

epSOS Transformation Manager (TM) epSOS Terminology Service Access Manager (TSAM)

Implementation documentation

Milada Kovarova, Roman Repiscak, Frantisek Rudik //PosAm

Slovakia, July 5, 2013

Contents

A.Transformation Manager.....	2
A.1 Short description.....	2
A.2 Public API.....	2
<i>Transformation to epSOS CDA pivot document: toEpSOSPivot()</i>	2
<i>Translation of epSOS CDA pivot document: translate()</i>	3
A.3 Setup.....	3
1.Libraries.....	3
2.Configuration	3
A.4 How to use component.....	4
A.5 Configuration parameters (tm.properties)	5
B.Terminology Services Access Manager	9
B.1 Short description.....	9
B.2 Public API.....	9
<i>Transcoding: getEpSOSConceptByCode()</i>	9
<i>Translation: getDesignationByEpSOSConcept()</i>	9
<i>Providing value sets: getValueSetConcepts()</i>	9
B.3 Setup.....	10
1.Libraries.....	10
2.Configuration	10
3.Database	10
B.4 How to use component.....	10

Implementation documentation

B.5 Configuration parameters (tsam.properties).....	11
C.Appendix	12
C.1 Complete list of TSAM Maven dependencies (testing environment).....	12
C.2 Complete list of TM Maven dependencies (testing environment)	12
C.3 Spring config – TSAM - application_context_tsam.xml (standalone version).....	15
C.4 Spring config – TM - application_context_tm.xml & application_context_tsam.xml	17
C.5 tm.properties.....	21
C.6 tsam.properties.....	23

A. Transformation Manager

A.1 Short description

Transformation Manager (TM) is responsible for data transformation from a national language to the epSOS Reference Terminology and from the epSOS Reference Terminology to a national language. It deals mainly with syntax of original content and epSOS pivot content making use of Terminology Services Access Manager component for transcoding/translation of particular coded elements.

Input content has to be compliant with epSOS specification of CDA documents for

- Patient Summary
- ePrescription
- eDispensation
- Healthcare Encounter Report
- Medication Related Overview

(both CDA level 3 and CDA level 1 embedding PDF).

Main scenarios are:

- data transformation from a national language to the epSOS Reference Terminology (transcoding)
- data transformation from the epSOS Reference Terminology to a national language (translation)

A.2 Public API

Transformation to epSOS CDA pivot document: toEpSOSPivot()

Method:	<i>TMResponseStructure toEpSOSPivot (Document epSOSOriginalData)</i>
Description:	Transformation of an original CDA document to epSOS pivot document.
Input parameters:	EpSOSOriginalData [mandatory] – CDA document in form : org.w3c.dom.Document
Output parameter:	Response Structure: epsos.ccd.posam.tm.response.TMResponseStructure

Implementation documentation

Translation of epSOS CDA pivot document: *translate()*

Method:	<i>translate()</i>
Description:	Transformation of an epSOS pivot document by adding translation to target language.
Input parameters:	<p>EpSOS CDA [mandatory]] – CDA document in form : org.w3c.dom.Document</p> <p>TargetLanguageCode [optional] – String If no language is entered, default value from TSAM configuration is used.</p>
Output parameter:	<p>Response Structure: epsos.ccd.posam.tm.response.TMResponseStructure</p>

A.3 Setup

Java 1.6 is required.

(Java code of Transformation Manager component is Java 1.5 compliant. However, TM is dependent from Audit Service component, which requires Java 1.6 version.)

1.Libraries

TM module directly requires libraries for handling XML files (processing, validation), logging.

It is dependent from TSAM module and AuditService module. (See TSAM documentation and AuditService documentation)

- Logback-classic (0.9.24) – basic logging
- Jcl-over-slf4j (1.6.0) basic logging
- Junit – (junit-4.7.jar) – testing
- Hibernate – (hibernate-core-3.3.1.GA.jar) - Mapping DB – Java Objects
- Spring framework (spring-2.5.6.jar)
- Saxon 9 HE (saxon9.jar) – XSLT processor used in process of schematron validation
- IHE Model based validator (<http://gazelle.ihe.net/epSOS/validator/cdaepsos-validator-app-1.6.zip>) - model based validation of CDA documents

See file pom.xml for basic maven dependencies.

See chapter C.2 for Complete list of TM Maven dependencies.

2. Configuration

2.1 - ***application_context_tm.xml*** – TM component is Spring based. *application_context_tm.xml* is start point of configuration, defines beans, properties which are necessary. Place this file the root of your project (folder in ClassPath).

See chapter C4 for application_context_tm.xml sample.

See chapter A4.1 for sample code for obtaining ApplicationContext.

Following beans are configurable:

Implementation documentation

placeholderConfig - property „locations“ point to properties files, in this case *tsam.properties* & *tm.properties*

tmConfiguration – defines properties for document types, validation, auditing

codedElementList - defines properties for coded element list identification. Singleton containing method for effective work with coded element list used in process of transcoding/translation.

schematronValidator - defines properties for schematron based validation of CDA documents. Bean is singleton which contains methods for validation itself.

2.2 - application_context_tsam.xml - TM module uses TSAM service methods. TSAM spring configuration is imported in *application_context_tm.xml* file. Place this file the root of your project (folder in ClassPath) or according to path in import clause.

See chapter C4 for application_context_tm.xml & application_context_tsam.xml sample. See TSAM documentation for more details. (Chapter B2)

Comment: it is also possible to put contents of both files (*application_context_tm.xml* and *application_context_tsam.xml*) into one file. Then import clause is not needed.

2.3 - **tm.properties**

Place configuration file *tm.properties* to the root of your project (or to path specified in previous step 2.1) and set configuration parameters (section A.5).

Properties in this file are describing paths to some other files needed by processing (described in section A.5).

2.4 - **tsam.properties**

Place configuration file *tsam.properties* to the root of your project (or to path specified in previous step 2.1) and set configuration parameters (section B.5).

Comment: it is also possible to put contents of both files (*tm.properties* and *tsam.properties*) into one file. Then the proper configuration of *placeholderConfig* bean is important, *locations* property should point to this file.

2.5 – **logback.xml**

Logging. Place configuration file *logback.xml* to root of your project. (If needed, see comments inside the file for help with log configuration.)

2.6 – audit service configuration – Audit service is called within TM methods. Audit service configuration files are also needed (*epsos.properties*, *log4j.properties*). Please refer to Audit service implementation documentation for further information.

A.4 How to use component

1. Obtaining TM service

TM module is Spring based. Service can be obtained in following way:

Implementation documentation

```
ClassPathXmlApplicationContext applicationContext=new ClassPathXmlApplicationContext("application_context_tm.xml");  
ITerminologyService service = (ITerminologyService) applicationContext.getBean(ITerminologyService.class.getName());
```

2. Calling methods

TM component has simple API consisting of these 2 methods:

- Call public API method *toEpsosPivot()* to get transcoded version of CDA document
- Call public API method *translate()* to get translated version of CDA document

Code sample:

```
// TRANSCODING  
  
TMResponseStructure response = service.toEpsosPivot(document);  
  
  
// TRANSLATION  
  
TMResponseStructure response = service.translate (document"sk-SK");
```

A.5 Configuration parameters (tm.properties)

See Chapter C5 for sample.

enable/disable model based validation

tm.mda.validation.enabled=true

current path to MDA validator configuration files

tm.mda.cda_xsd_path=config\\validator_res\\xsd\\CDA.xsd

tm.mda.cda_epsos_xsd_path=config\\validator_res\\xsd\\CDA_extended.xsd

tm.mda.cda_xsl_transformer_path=config\\validator_res\\mbvalidatorDetailedResult.xsl

tm.mda.value_set_repository_path=config\\validator_res\\valueSets\\

Audit Trail properties BEGIN

Audit Trail enabled/disabled

tm.audittrail.enabled=false

Audit Trail Event Log - The number of transaction including the epsos- prefix

tm.audittrail.transactionnumber=epsos-94

Audit Trail Event Log - The IP Address of the target Gateway

tm.audittrail.targetip=

Audit Trail Audit Service - The facility number according to log4j

tm.audittrail.facility=13

Audit Trail Audit Service - The severity of the message

tm.audittrail.severity=2

Implementation documentation

Transformation Manager

Terminology Services Access Manager

Audit Trail properties END

Property Name	Type	Value	Description
tm.documenttype.patientsummary	String	60591-5	epSOS code for Patient Summary CDA document
tm.documenttype.eprescription	String	57833-6	epSOS code for ePrescription CDA document
tm.documenttype.edispensation	String	60593-1	epSOS code for eDispensation CDA document
tm.documenttype.hcer	String	34133-9	epSOS code for HCER CDA document
tm.documenttype.mro	String	56445-0	epSOS code for MRO CDA document
tm.codedelementlist.enabled	boolean	true or false	Flag indicating that for identification of coded elements configurable list of elements should be used.
tm.codedelementlist.path	path	path relative or absolute	Path to the XML file (coded_element_list.xml): List of all coded elements which can be found in epSOS CDA pivot document. For each coded element contains: - xpath - usage (document type and optionality) - related value set - value set version - language_code (to which element should be translated, if TargetLanguageCode should not be used)
tm.schema.validation.enabled	boolean	true or false	Flag indicating that schema validation is enabled or disabled.
tm.schemafilepath	path	path relative or absolute	Current path to schema file (CDA_extended.xsd)
tm.schematron.validation.enabled	boolean	true or false	Flag indicating that schematron validation is enabled or disabled.
tm.schematron.path.patientsummary.friendly	path	path relative or absolute	Current path to schematron file for Patient Summary - Friendly.
tm.schematron.path.eprescription.friendly	path	path relative or absolute	Current path to schematron file for ePrescription - Friendly.
tm.schematron.path.edispensation.friendly	path	path relative or absolute	Current path to schematron file for eDispensation - Friendly.
tm.schematron.path.hcer.friendly	path	path relative or	Current path to schematron file for HCER - Friendly.

Implementation documentation

		absolute	
tm.schematron.path.mro.friendly	path	path relative or absolute	Current path to schematron file for MRO - Friendly.
tm.schematron.path.scannedDocument.friendly	path	path relative or absolute	Current path to schematron file for ScannedDocument - Friendly.
tm.schematron.path.patientsummary.pivot	path	path relative or absolute	Current path to schematron file for Patient Summary - Pivot.
tm.schematron.path.eprescription.pivot	path	path relative or absolute	Current path to schematron file for ePrescription - Pivot.
tm.schematron.path.edispensation.pivot	path	path relative or absolute	Current path to schematron file for eDispensation - Pivot.
tm.schematron.path.hcer.pivot	path	path relative or absolute	Current path to schematron file for HCER - Pivot.
tm.schematron.path.mro.pivot	path	path relative or absolute	Current path to schematron file for MRO - Pivot.
tm.schematron.path.scannedDocument.pivot	path	path relative or absolute	Current path to schematron file for ScannedDocument - Pivot.
tm.schematron.path.xsldir	path	path relative or absolute	Current path to schematron XSL directory
tm.mda.validation.enabled	boolean	true or false	Flag indicating that IHE model based validation is enabled or disabled.
tm.mda.cda_xsd_path	path	path relative or absolute	Current paths to IHE model based validator configuration files
tm.mda.cda_epsos_xsd_path	path	path relative or absolute	
tm.mda.cda_xsl_transformer_path	path	path relative or absolute	

Implementation documentation

tm.mda.value_set_repository_path	path	path relative or absolute	
tm.audittrail.enabled	boolean	true or false	Flag indicating that audit trail is enabled or disabled.
tm.audittrail.transactionnumber	String		Audit Trail Event Log - The number of transaction including the epsos- prefix
tm.audittrail.targetip	String		Audit Trail Event Log - The IP Address of the target Gateway
tm.audittrail.facility	String		Audit Trail Audit Service - The facility number according to log4j
tm.audittrail.severity	String		Audit Trail Audit Service - The severity of the message

B. Terminology Services Access Manager

B.1 Short description

Terminology Services Access Manager (TSAM) is responsible for translating a given concept designation into the requested target language as well as transcoding a given "local" coded concept into the appropriate epSOS coded concept using the information present in the Local Terminology Repository.

Main scenarios are

1. **translation** of a given concept designation into the requested target language in a country,
2. **transcoding** of a given "local" coded concept into the appropriate epSOS coded concept.

Translation means association of designation (display name) in a language of a member state to an epSOS coded concept. Operation is symmetric, and can be used to translate from local language to epSOS reference language (i.e. English) or from epSOS reference language to local language. The result of translation is equivalent to the original. Code of the concept remains unchanged.

Transcoding means association of a concept coded in classification system used in member state to an epSOS coded concept. Associated concept has to be either "synonymous" or more specific than the epSOS concept. More than one local concept may be associated with exactly one epSOS concept (N:1 mapping). Therefore, transcoding may be used only when transforming local document to epSOS pivot document. Local code and designation are appended with epSOS concept code and designation.

B.2 Public API

Transcoding: *getEpSOSConceptByCode()*

Method:	<i>getEpSOSConceptByCode()</i>
Description:	Transcoding of SourceConcept based on epSOS reference terminology.
Input parameters:	<ul style="list-style-type: none"> • (CodedElement) SourceConcept [mandatory]
Output parameter:	<ul style="list-style-type: none"> • (TSAMResponseStructure) ResponseStructure

Translation: *getDesignationByEpSOSConcept()*

Method:	<i>getDesignationByEpSOSConcept()</i>
Description:	Translation of SourceConcept based on epSOS reference terminology.
Input parameters:	<ul style="list-style-type: none"> • (CodedElement) SourceConcept [mandatory] • (String) TargetLanguageCode [optional]
Output parameter:	<ul style="list-style-type: none"> • (TSAMResponseStructure) ResponseStructure

Providing value sets: *getValueSetConcepts()*

Method:	<i>getValueSetConcepts()</i>
Description:	Browsing function returning list of all concepts of specified value set (and its version)
Input parameters:	<ul style="list-style-type: none"> • (String) Valueset OID [mandatory] • (String) Valueset Version Name [optional] • (String) LanguageCode [mandatory]
Output parameter:	<ul style="list-style-type: none"> • Collection of RetrievedConcepts

B.3 Setup

1. Libraries

Make sure you have dependant libraries in your classpath:

- Logback-classic (0.9.24) – basic logging
- Jcl-over-slf4j (1.6.0) basic logging
- Junit – (junit-4.7.jar) – testing
- Hibernate – (hibernate-core-3.3.1.GA.jar) - Mapping DB – Java Objects
- Spring framework (spring-2.5.6.jar)
- Saxon 9 HE (saxon9.jar) – XSLT processor used in process of schematron validation

See chapter C1 – complete list of Maven dependencies.

See pom.xml file (basic dependencies) for understanding

2. Configuration

1. Place spring configuration file *application_context_tsam.xml* to the root of your project (folder in ClassPath).

Following beans are configurable:

placeholderConfig – property „locations“ point to properties files, in this case *tsam.properties config*, *datasource* – configurable through properties in *tsam.properties file*

2. Place configuration file *tsam.properties* to the root of your project (or to path specified in previous step 2) and set configuration parameters (section B.5).
3. Logging. Place configuration file *logback.xml* to root of your project. (If needed, see comments inside the file for help with log configuration.)

3. Database

New empty LTR Database schema can be created by running DDL SQL script *schema.sql*.

B.4 How to use component

1. Obtaining TSAM service

TSAM module is Spring based. Service can be obtained in following way:

```
ClassPathXmlApplicationContext applicationContext=new ClassPathXmlApplicationContext("application_context_tsam.xml");  
ITerminologyService service = (ITerminologyService) applicationContext.getBean(ITerminologyService.class.getName());
```

2. Calling methods

TSAM component has very simple API consisting of these 3 methods:

- Call public API method `getEpSOSConceptByCode()` to get associated Epsos Concept for provided SourceConcept

Implementation documentation

- Call public API method `getDesignationByEpSOSConcept` to get translation of provided `SourceConcept` based on Epsos reference terminology
- Call public API method `getValueSetConcepts` to get List of all Concepts for specified value set (and its version)

Code sample:

```
//TRANSLATION
CodedElement ce = new CodedElement(originalXmlElement);
TSAMResponseStructure response = service.getDesignationByEpSOSConcept(ce, "sk-SK");

//TRANSCODING
CodedElement ce = new CodedElement(originalXmlElement);
TSAMResponseStructure response = service.getEpSOSConceptByCode(ce);

//PROVIDING VALUE SETS
List<RetrievedConcept> concepts = service.getValueSetConcepts("1.3.6.1.4.1.12559.11.10.", "MVC 1.0", "en");
```

B.5 Configuration parameters (*tsam.properties*)

Name	Type	Initial Value	Description
transcodingLanguage	String	en	Code of a language, which country A uses for designations in pivot documents created in transcoding (epSOS defines it as English)
translationLanguage	String	en	Code of a language, which country B uses for designations in pivot documents created in translation (local language of a country)
synchronizationEnabled	boolean	true or false	Flag indicating if LTR schema synchronization is enabled
ltr.db.url	String		JDBC Connection string to the database of Local Terminology Repository
ltr.db.user	String		User name for DB
ltr.db.password	String		Password for DB
ltr.db.driverClass	String		JDBC Driver class for LTR database
ltr.hibernate.dialect	String		Hibernate dialect class (depends on DB provider, for details see Hibernate documentation)

Implementation documentation

C. Appendix

C.1 Complete list of TSAM Maven dependencies (testing environment)

junit\junit\3.8.1\junit-3.8.1.jar

org\hibernate\hibernate\3.2.7.ga\hibernate-3.2.7.ga.jar

net\sf\ehcache\ehcache\1.2.3\ehcache-1.2.3.jar

commons-logging\commons-logging\1.0.4\commons-logging-1.0.4.jar

commons-collections\commons-collections\2.1.1\commons-collections-2.1.1.jar

javax\transaction\jta\1.0.1B\jta-1.0.1B.jar

asm\asm-attrs\1.5.3\asm-attrs-1.5.3.jar

dom4j\dom4j\1.6.1\dom4j-1.6.1.jar

antlr\antlr\2.7.6\antlr-2.7.6.jar

cglib\cglib\2.1_3\cglib-2.1_3.jar

asm\asm\1.5.3\asm-1.5.3.jar

org\springframework\spring\2.5.6.SEC02\spring-2.5.6.SEC02.jar

com\h2database\h2\1.2.142\h2-1.2.142.jar

commons-dbcp\commons-dbcp\1.4\commons-dbcp-1.4.jar

commons-pool\commons-pool\1.5.4\commons-pool-1.5.4.jar

ch\qos\logback\logback-classic\0.9.24\logback-classic-0.9.24.jar

ch\qos\logback\logback-core\0.9.24\logback-core-0.9.24.jar

org\slf4j\slf4j-api\1.6.0\slf4j-api-1.6.0.jar

org\slf4j\jcl-over-slf4j\1.6.0\jcl-over-slf4j-1.6.0.jar

C.2 Complete list of TM Maven dependencies (testing environment)

junit\junit\3.8.1\junit-3.8.1.jar

org\hibernate\hibernate\3.2.7.ga\hibernate-3.2.7.ga.jar

net\sf\ehcache\ehcache\1.2.3\ehcache-1.2.3.jar

commons-logging\commons-logging\1.0.4\commons-logging-1.0.4.jar

commons-collections\commons-collections\2.1.1\commons-collections-2.1.1.jar

javax\transaction\jta\1.0.1B\jta-1.0.1B.jar
asm\asm-attrs\1.5.3\asm-attrs-1.5.3.jar
dom4j\dom4j\1.6.1\dom4j-1.6.1.jar
antlr\antlr\2.7.6\antlr-2.7.6.jar
cglib\cglib\2.1_3\cglib-2.1_3.jar
asm\asm\1.5.3\asm-1.5.3.jar
org\springframework\spring\2.5.6.SEC02\spring-2.5.6.SEC02.jar
com\h2database\h2\1.2.142\h2-1.2.142.jar
commons-dbcp\commons-dbcp\1.4\commons-dbcp-1.4.jar
commons-pool\commons-pool\1.5.4\commons-pool-1.5.4.jar
ch\qos\logback\logback-classic\0.9.24\logback-classic-0.9.24.jar
ch\qos\logback\logback-core\0.9.24\logback-core-0.9.24.jar
org\slf4j\slf4j-api\1.6.0\slf4j-api-1.6.0.jar
org\slf4j\jcl-over-slf4j\1.6.0\jcl-over-slf4j-1.6.0.jar
net\sourceforge\saxon\saxon\9.1.0.8\saxon-9.1.0.8-dom.jar
net\sourceforge\saxon\saxon\9.1.0.8\saxon-9.1.0.8.jar
log4j\log4j\1.2.15\log4j-1.2.15.jar
javax\mail\mail\1.4\mail-1.4.jar
javax\activation\activation\1.1\activation-1.1.jar
javax\jms\jms\1.1\jms-1.1.jar
com\sun\jdk\jmxtools\1.2.1\jmxtools-1.2.1.jar
com\sun\jmx\jmxri\1.2.1\jmxri-1.2.1.jar
javax\xml\bind\jaxb-api\2.2\jaxb-api-2.2.jar
javax\xml\stream\stax-api\1.0-2\stax-api-1.0-2.jar
javax\xml\crypto\com.springsource.javax.xml.crypto\1.4.2\com.springsource.javax.xml.crypto-1.4.2.jar
org\apache\commons\com.springsource.org.apache.commons.logging\1.1.1\com.springsource.org.apache.commons.logging-1.1.1.jar
org\apache\xalan\com.springsource.org.apache.xalan\2.7.0\com.springsource.org.apache.xalan-2.7.0.jar
org\apache\xerces\com.springsource.org.apache.xerces\2.8.1\com.springsource.org.apache.xerces-2.8.1.jar
org\apache\xml\com.springsource.org.apache.xml.resolver\1.2.0\com.springsource.org.apache.xml.resolver-1.2.0.jar
org\apache\xmlcommons\com.springsource.org.apache.xmlcommons\1.3.3\com.springsource.org.apache.xmlcommons-1.3.3.jar
org\apache\bcel\com.springsource.org.apache.bcel\5.1.0\com.springsource.org.apache.bcel-5.1.0.jar

Implementation documentation



org\apache\regex\com.springsource.org.apache.regex\1.5.0\com.springsource.org.apache.regex-1.5.0.jar

epsos\ccd\posam\terminology-service-access-manager\1.0.6\terminology-service-access-manager-1.0.6.jar

lib\auditservice.jar

lib\configmanager.jar

lib\SecurityManager-1.0-SNAPSHOT.jar

lib\xmltooling-1.2.1.jar

lib\opensaml-2.3.0.jar

Implementation documentation

C.3 Spring config – TSAM - application_context_tsam.xml (standalone version)

This configuration allows standalone usage of TSAM component – for example in Junit tests.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
    http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
    http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-2.5.xsd">

  <bean id="placeholderConfig"
    class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="locations">
      <list>
        <value>tsam.properties</value>
      </list>
    </property>
  </bean>

  <bean id="config" class="epsos.ccd.posam.tsam.util.TsamConfiguration">
    <property name="translationLang" value="{translationLanguage}"/>
    <property name="transcodingLang" value="{transcodingLanguage}"/>
    <property name="synchronizationEnabled" value="{synchronizationEnabled}"/>
  </bean>

  <bean id="datasource" class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName" value="{tr.db.driverClass}" />
    <property name="url" value="{tr.db.url}" />
    <property name="username" value="{tr.db.user}" />
    <property name="password" value="{tr.db.password}" />
  </bean>

  <bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource" ref="datasource" />
    <property name="hibernateProperties">
      <props>
        <prop key="hibernate.dialect">{tr.hibernate.dialect}</prop>
        <prop key="hibernate.show_sql">true</prop>
        <prop key="hibernate.cache.use_query_cache">true</prop>
        <prop key="hibernate.cache.provider_class">net.sf.ehcache.hibernate.SingletonEhCacheProvider</prop>
        <prop key="hibernate.generate_statistics">true</prop>
        <prop key="hibernate.cache.use_structured_entries">true</prop>
      </props>
    </property>
    <property name="mappingResources">
      <list>
        <value>CodeSystem.hbm.xml</value>
        <value>CodeSystemConcept.hbm.xml</value>
        <value>CodeSystemVersion.hbm.xml</value>
        <value>Designation.hbm.xml</value>
        <value>ValueSet.hbm.xml</value>
        <value>ValueSetVersion.hbm.xml</value>
        <value>TranscodingAssociation.hbm.xml</value>
      </list>
    </property>
  </bean>
</beans>
```

Implementation documentation

```
<tx:annotation-driven/>

<bean id="transactionManager"
      class="org.springframework.orm.hibernate3.HibernateTransactionManager">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>

<bean id="epsos.ccd.posam.tsam.dao.ITsamDao" class="epsos.ccd.posam.tsam.dao.impl.TsamDao">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>

<bean id="epsos.ccd.posam.tsam.service.ITerminologyService" class="epsos.ccd.posam.tsam.service.impl.TerminologyService">
  <property name="dao" ref="epsos.ccd.posam.tsam.dao.ITsamDao"></property>
</bean>

</beans>
```

Implementation documentation

C.4 Spring config – TM - application_context_tm.xml & application_context_tsam.xml

application_context_tm.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-
beans-2.0.xsd
       http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.0.xsd" >

    <!--
        allow the user of properties from application.properties throughout
        this configuration file(s)
    -->
    <bean id="placeholderConfig"
        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="locations">
            <list>
                <value>config\tsam.properties</value>
                <value>config\tm.properties</value>
            </list>
        </property>
    </bean>

    <bean id="tmConfiguration" class="epsos.ccd.posam.tm.util.TMConfiguration"
        factory-method="getInstance">
        <property name="patientSummaryCode">
            <value>${tm.documenttype.patientsummary}</value>
        </property>
        <property name="ePrescriptionCode">
            <value>${tm.documenttype.eprescription}</value>
        </property>
        <property name="eDispensationCode">
            <value>${tm.documenttype.edispensation}</value>
        </property>
        <property name="hcerCode">
            <value>${tm.documenttype.hcer}</value>
        </property>
        <property name="mroCode">
            <value>${tm.documenttype.mro}</value>
        </property>

        <property name="schemaFilePath">
            <value>${tm.schemafilepath}</value>
        </property>

        <property name="patientSummarySchematronFriendlyPath">
            <value>${tm.schematron.path.patientsummary.friendly}</value>
        </property>
        <property name="ePrescriptionSchematronFriendlyPath">
            <value>${tm.schematron.path.eprescription.friendly}</value>
        </property>
        <property name="eDispensationSchematronFriendlyPath">
            <value>${tm.schematron.path.edispensation.friendly}</value>
        </property>

        <property name="patientSummarySchematronPivotPath">
            <value>${tm.schematron.path.patientsummary.pivot}</value>
        </property>
        <property name="ePrescriptionSchematronPivotPath">
            <value>${tm.schematron.path.eprescription.pivot}</value>
        </property>
    </bean>
</beans>
```

Implementation documentation

```

<property name="eDispensationSchematronPivotPath">
  <value>${tm.schematron.path.edispensation.pivot}</value>
</property>

<property name="mroSchematronFriendlyPath">
  <value>${tm.schematron.path.scannedDocument.friendly}</value>
</property>
<property name="mroSchematronPivotPath">
  <value>${tm.schematron.path.scannedDocument.pivot}</value>
</property>

<property name="hcerSchematronFriendlyPath">
  <value>${tm.schematron.path.scannedDocument.friendly}</value>
</property>
<property name="hcerSchematronPivotPath">
  <value>${tm.schematron.path.scannedDocument.pivot}</value>
</property>

<property name="scannedDocumentFriendlyPath">
  <value>${tm.schematron.path.scannedDocument.friendly}</value>
</property>
<property name="scannedDocumentPivotPath">
  <value>${tm.schematron.path.scannedDocument.pivot}</value>
</property>

<property name="schematronValidationEnabled">
  <value>${tm.schematron.validation.enabled}</value>
</property>
<property name="schemaValidationEnabled">
  <value>${tm.schema.validation.enabled}</value>
</property>
<property name="auditTrailEnabled">
  <value>${tm.audittrail.enabled}</value>
</property>
<property name="auditTrailTransactionNumber">
  <value>${tm.audittrail.transactionnumber}</value>
</property>
<property name="auditTrailTargetIP">
  <value>${tm.audittrail.targetip}</value>
</property>
<property name="auditTrailFacility">
  <value>${tm.audittrail.facility}</value>
</property>
<property name="auditTrailSeverity">
  <value>${tm.audittrail.severity}</value>
</property>
<property name="modelValidationEnabled">
  <value>${tm.mda.validation.enabled}</value>
</property>
<property name="mdaCdaXsdPath">
  <value>${tm.mda.cda_xsd_path}</value>
</property>
<property name="mdaCdaEpsosXsdPath">
  <value>${tm.mda.cda_epsos_xsd_path}</value>
</property>
<property name="mdaCdaXslTransformerPath">
  <value>${tm.mda.cda_xsl_transformer_path}</value>
</property>
<property name="mdaValuesetRepositoryPath">
  <value>${tm.mda.value_set_repository_path}</value>
</property>

```

```

</bean>

```

```

<bean id="codedElementList" class="epsos.ccd.posam.tm.util.CodedElementList"

```

Implementation documentation

```

        factory-method="getInstance">
        <property name="configurableElementIdentification">
            <value>${tm.codedelementlist.enabled}</value>
        </property>
        <property name="codedElementListPath">
            <value>${tm.codedelementlist.path}</value>
        </property>
    </bean>

    <bean id="schematronValidator" class="epsos.ccd.posam.tm.util.SchematronValidator"
        factory-method="getInstance">
        <property name="xslDirectoryPath">
            <value>${tm.schematron.path.xsldir}</value>
        </property>
        <property name="config" ref="tmConfiguration"/>
    </bean>

    <bean id="mdaValidator" class="epsos.ccd.posam.tm.util.ModelBasedValidator"
        factory-method="getInstance">
        <property name="config" ref="tmConfiguration"/>
    </bean>

    <!-- ::: Service ::: -->
    <import resource="ctx_tsam.xml" />

    <bean class="epsos.ccd.posam.tm.service.impl.TransformationService"
        id="epsos.ccd.posam.tm.service.ITransformationService">
        <property name="tsamApi"
            ref="epsos.ccd.posam.tsam.service.ITerminologyService" />
        <property name="config" ref="tmConfiguration"/>
    </bean>
</beans>

```

application_context_tsam.xml (imported by application_context_tm.xml):

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
        http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd">

    <!-- placeholderConfig is defined in application_context_tm.xml-->
    <!--
    <bean id="placeholderConfig"
        class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
        <property name="locations">
            <list>
                <value>classpath:tsam.properties</value>
            </list>
        </property>
    </bean>
    -->

    <bean id="config" class="epsos.ccd.posam.tsam.util.TsamConfiguration">
        <property name="translationLang" value="${translationLanguage}"/>
        <property name="transcodingLang" value="${transcodingLanguage}"/>
        <property name="synchronizationEnabled" value="${synchronizationEnabled}"/>
    </bean>

```

Implementation documentation

```

<bean id="datasource" class="org.apache.commons.dbcp.BasicDataSource">
  <property name="driverClassName" value="\${ltr.db.driverClass}" />
  <property name="url" value="\${ltr.db.url}" />
  <property name="username" value="\${ltr.db.user}" />
  <property name="password" value="\${ltr.db.password}" />
</bean>

<bean id="sessionFactory"
  class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
  <property name="dataSource" ref="datasource" />
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">org.hibernate.dialect.H2Dialect</prop>
      <prop key="hibernate.show_sql">>true</prop>
      <prop key="hibernate.cache.use_query_cache">>true</prop>
      <prop
key="hibernate.cache.provider_class">net.sf.ehcache.hibernate.SingletonEhCacheProvider</prop>
      <prop key="hibernate.generate_statistics">>true</prop>
      <prop key="hibernate.cache.use_structured_entries">>true</prop>
    </props>
  </property>
  <property name="mappingResources">
    <list>
      <value>CodeSystem.hbm.xml</value>
      <value>CodeSystemConcept.hbm.xml</value>
      <value>CodeSystemVersion.hbm.xml</value>
      <value>Designation.hbm.xml</value>
      <value>ValueSet.hbm.xml</value>
      <value>ValueSetVersion.hbm.xml</value>
      <value>TranscodingAssociation.hbm.xml</value>
    </list>
  </property>
</bean>

<tx:annotation-driven/>

<bean id="transactionManager"
  class="org.springframework.orm.hibernate3.HibernateTransactionManager">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>

<bean id="epsos.ccd.posam.tsam.dao.ITsamDao" class="epsos.ccd.posam.tsam.dao.impl.TsamDao">
  <property name="sessionFactory" ref="sessionFactory" />
</bean>

<bean id="epsos.ccd.posam.tsam.service.ITerminologyService" class="epsos.ccd.posam.tsam.service.impl.TerminologyService">
  <property name="dao" ref="epsos.ccd.posam.tsam.dao.ITsamDao"></property>
</bean>

</beans>

```

Implementation documentation

C.5 tm.properties

```
# epSOS code for Patient summary CDA document
tm.documenttype.patientsummary=60591-5

# epSOS code for ePrescription CDA document
tm.documenttype.eprescription=57833-6

# epSOS code for eDispensation CDA document
tm.documenttype.edispensation=60593-1

# epSOS code for HCER CDA document
tm.documenttype.hcer=34133-9

# epSOS code for MRO CDA document
tm.documenttype.mro=56445-0

# enable/disable coded element processing
tm.codedelementlist.enabled=true

# current path to XML file containing coded element list
tm.codedelementlist.path=config\\coded_element_list_epSOS2.xml

# enable/disable schema validation
tm.schema.validation.enabled=false

# current path to schema file
tm.schemafilepath=config\\schema\\CDA_extended.xsd

# enable/disable schematron validation
tm.schematron.validation.enabled=false

# current path to schematron file for Patient Summary CDA document in „epSOS friendly“ format
tm.schematron.path.patientsummary.friendly=config\\schematron\\epSOS-PatientSummary-Friendly.sch

# current path to schematron file for ePrescription CDA document in „epSOS friendly“ format
tm.schematron.path.eprescription.friendly=config\\schematron\\epSOS-ePrescription-Friendly.sch

# current path to schematron file for eDispensation CDA document in „epSOS friendly“ format
tm.schematron.path.edispensation.friendly=config\\schematron\\epSOS-eDispensation-Friendly.sch

# current path to schematron file for HCER CDA document in „epSOS friendly“ format
tm.schematron.path.hcer.friendly=config\\schematron\\epSOS-HCER-Friendly.sch

# current path to schematron file for MRO CDA document in „epSOS friendly“ format
tm.schematron.path.mro.friendly=config\\schematron\\epSOS-MRO-Friendly.sch

# current path to schematron file for scanned document (CDA level 1) in „epSOS friendly“ format
tm.schematron.path.scannedDocument.friendly=config\\schematron\\epSOS-ScannedDocument-Friendly.sch

# current path to schematron file for Patient Summary CDA document in „epSOS pivot“ format
tm.schematron.path.patientsummary.pivot=config\\schematron\\epSOS-PatientSummary-Pivot.sch

# current path to schematron file for ePrescription CDA document in „epSOS pivot“ format
tm.schematron.path.eprescription.pivot=config\\schematron\\epSOS-ePrescription-Pivot.sch

# current path to schematron file for eDispensation CDA document in „epSOS pivot“ format
tm.schematron.path.edispensation.pivot=config\\schematron\\epSOS-eDispensation-Pivot.sch

# current path to schematron file for HCER CDA document in „epSOS pivot“ format
tm.schematron.path.hcer.pivot=config\\schematron\\epSOS-HCER-Pivot.sch

# current path to schematron file for MRO CDA document in „epSOS pivot“ format
tm.schematron.path.mro.pivot=config\\schematron\\epSOS-MRO-Pivot.sch

# current path to schematron file for scanned document (CDA level 1) in „epSOS pivot“ format
```

Implementation documentation

```
tm.schematron.path.scannedDocument.pivot=config\schematron\ epSOS-ScannedDocument-Pivot.sch

# current path to schematron XSL directory
tm.schematron.path.xsldir=config\xsl

# enable/disable model based validation
tm.mda.validation.enabled=true

# current path to MDA validator configuration files
tm.mda.cda_xsd_path=config\validator_res\xsd\CDA.xsd
tm.mda.cda_epsos_xsd_path=config\validator_res\xsd\CDA_extended.xsd
tm.mda.cda_xsl_transformer_path=config\validator_res\mbvalidatorDetailedResult.xsl
tm.mda.value_set_repository_path=config\validator_res\valueSets\

# Audit Trail properties BEGIN
# Audit Trail enabled/disabled
tm.audittrail.enabled=false
# Audit Trail Event Log - The number of transaction including the epsos- prefix
tm.audittrail.transactionnumber=epsos-94
# Audit Trail Event Log - The IP Address of the target Gateway
tm.audittrail.targetip=
# Audit Trail Audit Service - The facility number according to log4j
tm.audittrail.facility=13
# Audit Trail Audit Service - The severity of the message
tm.audittrail.severity=2
# Audit Trail properties END
```

C.6 tsam.properties

```
translationLanguage=en  
transcodingLanguage=en  
synchronizationEnabled=false
```

```
ltr.db.url=jdbc:h2:file:db/tsam;AUTO_SERVER=TRUE  
ltr.db.user=sa  
ltr.db.password=  
ltr.hibernate.dialect=org.hibernate.dialect.H2Dialect  
ltr.db.driverClass=org.h2.Driver
```