

PID-GENERATOR

SOAP-SCHNITTSTELLE

JUTTA MOORMANN

INSTITUT FÜR MEDIZINISCHE BIOMETRIE, EPIDEMIOLOGIE UND INFORMATIK

JOHANNES-GUTENBERG-UNIVERSITÄT MAINZ

`moormann@imbei.uni-mainz.de`

30. MÄRZ 2005

1 Beschreibung

Die SOAP-Schnittstelle des PID-Dienstes ermöglicht die Client-Server- bzw. Server-Server-Kommunikation zwischen einer externen Applikation und dem PID-Dienst (PID-Generator + Patientenliste). Sie wird durch den Webservice `SubjectList` realisiert und bietet Methoden zur Bearbeitung eines PID-Requestes (Methode `getSubjectID`), für die Abfrage von Patientendaten (Methode `getSubjectData`) und für die Überprüfung der Gültigkeit einer PID (Methode `isSubjectIDValid`).

Die Methode `getSubjectID` ruft den PID-Generator über die vorhandene cgi-Schnittstelle auf. Hierzu sind Anpassungen der Konfiguration des PID-Generators erforderlich (siehe 2.1.1). Die anderen Funktionen benötigen keine Einbindung des PID-Generators. Die Abfrage der Patientendaten bzw. der Validität einer PID wird direkt über eine SQL-Query der Patientenliste durchgeführt.

Die Implementierung basiert auf den Spezifikationen vom 20. Januar 2005 (in `pid-interface-concept.20050120.pdf`). Einschränkungen der hier beschriebenen Funktionalität sind unter 3 beschrieben.

2 Installation des Webservices SubjectList

Voraussetzungen für die Installation der SOAP-Schnittstelle sind

- eine Instanz des PID-Generators sowie der Patientenliste
- inklusive des Web-Interfaces, also eines Webservers (z. B. Apache HTTP Server),
- ein Applikationsserver (Apache Jakarta Tomcat 5.0.32),
- die SOAP-Implementierung von Apache (Axis 1.2_RC2),
- das Java Development Kit (JDK 1.4).

Empfohlen wird außerdem die Verwendung des Build-Werzeuges Ant von Apache zur Vereinfachung der Installation.

Die Auslieferung der Software erfolgt in einem komprimierten tar-Archiv (`tmf-soap-interface-1.0.tar.gz`). Dieses kann mit dem Befehl `tar -xzvf tmf-soap-interface-1.0.tar.gz` an beliebiger Stelle entpackt werden, wodurch das Verzeichnis `tmf-soap-interface-1.0` erzeugt wird. Im Weiteren wird der Pfad zu diesem Verzeichnis als `$TMF_HOME` bezeichnet, während die Dateien des PID-Generators im Verzeichnis `$PSX_HOME` und die des Applikationsservers in `$TOMCAT_HOME` liegen.

2.1 Konfiguration der Software

2.1.1 Setup des PID-Generators

Eine Instanz des PID-Generators und der Patientenliste wird vorausgesetzt. Im Folgenden werden nur Maßnahmen beschrieben, die bei der Verwendung der SOAP-Schnittstelle zusätzlich durchgeführt werden müssen.

- Die SOAP-Schnittstelle ruft den PID-Generator über die cgi-Schnittstelle auf. Um das Ergebnis der Anfrage in einem bestimmten Format zu erhalten, werden so genannte Templates definiert (`psx-soap-ret.tpl`, `psx-soap-msg.tpl`). Diese sind im Ordner `$PSX_HOME/tpl` enthalten und bedürfen keiner weiteren Anpassung.
- Bei Verwendung der SOAP-Schnittstelle muss außerdem eine gesonderte Konfigurationsdatei angelegt werden, welche - im Gegensatz zur ursprünglichen Konfigurationsdatei - die Pfade zu den oben erwähnten Templates enthält. Von dieser Änderung abgesehen muss die Datei unbedingt identisch sein mit der ursprünglichen Konfigurationsdatei, damit z. B. die gleichen Schlüssel zur PID-Erzeugung verwendet werden. Dies kann durch einfaches Kopieren der Konfigurationsdatei (z. B. `cp psx.cfg psx-soap.cfg`) mit anschließender Bearbeitung der neu erzeugten Datei erfolgen.

psx-soap.cfg (Auszug)

```
tpl.ret      = $PSX_HOME/tpl/psx-soap-ret.tpl
tpl.msg      = $PSX_HOME/tpl/psx-soap-msg.tpl
```

- Der PID-Generator wird über die cgi-Schnittstelle aufgerufen, d. h. ein entsprechendes Skript `psx-soap.cgi` muss im `cgi-bin`-Verzeichnis vorhanden sein. Die Anweisungen entsprechen denen des „normalen“ Web-Interfaces, jedoch sollte der Pfad für die Konfigurationsdatei auf die eben beschriebene Datei (`psx-soap.cfg`) verweisen.

psx-soap.cgi

```
#!/bin/sh

dir="$PSX_HOME"
prg="${dir}/src/psx"
cfg="${dir}/samples/psx-soap.cfg"
cmd="${prg} gui -c:${cfg}"

${cmd}
```

2.1.2 Zuordnung der OID zu Feldern der Patientenliste

Laut der Schnittstellenbeschreibung sind die OID (Object Identifier) der von Clientseite gelieferten IDAT (Identifizierende Daten) frei wählbar. Im Gegensatz dazu wird der PID-Generator bzw. die Patientenliste einmalig mit vorher festgelegten Feldbezeichnungen initialisiert, die strengerer Regeln unterliegen. Es ist also notwendig, die übermittelten OID auf die Datenbankfelder abzubilden. Diese Zuordnung ist in einer Datei (z. B. `Study-TMF.txt`) enthalten, welche pro Zeile eine OID und den davon durch ein Komma getrennten Namen des Datenbankfeldes enthält (z. B. `Arzt-ID`, `ArztID` oder `Geschlecht`, `sex`). Es müssen unbedingt ALLE Felder, die in der Patientenliste vorhanden sind, berücksichtigt werden (sonst liefert die cgi-Schnittstelle kein sinnvolles Ergebnis). Die einzige Ausnahme bildet das Geburtsdatum. Bis zur Version 1.0 des PID-Generators wurde dieses in drei Feldern gespeichert (`Geburtstag`, `Geburtsmonat`, `Geburtsjahr`), während die IDAT das Geburtsdatum im Format YYYY-MM-DD enthalten. Durch eine Angabe in der Konfigurationsdatei (`decomp.birthdate=„yes“`) wird die automatische Zerlegung des Datums in die drei Felder aktiviert, es reicht dann also die Zuordnung auf den Wert `Geburtsdatum` aus (z. B. `birthdate`, `Geburtsdatum`). Empfohlen wird jedoch die Verwendung der aktuellen Version des PID-Generators (Version 1.1), welche die Speicherung des Geburtsdatums in einem einzigen Feld des Typs DATE erlaubt (s. Manual des PID-Generators). Das Zerlegen des Datums wird dadurch überflüssig und die Bezeichnung des zugehörigen Feldnamens frei wählbar.

Die Zuordnungsdatei des Kompetenznetzes AHF befindet sich in `$TMF_HOME/etc/resources/Study-TMF.txt`.

2.1.3 Anpassung der Konfigurationsdatei

Konfigurationsoptionen werden in der Form `Attribut=„Wert“` angegeben. Ein Beispiel für eine Konfigurationsdatei befindet sich in `$TMF_HOME/etc/SubjectListImplService.cfg`.

Die Schnittstelle sucht die Konfigurationsdatei per Default unter `/etc/tmf/SubjectListImplService.cfg`. Dieses Verzeichnis muss erzeugt und die Datei hierin kopiert oder verschoben werden. Sollte die Konfigurationsdatei unter einem anderen Pfad oder Namen vorliegen, muss der Pfad im Quelltext selber angegeben werden, wie unter 2.1.4 beschrieben.

Folgende Attribute werden in der Konfigurationsdatei definiert:

- `psx.cgi.url`
URL des cgi-Skripts; an diese müssen außerdem die GET-Variablen

`ctx=req` und `sts=1` angehängt werden
(z. B. `https://myhost.mydomain/cgi-bin/psx-soap.cgi?ctx=req&sts=1`).

- **field_map**
absoluter Pfad zur Feldzuordnungsdatei
(z. B. `$TMF_HOME/etc/resources/Study-TMF.txt`)
- **field_sex** (optional, Default: **Geschlecht**)
Der PID-Generator kodiert das Geschlecht mit den Werten m (male), f (female) und n (not known). Da die Schnittstellenspezifikation hier die Werte 1, 2 und 3 vorsieht, ist eine Umkodierung notwendig. Mit **field_sex** wird angegeben, welches Feld die Geschlechtsangaben enthält.
- **DBUrl**
Url der Patientenliste (z. B. `jdbc:postgresql:patliste`)
- **DBUser**
Datenbankbenutzer
- **DBPwd**
Passwort für den Datenbankbenutzer
- **DBDriver**
Datenbanktreiber (getestet mit `org.postgresql.Driver` - hierfür muss die JAR-Datei `postgresql.jar` in den `CLASSPATH` aufgenommen werden)
- **decompose_birthdate** (optional, Default: **no**)
Sollte nur verwendet werden, wenn in der Patientenliste die Felder **Geburtsstag**, **Geburtsmonat** und **Geburtsjahr** verwendet werden, während die Schnittstellenbeschreibung das Datum im Format `YYYY-MM-DD` mit der OID **Geburtsdatum** vorsieht.

2.1.4 Anpassung des Quelltextes

Die eben beschriebene Konfigurationsdatei wird von der SOAP-Schnittstelle eingelesen. Sollte diese nicht im erwarteten Verzeichnis (`/etc/tmf/`) liegen, muss der absolute Pfad der Konfigurationsdatei im Quelltext selber der Variablen `configfile` zugewiesen werden. Diese Anweisung ist in der Datei `$TMF_HOME/src/org/tmf/www/services/SubjectListImpl.java` anzupassen.

2.2 Bereitstellen des Webservices

Die Installation und Bereitstellung des Webservices kann mit Hilfe des mitgelieferten Ant-Skripts (`$TMF_HOME/build.xml`) durchgeführt werden. Dieses steuert

die Kompilierung, das Kopieren der Klassen und das Deployen des Web-Services. Die Verwendung von Apache Jakarta Tomcat als Applikationsserver wird hierbei vorausgesetzt.

2.2.1 Anpassung der build.properties

Als einzige Information benötigt das Ant-Tool die Angabe von `$TOMCAT_HOME` in der Datei `$TMF_HOME/build.properties` (z. B. `tomcat.home = usr/local/jakarta-tomcat-5.0`).

2.2.2 Kompilieren und Deployen des Webservices

In `$TMF_HOME` kann nun das Ant-Tool aufgerufen werden. Mit folgenden Befehlen wird der Webservice für externe Applikationen sichtbar gemacht:

- `ant towebapp`
Kompilieren der Klassen und Kopieren in das Tomcat-Verzeichnis.
- `$TOMCAT_HOME/bin/startup.sh`
Tomcat starten.
- `ant deploy`
Deployen des Webservices.

Das Deployen ist nur einmalig notwendig, die hierfür notwendige Datei `deploy.wsdd` liegt in `$TMF_HOME/src/org/tmf/www/services/`. Bei Änderungen im Quelltext ist lediglich ein erneutes Kompilieren und Kopieren der Dateien (mit `ant towebapp`) und ein Neustart des Tomcat-Servers erforderlich. Der Webservice kann mit `ant undeploy` wieder entfernt werden.

3 Einschränkungen

Wie in 2.1.2 beschrieben, werden die clientseitig übermittelten OID auf die Felder der Patientenliste abgebildet. Die Zuordnung wird in einer anzupassenden, externen Datei definiert, so dass die Schnittstelle generisch für alle Instanzen des PID-Generators einsetzbar ist. Die mitgelieferten Beispiele stellen die Konfiguration und Zuordnung des Kompetenznetzes AHF dar. Nicht umgesetzt wurde die in der Schnittstellendefinition beschriebene Möglichkeit, beliebige „Study Definitions“ einzubinden, welche verschiedenen Clients die Verwendung unterschiedlicher OID erlauben sollte. Die Clients sind also an die Übermittlung der IDAT mit den einmal festgelegten Feldnamen gebunden.