Ein Mausklick auf die Schaltfläche 'Änderungen verwerfen' macht alle nicht persistent gemachten Änderungen an Zugriffsrechten rückgängig.

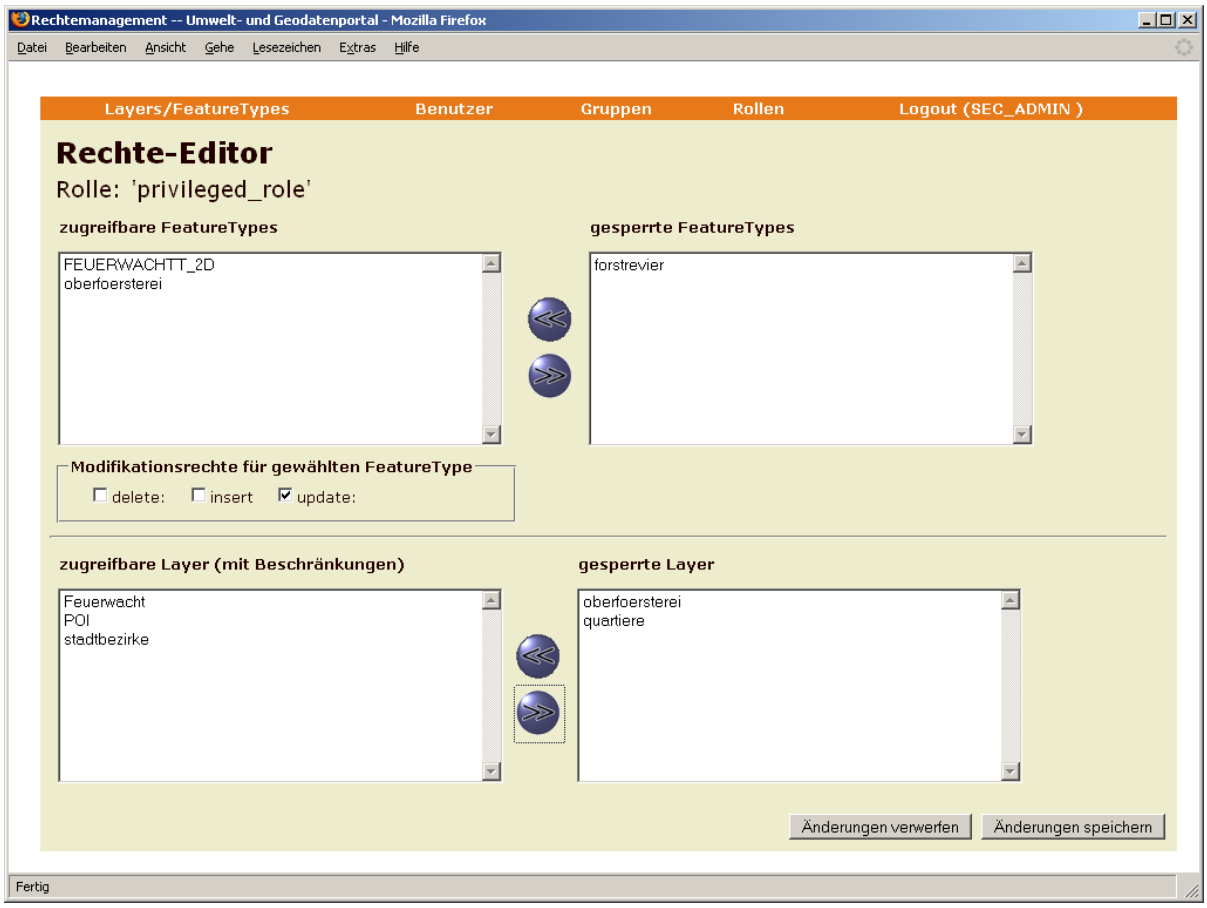

Abbildung 8: Rechte-Editor

5 Konfiguration über Kommandozeile

Neben der Möglichkeit der Definition von Rechten, Rollen, Nutzern und Gruppen mit Hilfe der Weboberfläche von U3R, ist es ebenfalls möglich eine Kommandozeilenwerkzeug zu benutzen. Dieses Werkzeug kann zur Verwaltung von U3R durch andere Programme genutzt werden, z. B. um automatisiert große Mengen von Nutzern oder Layern anzulegen.

5.1 Programmaufruf

Das Werkzeug kann mit dem folgenden Programmaufruf gestartet werden:

```
$JAVA_HOME$/bin/java –classpath .;deegree2.jar;$database JDBC
driver$;$additional libraries$ org.deegree.tools.security.DRMAccess
```

An diese Aufruf müssen die im Weiteren beschriebenen Parameter angehängt werden.

5.2 Allgemeine Parameter

Jede Operation / jeder Aufruf benötigt die Definition der die Datenbankverbindung beschreibenden Parameter.

`-driver [JDBC driver]` (z. B. sun.jdbc.odbc.JdbcOdbcDriver für eine ODBC databases)

`-logon jdbc:odbc:security [logon to database]` (z.B. . ODBC name)

`-user [user name]` (optional)

`-pw [users password]` (optional)

5.3 Aktionen / Operationen

`action (addUser, removeUser, addGroup, removeGroup, addRole, removeRole, addUserToGroup, assignRoleWithGroup, addSecuredObject, removeSecuredObject, assignRights, removeRights, clean)`

definiert die durchzuführende Aktion. Die möglichen Aktionen sind in Klammern angegeben.

addUser -> Legt einen neuen Benutzer an

`-name [users login name]`

`-password [users password]`

`-firstName [first name of the user]`

`-lastName [last name of the user]`

`-email [email address of the user]`

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action addUser -name lkee -password lkee01 -firstName Andreas
-lastName Poth  -email info@lat-lon.de
```

removeUser -> Löscht einen Benutzer

```
-name [users login name]
```

Beispiel:

```
java -classpath .;deegree.jar;$database JDBC driver$
org.deegree.tools.security.DRMAccess -driver
sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security -action
removeUser -name latlon
```

addGroup -> Legt eine neue Gruppe an

```
-name [name of the group]
```

```
-title [title of the group]
```

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action addGroup -name Group1 -title TGroup1
```

removeGroup -> Löscht eine Gruppe

```
-name [name of the group to be removed]
```

addRole -> Legt eine Rolle an

```
-name [name of the role]
```

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action addRole -name Role1
```

removeRole -> Löscht eine Rolle

`-name [name of the role to be removed]`

addUserToGroup -> Fügt einen Nutzer zu einer Gruppe hinzu

`-userName [name of the user]`

`-groupName [name of the group]`

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action addUserToGroup -userName lkee -groupName Group1
```

assignRoleWithGroup -> Ordnet einer Gruppe eine Rolle zu

`-groupName [name of the group]`

`-roleName [name of the role]`

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action assignRoleWithGroup -groupName Group1 -roleName Role1
```

addSecuredObject -> Erzeugt ein neues secured

`-soType [type of the secured object]` (z. B. Layer, FeatureType, Coverage ...)

`-soName [name of the secured object]`

`-soTitle [title of the secured object]`

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action addSecuredObject -soType Layer -soName oberfoersterei
-soTitle Oberfoersterei
```

removeSecuredObject -> Löscht ein secured object

`-soType [type of the secured object]` (z. B. Layer, FeatureType, Coverage ...)

`-soName [name of the secured object]`

assignRights -> Vergibt Rechte an einem bestimmten secured object an eine Rolle

`-constraints [comma separated list of absolut paths to filter encoding files]`

`-rights [comma separated list of rights to assign]` (Die Anzahl der Rechte muss der Zahl der constraints entsprechen)

`-soName [name of the secured object]`

`-soType [type of the secured object]`

`-role [name of the role the rights shall be given to]`

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action assignRights -constraints -;-;- -soName oberfoersterei
-soType Layer -role Role1 -rights
GetLegendGraphic;GetMap;GetFeatureInfo
```

```
.;deegree.jar org.deegree.tools.security.DRMAccess -driver
sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security -action
assignRights -constraints -;e:/temp/ComplexFilter.xml -soName
{http://www.deegree.org/app}:WPVS -soType Featuretype -role
anonymous_role -rights GetFeature,GetFeature_Response
```

ComplexFilter.xml:

```
<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
   <ogc:And>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>horizontalAccuracy</ogc:PropertyName>
         <ogc:Literal>10</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>verticalAccuracy</ogc:PropertyName>
         <ogc:Literal>1</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>xslt</ogc:PropertyName>
         <ogc:Literal>file:/d:/temp/test.xsl</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>instanceFilter</ogc:PropertyName>
```

```
        <ogc:Literal>
            <![CDATA[<ogc:Filter xmlns:ogc="http://www.opengis.net/ogc"
               xmlns:app="http://www.deegree.org/app">
        <ogc:And>
           <ogc:PropertyIsGreaterThan>
              <ogc:PropertyName>app:dateOfBirth</ogc:PropertyName>
              <ogc:Literal>1820</ogc:Literal>
           </ogc:PropertyIsGreaterThan>
           <ogc:PropertyIsEqualTo>
               <ogc:PropertyName>
                  app:placeOfBirth/app:Place/app:country/app:Country/app:name
                </ogc:PropertyName>
              <ogc:Literal>Germany</ogc:Literal>
           </ogc:PropertyIsEqualTo>
        </ogc:And>
      </ogc:Filter>]]></ogc:Literal>
      </ogc:PropertyIsEqualTo>
   </ogc:And>
</ogc:Filter>
```

Folgende Typen von secured objects (-soType) werden zur Zeit unterstützt:

- Layer
- Featuretype
- MetadataSchema

Folgende Rechte (-rights) werden zur Zeit unterstützt:

- access
- query
- delete
- delete_Response
- insert
- insert_Response
- execute
- update
- update_Response
- view
- grant
- GetMap
- GetMap_Response
- GetFeatureInfo
- GetFeatureInfo_Response

- GetLegendGraphic
- GetLegendGraphic_Response
- GetFeature
- GetFeature_Response
- DescribeFeatureType
- DescribeFeatureType_Response
- GetCoverage
- GetCoverage_Response
- DescribeCoverage
- DescribeCoverage_Response
- GetRecords
- GetRecords_Response
- GetRecordById
- GetRecordById_Response
- DescribeRecordType
- DescribeRecordType_Response

removeRights -> Entfernt die Rechte an einem bestimmten secured Object von einer Rolle

```
-rights [comma separated list of rights to remove.]
-soName [name of the secured object]
-soType [type of the secured object]
-role [name of the role the rights shall be given to]
```

Beispiel:

```
java -classpath .;deegree.jar;$database JDBC driver$
org.deegree.tools.security.DRMAccess -driver
sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security -action
removeRights -soName stadtbezirke -soType Layer -role testAccess
-rights GetLegendGraphic
```

hasRight -> Überprüft ob ein bestimmter Nutzer ein bestimmtes Recht besitzt

```
-userName [name of the user]
```

`-password [the users password]`

`-soName [name of the secured object that shall be accessed]`

`-soType [type of the secured object ]`(z. B. Layer, FeatureType, ...)

`-right [right that shall be accessed]` (z. B. GetMap, GetFeature, ...)

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action hasRight -user lkee -password lkee01 -soName oberfoersterei
-soType Layer -right GetMap
```

clean -> Löscht alle Einträge in der U3R-Instanz (!)

Beispiel:

```
java -classpath .;deegree.jar org.deegree.tools.security.DRMAccess
-driver sun.jdbc.odbc.JdbcOdbcDriver -logon jdbc:odbc:security
-action clean
```

Anhang A: U3R Datenbankschema

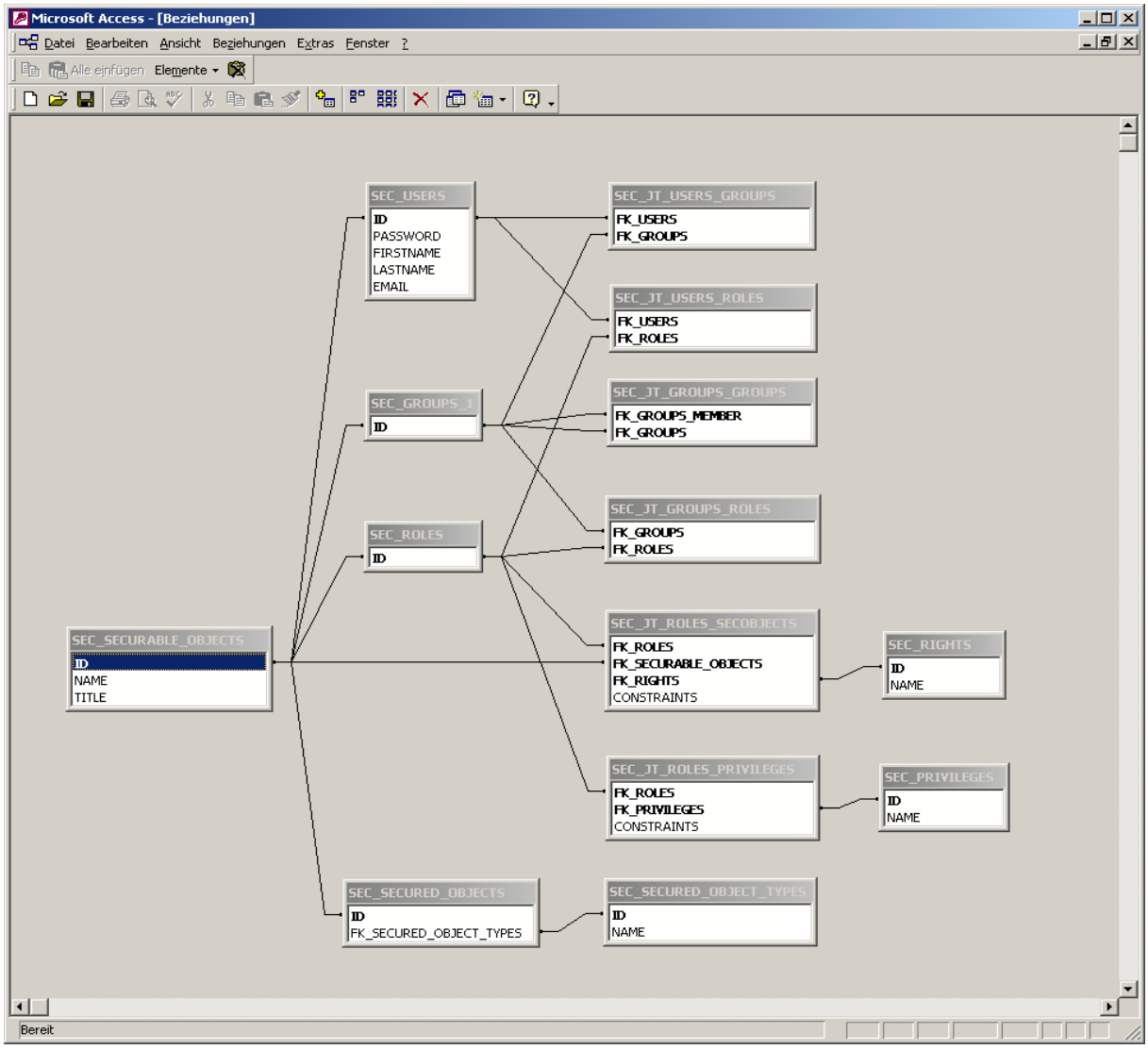

Abbildung 9 Datenbankschema von U3R